User's Manual

16

78K0R/KC3-L, 78K0R/KE3-L (On-Chip USB Controller)

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

μPD78F1022
μPD78F1023
μPD78F1024
μPD78F1025
μPD78F1026

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.2.01 Sep 2011

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/KC3-L and 78K0R/KE3-L and design and develop application systems and programs for these devices.

The target products are as follows.

- 78K0R/KC3-L: μPD78F1022, 78F1023, 78F1024
- 78K0R/KE3-L: μPD78F1025, 78F1026

Purpose

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

Organization

The 78K0R/KC3-L, KE3-L manual is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller).

78K0R/KC3-L, KE3-L 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**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what." field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R.
- To know details of the 78K0R Microcontroller instructions:
  → Refer to the separate document **78K0R Microcontroller Instructions User's Manual (R01US0029E)**.

<R>

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/KC3-L, KE3-L User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	R01US0029E

<R>

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	U18010E
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-MINI2 On-Chip Debug Emulator with Programming Function	U18371E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R20UT0008E

<R>

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.
<R> RENESAS MICROCOMPUTER GENERAL CATALOG	R01CS0001E
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.

All trademarks and registered trademarks are the property of their respective owners.
EEPROM is a trademark of Renesas Electronics Corporation.
Windows and Windows NT are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

- ○ Minimum instruction execution time can be changed from high speed (0.05 μs: @ 20 MHz operation (When USB is not in use) 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

	Product name		Program memory (ROM)		Data memory (RAM)
	Referred to as	Part number			
<R>	78K0R/KC3-L (48 pin)	μPD78F1022	Flash memory	64 KB	6KB
		μPD78F1023		96 KB	8 KB Note
		μPD78F1024		128 KB	8 KB Note
<R>	78K0R/KE3-L (64 pin)	μPD78F1025		96 KB	8 KB Note
		μPD78F1026		128 KB	8 KB Note

Note This is 7 KB when the self-programming function is used.

- ○ On-chip internal high-speed oscillation clocks
  - 20 MHz Internal high-speed oscillation clock: 20 MHz (TYP.)
  - 8 MHz Internal high-speed oscillation clock: 8 MHz (TYP.)
  - 1 MHz Internal high-speed oscillation clock: 1 MHz (TYP.)
- ○ 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 (operable with the dedicated internal low-speed oscillation clock)
- ○ On-chip multiplier/divider (16 bits × 16 bits, 32 bits ÷ 32 bits)
- ○ On-chip key interrupt function
- ○ On-chip clock output/buzzer output controller
- ○ On-chip BCD adjustment
- ○ I/O ports: 53/39 (N-ch open drain: 4)
- ○ Timer: 10 channels
  - 16-bit timer: 8 channels
  - Watchdog timer: 1 channel
  - Real-time counter: 1 channel
- ○ Serial interface
  - CSI: 1 channel/UART: 1 channel
  - CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
  - CSI: 1 channel Note /UART: 1 channel Note /simplified I^2C: 1 channel Note
  - UART (LIN-bus supported): 1 channel
  - I^2C: 1 channel

- USB controller
  - USB function (Full speed): 1 Channel

Note Only for 78K0R/KE3-L

- 10-bit resolution A/D converter (AV_{REF} = 1.8 to 3.6 V): 8 channels
- <R> Power supply voltage: V_{DD} = 1.8 to 3.6 V (When USB is not in use), V_{DD} = 3.0 to 3.6 V (When USB is in use),
- Operating ambient temperature: T_A = –40 to +85°C

1.2 Applications

- Equipment using USB interface
- Audio visual equipment
- Home appliances
- Industrial equipment
- Health care equipment

1.3 Ordering Information

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

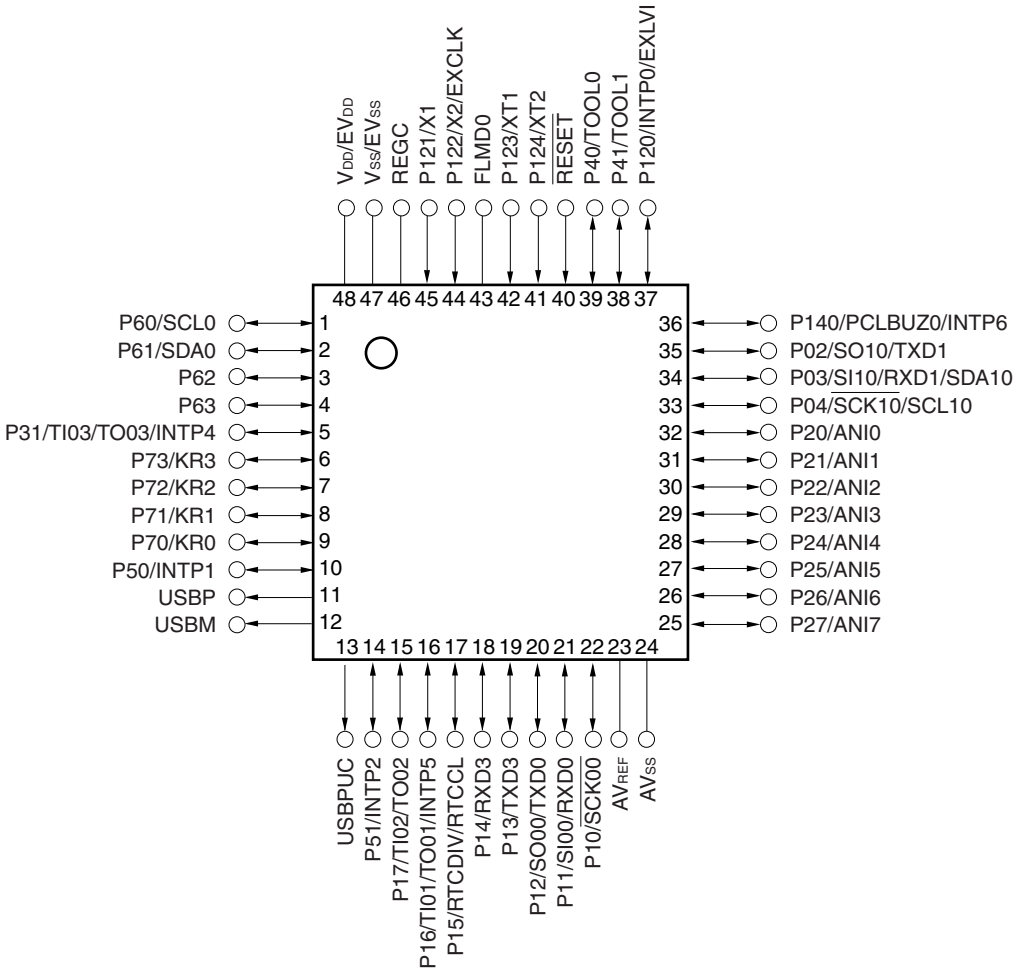
78K0R/Kx3-L Microcontroller	Package	Part Number
<R> 78K0R/KC3-L	48 pin Plastic TQFP(Fine pitch)(7x7)	μPD78F1022GA-HAA-AX, 78F1023GA-HAA-AX, 78F1024GA-HAA-AX
	48 pin Plastic WQFN(Fine pitch)(7x7)	μPD78F1022K8-5B4-AX, 78F1023K8-5B4-AX, 78F1024K8-5B4-AX,
<R> 78K0R/KE3-L	64 pin Plastic LQFP(Fine pitch)(10x10)	μPD78F1025GB-GAH-AX, 78F1026GB-GAH-AX
	64 pin Plastic TQFP(Fine pitch)(7x7)	μPD78F1025GA-HAB-AX, 78F1026GA-HAB-AX
	64 pin Plastic FBGA(5x5)	μPD78F1025F1-AN9-A, 78F1026F1-AN9-A

Caution The 78K0R/KC3-L, 78K0R/KE3-L has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

1.4 Pin Configuration (Top View)

- 48-pin plastic TQFP (Fine pitch) (7 × 7)

<R> • 48-pin plastic WQFN (7 × 7)



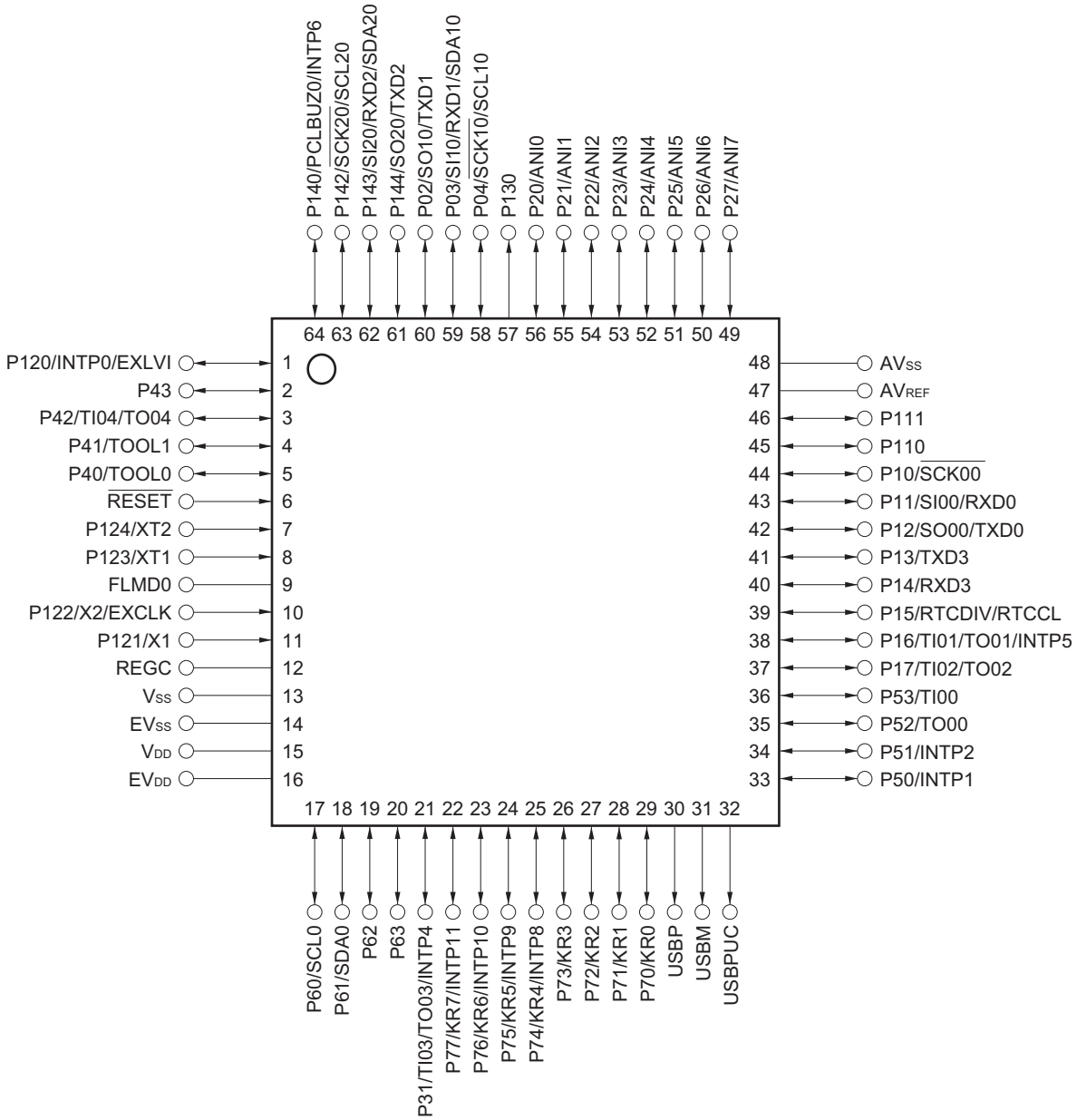
Cautions 1. Make AVSS and EVSS the same potential as VSS.

<R> **2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).**

3. P20/ANI0 to P27/ANI7 are set as analog inputs in the order of P27/ANI7, …, P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 as analog inputs, start designing from P27/ANI17 (see 10.3 (6) A/D port configuration register (ADPC) for details).

<R> • 64-pin plastic TQFP (7 × 7)

<R> • 64-pin plastic LQFP (10 × 10)



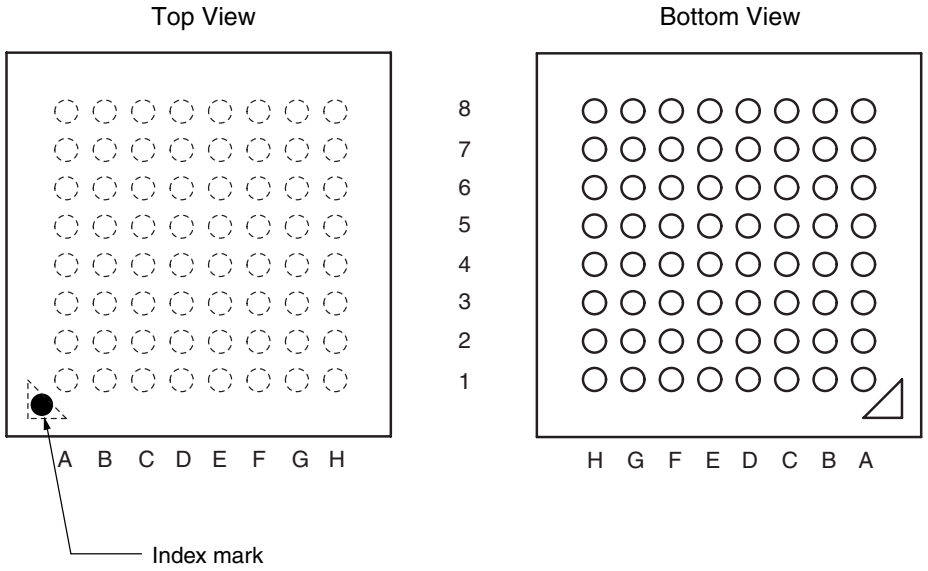
Cautions 1. Make AVSS and EVSS the same potential as VSS.

2. Please make EVDD the same potential as VDD.

<R> **3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).**

4. P20/ANI0 - P27/ANI7 are set as analog inputs in the order of P27/ANI7, …, P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 as analog inputs, start designing from P27/ANI17 (see 10.3 (6) A/D port configuration register (ADPC) for details).

<R> • 64-pin plastic FBGA (5 × 5)



Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
A1	USBM	C1	P53/TI00	E1	P13/TxD3	G1	AV_{REF}
A2	USBP	C2	P51/INTP2	E2	P12/SO00/TxD0	G2	P27/ANI7
A3	P71/KR1	C3	P52/TO00	E3	P110	G3	P25/ANI5
A4	P73/KR3	C4	P74/KR4/INTP8	E4	P111	G4	P23/ANI3
A5	P31/TI03/TO03/INTP4	C5	P76/KR6/INTP10	E5	P143/SI20/RxD2/ SDA20	G5	P21/ANI1
A6	P61/SDA0	C6	P77/KR7/INTP11	E6	P41/TOOL1	G6	P04/$\overline{SCK10}$/SCL10
A7	P60/SCL0	C7	V_{SS}	E7	$\overline{RESET}$	G7	P02/SO10/TxD1
A8	EV_{DD}	C8	P121/X1	E8	FLMD0	G8	P124/XT2
B1	P50/INTP1	D1	P14/RxD3	F1	P10/$\overline{SCK00}$	H1	AV_{SS}
B2	USBPUC	D2	P16/TI01/TO01/ INTP5	F2	P11/SI00/RxD0	H2	P26/ANI6
B3	P70/KR0	D3	P15/RTCDIV/RTCCL	F3	P130	H3	P24/ANI4
B4	P72/KR2	D4	P17/TI02/TO02	F4	P144/SO20/TxD2	H4	P22/ANI2
B5	P63	D5	P75/KR5/INTP9	F5	P142/$\overline{SCK20}$/SCL20	H5	P20/ANI0
B6	P62	D6	P40/TOOL0	F6	P43	H6	P03/SI10/RxD1/ SDA10
B7	V_{DD}	D7	REGC	F7	P42/TI04/TO04	H7	P140/PCLBUZ0/ INTP6
B8	EV_{SS}	D8	P122/X2/EXCLK	F8	P123/XT1	H8	P120/INTP0/EXLVI

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

2. Make EV_{DD} the same potential as V_{DD}.

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

1.5 Pin Identification

ANI0-ANI7 :Analog Input
AV_{REF} :Analog Reference Voltage
AV_{SS} :Analog Ground
EV_{DD} :Power Supply for Port
EV_{SS} :Ground for Port
EXLVI :External potential Input for Low-voltage detector
FLMD0 :Flash Programming Mode
INTP0-INTP2, INTP4-INTP6, INTP8-INTP11 :External Interrupt Input
KR0-KR7 :Key Return
PCLBUZ0 :Clock out put/ Buzzer output
P02-P04 :Port 0
P10-P17 :Port 1
P20-P27 :Port 2
P31 :Port 3
P40-P43 :Port 4
P50-P53 :Port 5
P60-P63 :Port 6
P70-P77 :Port 7
P110, P111 :Port 11
P120-P124 :Port 12
P130 :Port 13
P140-P144 :Port 14
REGC :Regulator Capacitance
$\overline{RESET}$:Reset
RTCCL :Real-time Counter Clock (32 kHz Original Oscillation) Output
RTCDIV :Real-time Counter Clock (32 kHz Divided Frequency) Output
RxD0-RxD3 :Receive Data
$\overline{SCK00}$, $\overline{SCK10}$, $\overline{SCK20}$: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-TI04 :Timer Input
TO00-TO04 :Timer Output
TOOL0 :Data Input/Output for Tool
TOOL1 :Clock Output for Tool
TxD0-TxD3 :Transmit Data
USBP :USB Data I/O (+)
USBM :USB Data I/O (-)
USBPUC :USB Pull-up Resistor Control
V_{DD} :Power Supply
V_{SS} :Ground
X1, X2 :Crystal Oscillator (Main SystemClock)
XT1, XT2 :Crystal Oscillator (SubsystemClock)

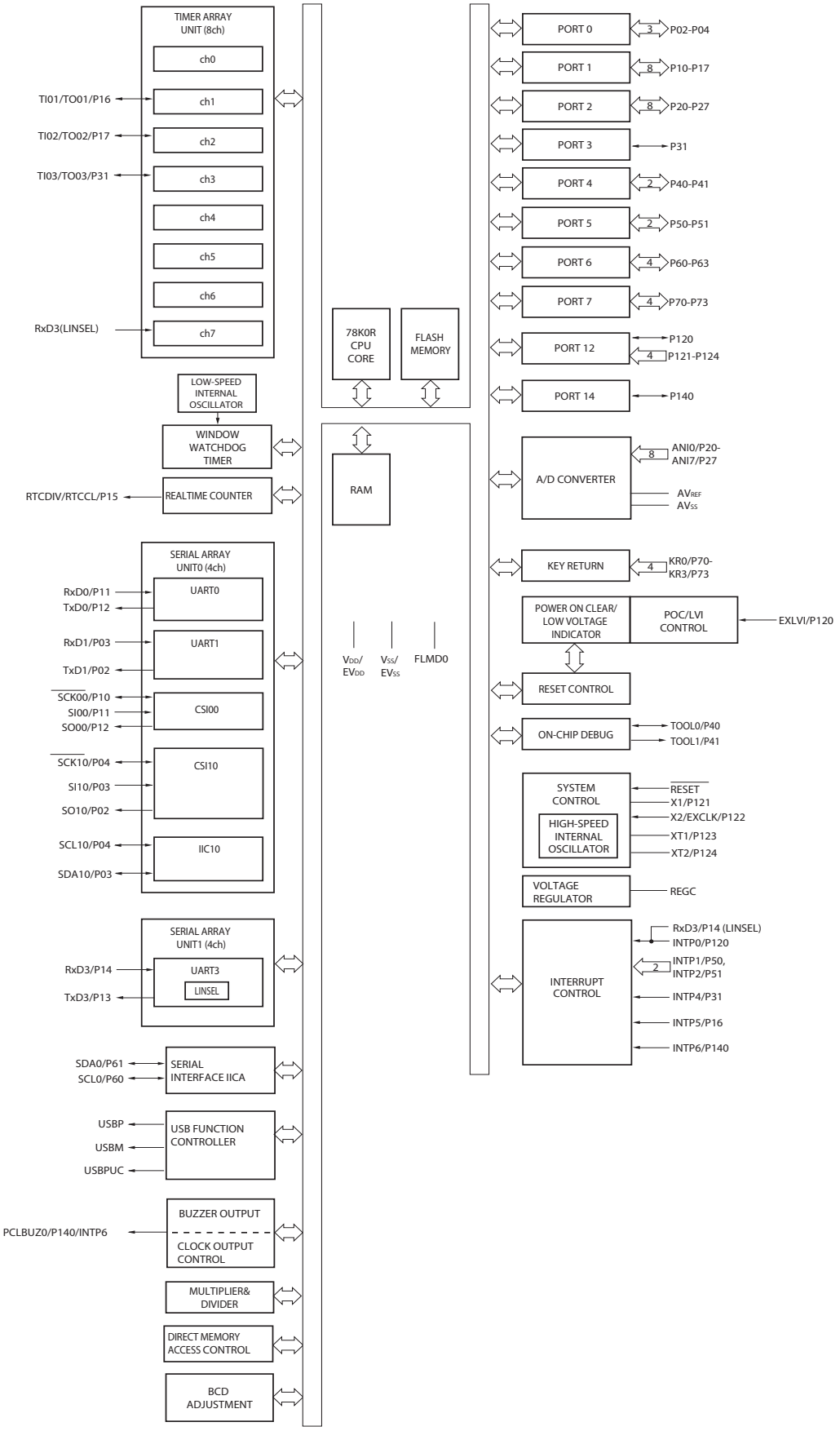
1.6 78K0R/KX3-L Microcontroller Lineup

	ROM	RAM	78K0R/KC3-L	78K0R/KE3-L
			48 pin	64 pin
<R>	128 KB	8 KB[note]	μPD78F1024	μPD78F1026
<R>	96 KB	8 KB[note]	μPD78F1023	μPD78F1025
	96 KB	6 KB	–	–
<R>	64 KB	6 KB	μPD78F1022	–

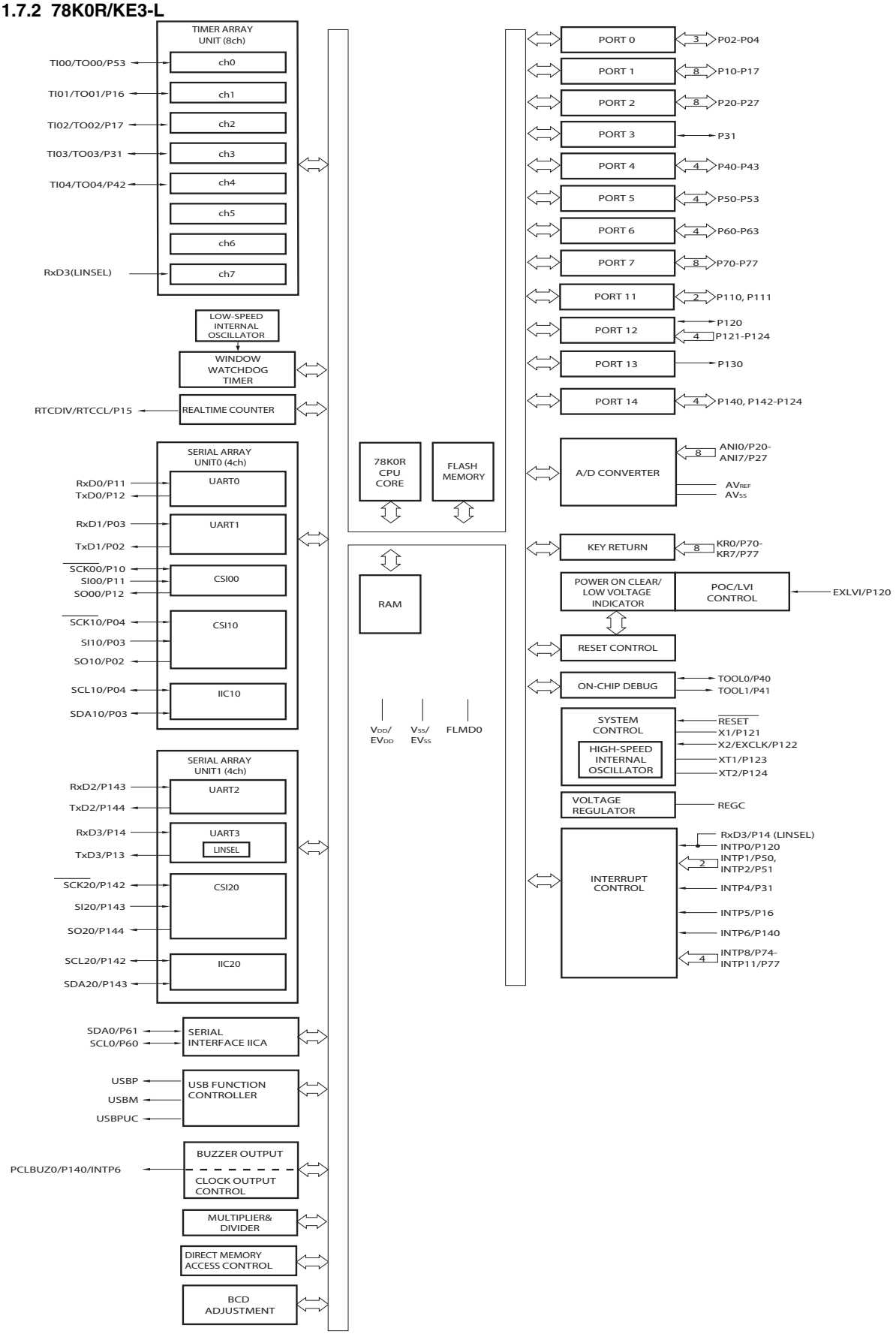
Note This is 7 KB when the self-programming function is used.

1.7 Block Diagram

1.7.1 78K0R/KC3-L



1.7.2 78K0R/KE3-L



1.8 Outline of Functions

(1/2)

<R>

Item		78K0R/KC3-L			78K0R/KE3-L	
		μPD78F1022	μPD78F1023	μPD78F1024	μPD78F1025	μPD78F1026
Internal Memory	Flash memory (self-programming supported)	64 KB	96 KB	128 KB	96 KB	128 KB
	RAM	6 KB	8 KB[note1]		8 KB[note1]	
Memory space		1 MB				
Main system clock (Oscillation frequency)	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 20 MHz: V_{DD} = 2.7 to 3.6 V, 2 to 5 MHz: V_{DD} = 1.8 to 3.6V				
	Internal high-speed oscillation clock	Internal oscillation 1 MHz ±13%, 8 MHz ±1.8%: V_{DD} = 1.8 to 3.6 V				
	20 MHz internal high-speed oscillation clock	Internal oscillation 20 MHz ±2.4%: V_{DD} = 2.7 to 3.6 V				
USB Clock		48 MHz (Divide, multiply and generate High speed system clock 12 MHz, 16 MHz, 20 MHz(V_{DD} = 3.0 to 3.6 V)				
Subsystem clock (Oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 1.8 to 3.6 V				
Low speed internal oscillation clock(WDT specific)		Internal Oscillation 30 kHz(TYP.):V_{DD} = 1.8 to 3.6 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)				
		61 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)				
Instruction set		• 8-bit operation, 16-bit operation • Multiplication (8 bits × 8 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.				
I/O PORT	Total	39			53	
	CMOS I/O	30			43	
	CMOS Input	4			4	
	CMOS Output	–			1	
	N-ch O.DI/O (6 V Tolerance)	4			4	
	USB for Buffer control	1			1	
Timer		• 16 Bit timer :8Channel • Watchdog Timer :1Channel • Real-time counter :1Channel				
	Timer Output	3(PWM Output:2[note2])			5(PWM Output:4[note2])	
	RTC Output	1 • 512 Hz or 16.384 kHz or 32.768 kHz(Subsystem clock:f_{SUB} = 32.768 kHz)				

Notes 1. This is 7 KB when the self-programming function is used.

2. The number of outputs varies, depending on the setting.

(2/2)

Item		78K0R/KC3-L			78K0R/KE3-L	
<R>		μPD78F1022	μPD78F1023	μPD78F1024	μPD78F1025	μPD78F1026
Clock Output/Buzzer Output		1 • 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		10-bit resolution × 8 channels (AV_{REF} = 1.8 to 3.6 V)				
Serial interface	SAU0	• CSI:1Channel/UART:1Channel • CSI:1Channel/UART:1Channel/simplifiedI^2C:1Channel			• CSI:1Channel/UART:1Channel • CSI:1Channel/UART:1Channel/simplified I^2C:1Channel	
	SAU1	• UART(LIN-bus supported):1Channel			• CSI:1Channel/UART:1Channel/simplified I^2C:1Channel • UART(LIN-bus supported):1Channel	
	IICA	• I^2C Bus:1Channel				
Multiplier/divider		• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 32 bits = 32 bits (division)				
USB Function controller		USB2.0 Full speed function controller:1ch				
DMA Controller		2Channel				
Vectored interrupt sources	Internal	36			41	
	External	7			11	
Key Interrupt		4Channel(KR0-KR3)			8Channel(KR0-KR7)	
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 • Internal reset by a reset processing check error				
<R> Power-on-clear circuit		• Power-on-reset: 1.61 ±0.09 V • Power-down-reset: 1.59 ±0.09 V				
Low-voltage detector		1.91 V to 3.45 V (11 stages)				
On-chip debug function		Provided				
Power supply voltage		When USB is not in use V_{DD} = 1.8 to 3.6 V, When USB is in use V_{DD} = 3.0 to 3.6 V				
Operating ambient temperature		T_A = −40 to +85 °C				
Package		48 pin LQFP(7 x 7)(0.50 mm pitch)			64 pin TQFP(7 x 7) (0.40 mm pitch) 64 pin LQFP(10 x 10) (0.50 mm pitch) 64 pin FBGA(5 x 5) (0.50 mm pitch)	

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 2 PIN FUNCTIONS

2.1 Pin Function List

There are three types of pin I/O buffer power supplies: AV_{REF}, 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_{REF}	P20 - P27
EV_{DD}	• Port pins other than P20 - P27 • $\overline{RESET}$ and FLMD0 pins
V_{DD}	• P121 - P124 • Pins other than port pins (excluding $\overline{RESET}$ and FLMD0 pins)

2.1.1 78K0R/KC3-L

(1) Port functions (1/2): 78K0R/KC3-L

Function Name	I/O	Function	After Reset	Alternate Function
P02	I/O	Port 0. 3-bit I/O port. Input of P03, P04 can be set to TTL input buffer. Output of P02 - P04 can be set to N-ch open-drain output (V_{DD} tolerance). 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	SO10/TxD1
P03				SI10/RxD1/SDA10
P04				$\overline{SCK10}$/SCL10
P10	I/O	Port 1. 8-bit I/O port. Input of P10, P11 can be set to TTL input buffer. Output of P10, P12 can be set to N-ch open-drain output (V_{DD} tolerance). 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	$\overline{SCK00}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
P20 - P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0 - ANI7
P31	I/O	Port 3. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI03/TO03/INTP4
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
P50	I/O	Port 5. 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	INTP1
P51				INTP2
P60		Port 6. 4-bit I/O port. Output of P60 - P63 can be set to N-ch open-drain output (6V tolerance). Input/output can be specified in 1-bit units.		SCL0
P61				SDA0
P62, P63				-

Note If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in **2.2.5 P40 - P47 (Port 4)**.

(1) Port functions (2/2): 78K0R/KC3-L

P70- P73	I/O	Port 7. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR0-KR3
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. 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
P140	I/O	Port 14. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCLBUZ0/INTP6

(2) Non-port functions (1/2): 78K0R/KC3-L

Function Name	I/O	Function	After Reset	Alternate Function
ANI0 - ANI7	Input	A/D converter analog input	Digital input port	P20 - P27
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
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				P50
INTP2				P51
INTP4				P31/TI03/TO03
INTP5				P16/TI01/TO01
INTP6				P140/PCLBUZ0
KR0 - KR3	Input	Key interrupt input	Input port	P70 - P73
PCLBUZ0	Output	Clock output/buzzer output	Input port	P140/INTP6
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	P15/RTCCL
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P15/RTCDIV
$\overline{\text{RESET}}$	Input	System reset input	–	–
RxD0	Input	Serial data input to UART0	Input port	P11/SI00
RxD1	Input	Serial data input to UART1	Input port	P03/SI10/SDA10
RxD3	Input	Serial data input to UART3	Input port	P14
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00, CSI10, CSI20	Input port	P10
$\overline{\text{SCK10}}$				P04/SCL10
SCL0	I/O	Clock input/output for I^2C	Input port	P60
SCL10	I/O	Clock input/output for simplified I^2C	Input port	P04/$\overline{\text{SCK10}}$
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	P03/SI10/RxD1
SI00	Input	Serial data input to CSI00, CSI10	Input port	P11/RxD0
SI10				P03/RxD1/SDA10

(2) Non-port functions (2/2): 78K0R/KC3-L

Function Name	I/O	Function	After Reset	Alternate Function
SO00	Output	Serial data output from CSI00 and CSI10	Input port	P12/TxD0
SO10				P02/TxD1
TI01	Output	External count clock input to 16-bit timer 01		P16/TO01/INTP5
TI02		External count clock input to 16-bit timer 02		P17/TO02
TI03		External count clock input to 16-bit timer 03		P31/TO03/INTP4
TO01	Output	16-bit timer 01 output	Input port	P16/TI01/INTP5
TO02		16-bit timer 02 output		P17/TI02
TO03		16-bit timer 03 output		P31/TI03/INTP4
TxD0	Output	Serial data output from UART0	Input port	P12/SO00
TxD1		Serial data output from UART1		P02/SO10
TxD3		Serial data output from UART3		P13
USBM	I/O	USB Data Input Output (-)	Input port	-
USBP	I/O	USB Data Input Output (+)	Input port	-
USBPUC	Output	USB Pull-up resistor control pin.	Low level Output	-
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 (P121 - P124 and other than ports (excluding RESET and FLMD0 pins))	–	–
EV_{DD}	–	Positive power supply for ports (other than P20- P27, P121- P124) and RESET and FLMD0 pins	–	–
AV_{REF}	–	• A/D converter reference voltage input • Positive power supply for P20-P27 and A/D converter	–	–
V_{SS}	–	Ground potential (P121 - P124 and other than ports (excluding RESET and FLMD0 pins))	–	–
EV_{SS}	–	Ground potential for ports (other than P20-P27, P121- P124) and RESET and FLMD0 pins	–	–
AV_{SS}	–	Ground potential for A/D converter, P20- P27. Use this pin with the same potential as EV_{SS} and V_{SS}.	–	–
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. 1. 2 78K0R/KE3-L

(1)Port functions (1/2):78K0R/KE3-L

Function Name	I/O	Function	After Reset	Alternate Function
P02	I/O	Port 0. 3-bit I/O port. Input of P03, P04 can be set to TTL input buffer. Output of P02 - P04 can be set to N-ch open-drain output (V_{DD} tolerance). 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	SO10/TxD1
P03				SI10/RxD1/SDA10
P04				$\overline{SCK10}$/SCL10
P10	I/O	Port 1. 8-bit I/O port. Input of P10, P11 can be set to TTL input buffer. Output of P10, P12 can be set to N-ch open-drain output (V_{DD} tolerance). 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	$\overline{SCK00}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
P20-P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital Input Port	ANI0-ANI7
P31	I/O	Port 3. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input Port	TI03/TO03/INTP4
P40[Note]	I/O	Port 4. 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	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				–
P50	I/O	Port 5. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input Port	INTP1
P51				INTP2
P52				TO00
P53				TI00
P60	I/O	Port 6. 4-bit I/O port. Output of P60 - P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input Port	SCL0
P61				SDA0
P62, P63				–

Note If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in **2.2.5 P40 - P47 (port 4)**

(1)Port functions (2/2):78K0R/KE3-L

Function Name	I/O	Function	After Reset	Alternate Function
P70-P73	I/O	Port 7. 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	KR0-KR3
P74-P77				KR4/INTP8-KR7/INTP11
P110, P111	I/O	Port 11. 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	–
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. 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	I/O	Port 14. 4-bit I/O port. Input of P142, P143 can be set to TTL input buffer. Output of P142 - P144 can be set to the N-ch open-drain output (V_{DD} tolerance). 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	PCLBUZ0/INTP6
P142				$\overline{SCK20}$/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2

(2) Non-port Function(1/2):78K0R/KE3-L

Function Name	I/O	Function	After Reset	Alternate Function
ANI0 - ANI7	Input	A/D converter analog input	Digital input port	P20 - P27
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
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				P50
INTP2				P51
INTP4				P31/TI03/TO03
INTP5				P16/TI01/TO01
INTP6				P140/PCLBUZ0
INTP8 - INTP11				P74/KR4 - P77/KR7
KR0 - KR3	Input	Key interrupt input	Input port	P70 - P73
KR4 - KR7				P74/INTP8 - P77/INTP11
PCLBUZ0	Output	Clock output/buzzer output	Input port	P140/INTP6
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 - 1 μF).	–	–
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P15/RTCCL
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P15/RTCDIV
$\overline{\text{RESET}}$	Input	System reset input	–	–
RxD0	Input	Serial data input to UART0	Input port	P11/SI00
RxD1	Input	Serial data input to UART1	Input port	P03/SI10/SDA10
RxD2	Input	Serial data input to UART2	Input port	P143/SI20/SDA20
RxD3	Input	Serial data input to UART3	Input port	P14
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00, CSI10, CSI20	Input port	P10
$\overline{\text{SCK10}}$				P04/SCL10
$\overline{\text{SCK20}}$				P142/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	P04/$\overline{\text{SCK10}}$
SCL20				P142/$\overline{\text{SCK20}}$
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	P03/SI10/RxD1
SDA20				P143/SI20/RxD2
SI00	Input	Serial data input to CSI00, CSI10, CSI20	Input port	P11/RxD0
SI10				P03/RxD1/SDA10
SI20				P143/RxD2/SDA20

(2) Non-port Function(2/2):78K0R/KE3-L

Function Name	I/O	Function	After Reset	Alternate Function
SO00	Output	Serial data output from CSI00, CSI10, CSI20	Input Port	P12/TxD0
SO10				P02/TxD1
SO20				P144/TxD2
TI00	Input	External count clock input to 16-bit timer 00	Input Port	P53
TI01		External count clock input to 16-bit timer 01		P16/TO01/INTP5
TI02		External count clock input to 16-bit timer 02		P17/TO02
TI03		External count clock input to 16-bit timer 03		P31/TO03/INTP4
TI04		External count clock input to 16-bit timer 04		P42/TO04
TO00	Output	16-bit timer 00 output	Input Port	P52
TO01		16-bit timer 01 output		P16/TI01/INTP5
TO02		16-bit timer 02 output		P17/TI02
TO03		16-bit timer 03 output		P31/TI03/INTP4
TO04		16-bit timer 04 output		P42/TI04
TxD0	Output	Serial data output from UART0	Input Port	P12/SO00
TxD1		Serial data output from UART1		P02/SO10
TxD2		Serial data output from UART2		P144/SO20
TxD3		Serial data output from UART3		P13
USBM	I/O	USB Data Input Output (-)	Input Port	–
USBP	I/O	USB Data Input Output (+)	Input Port	–
USBPUC	Output	USB Pull-up resistor control pin	Low level Output	–
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 (P121 - P124 and other than ports (excluding $\overline{RESET}$ and FLMD0 pins))	–	–
EV_{DD}	–	Positive power supply for ports (other than P20- P27, P121- P124) and $\overline{RESET}$ and FLMD0 pins	–	–
AV_{REF}	–	• A/D converter reference voltage input • Positive power supply for P20-P27 and A/D converter	–	–
V_{SS}	–	Ground potential (P121 - P124 and other than ports (excluding $\overline{RESET}$ and FLMD0 pins))	–	–
EV_{SS}	–	Ground potential for ports (other than P20-P27, P121- P124) and $\overline{RESET}$ and FLMD0 pins	–	–
AV_{SS}	–	Ground potential for A/D converter, P20- P27. Use this pin with the same potential as EV_{SS} and V_{SS}.	–	–
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 P02 - P04 (port 0)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P02/SO10/TxD1	√	√
P03/SI10/RxD1/SDA10	√	√
P04/$\overline{\text{SCK10}}$/SCL10	√	√

It functions as an I/O port. These pins also function as timer I/O, serial interface data I/O, and clock I/O.

Input to the P03, P04 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units, using port input mode register 0 (PIM0).

Output from the P02 - P04 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 0 (POM0).

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

(1) Port mode

It functions as an I/O port. It 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) Control mode

It functions as timer I/O, serial interface data I/O, and clock I/O.

(a) SI10

This is a serial data input pin of serial interface CSI10.

(b) SO10

This is a serial data output pin of serial interface CSI10.

(c) $\overline{\text{SCK10}}$

This is a serial clock I/O pin of serial interface CSI10.

(d) TxD1

This is a serial data output pin of serial interface UART1.

(e) RxD1

This is a serial data input pin of serial interface UART1.

(f) SDA10

This is a serial data I/O pin of serial interface for simplified I^2C.

(g) SCL10

This is a serial clock I/O pin of serial interface for simplified I^2C.

Caution To use P02/SO10/TxD1, P04/$\overline{\text{SCK10}}$/SCL10 as general-purpose ports, set serial communication operation setting register 02 (SCR02) to the default status (0087H). In addition, clear port output mode register 0 (POM0) to 00H.

2.2.2 P10 - P17 (port 1)

	78K0R/KC3-L(48 Pin) (μ PD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P10/ SCK00	√	√
P11/SI00/RxD0	√	√
P12/SO00/TxD0	√	√
P13/TxD3	√	√
P14/RxD3	√	√
P15/RTCDIV/RTCCL	√	√
P16/TI01/TO01/INTP5	√	√
P17/TI02/TO02	√	√

It functions as an I/O port. These pins also function as external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and real-time counter clock output.

Input to the P10, P11 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, P12 pins can be specified as normal CMOS output or 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

It functions as an I/O port. It 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

It functions as an external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and real-time counter clock output.

(a) SI00

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

(b) SO00

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

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

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

(d) RxD0, RxD3

These are the serial data input pins of serial interface UART0, UART3.

(e) TxD0, TxD3

These are the serial data output pins of serial interface UART0, UART3.

(f) TI01, TI02

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 01, 02.

(g) TO01, TO02

These are the timer output pins of 16-bit timers 01, 02.

(h) INTP5

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.

(i) RTCDIV

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

(j) RTCCL

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

Cautions 1. To use P10/$\overline{\text{SCK00}}$, P12/SO00/TxD0 as general-purpose ports, set serial communication operation setting register 00 (SCR00) to the default status (0087H).

2. Do not enable outputting RTCCL and RTCDIV at the same time.

2.2.3 P20 - P27 (port 2)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μ PD78F102y:y = 5, 6)
P20/ANI0	√	√
P21/ANI1	√	√
P22/ANI2	√	√
P23/ANI3	√	√
P24/ANI4	√	√
P25/ANI5	√	√
P26/ANI6	√	√
P27/ANI7	√	√

It functions as an I/O port. These pins also function as A/D converter analog input.
The following operation modes can be specified in 1-bit units.

(1) Port mode

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

(2) Control mode

It functions as A/D converter analog input pins (ANI0 - ANI7). When using these pins as analog input pins, refer to **10.6 (5) ANI0/P20 - ANI7/P27**

Caution ANI0/P20 - ANI7/P27 is set in the digital input (general-purpose port) mode after release of reset.

2.2.4 P31 (port 3)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P31/TI03/TO03/INTP4	√	√

It functions as an I/O port. These pins also function as external interrupt request input, timer I/O, and real-time counter correction clock output.
The following operation modes can be specified in 1-bit units.

(1) Port mode

It functions as an I/O port. It 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

It functions as external interrupt request input, timer I/O, and real-time counter correction clock output.

(a) INTP3, INTP4

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

(b) TI03

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 03.

(c) TO03

This is a timer output pin from 16-bit timer 03.

(d) RTC1HZ

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

2.2.5 P40 - P43 (port 4)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P40/TOOL0	√	√
P41/TOOL1	√	√
P42/TI04/TO04	–	√
P43	–	√

It functions as an I/O port. These pins also function as serial interface data I/O, clock I/O, data I/O for a flash memory programmer/debugger, clock output, and timer I/O.

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

(1) Port mode

It functions as an I/O port. It 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

It functions as serial interface data I/O, clock I/O, data I/O for a flash memory programmer/debugger, clock output, and timer I/O.

(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).

(c) TI04

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 04.

(d) TO04

This is a timer output pins from 16-bit timer 04.

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 EV_{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 EV_{DD} via an external resistor.

2.2.6 P50 - P53 (port 5)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P50/INTP1	√	√
P51/INTP2	√	√
P52/TO00	–	√
P53/TI00	–	√

It functions as an I/O port. These pins also function as external interrupt request input, timer I/O, and clock/buzzer output.

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

(1) Port mode

It functions as an I/O port. It 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

It functions as external interrupt request input, timer I/O, and clock/buzzer output.

(a) INTP1, INTP2

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

(b) TI00

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 00.

(c) TO00

These are the timer output pins of 16-bit timers 00.

2.2.7 P60 - P63 (port 6)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μ PD78F102y:y = 5, 6)
P60/SCL0	√	√
P61/SDA0	√	√
P62	√	√
P63	√	√

It functions as an I/O port. These pins also function as serial interface data I/O, clock I/O, and timer I/O.
The following operation modes can be specified in 1-bit units.

(1) Port mode

It functions as an 8-bit I/O port. It can be set to input port or output port in 1-bit units using port mode register 6 (PM6).
Output of P60 - P63 is N-ch open-drain output (6 V tolerance).

(2) Control mode

It functions as serial interface data I/O, clock I/O, and timer I/O.

(a) SDA0

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

(b) SCL0

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

2.2.8 P70 - P77 (port 7)

	78K0R/KC3-L(48 Pin) (μ PD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μ PD78F102y:y = 5, 6)
P70/KR0	√	√
P71/KR1	√	√
P72/KR2	√	√
P73/KR3	√	√
P74/KR4/INTP8	–	√
P75/KR5/INTP9	–	√
P76/KR6/INTP10	–	√
P77/KR7/INTP11	–	√

It functions as an I/O port. These pins also function as key interrupt input and external interrupt request input.
The following operation modes can be specified in 1-bit units.

(1) Port mode

It functions as an I/O port. It can be set to input or output port in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

It functions as key interrupt input, and external interrupt request input.

(a) KR0 - KR7

These are the key interrupt input pins.

(b) INTP8 - INTP11

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

2.2.9 P110, P111 (port 11)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P110	–	√
P111	–	√

It functions as an I/O port.

It can be set to input or output port in 1-bit units using port mode register 11 (PM11). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 11 (PU11).

2.2.10 P120 - P124 (port 12)

P120 function as a 1-bit I/O port. P121 - P124 functions as a 4-bit input port. These pins also function as 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.

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

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P120/INTP0/EXLVI	√	√
P121/X1	√	√
P122/X2/EXCLK	√	√
P123/XT1	√	√
P124/XT2	√	√

(1) Port mode

P120 functions as a 1-bit I/O port. It can be set to input 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 - P124 functions as a 4-bit input port.

(2) Control mode

It functions as 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.

(a) 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.

(b) EXLVI

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

(c) X1, X2

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

(d) EXCLK

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

(e) XT1, XT2

These are the pins for connecting a resonator for subsystem clock.

2.2.11 P130 (port 13)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P130	–	√

It functions as an output port.

Remark When the device is reset, P130 outputs a low level. Therefore, to output a high level from P130 before the device is reset, the output signal of P130 can be used as a pseudo reset signal of the CPU (see the figure for **Remark** in **4.2.12 Port 13**).

2.2.12 P140, P142 - P144 (port 14)

	78K0R/KC3-L(48 Pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64 Pin) (μPD78F102y:y = 5, 6)
P140/PCLBUZ0/INTP6	√	√
P142/$\overline{\text{SCK20}}$/SCL20	–	√
P143/SI20/RxD2/SDA20	–	√
P144/SO20/TxD2	–	√

It functions as an I/O port. These pins also function as external interrupt request input, clock/buzzer output, serial interface data I/O, and clock I/O.

Input to the P142, P143 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 14 (PIM14).

Output from the P142 - P144 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 14 (POM14).

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

(1) Port mode

It functions as an I/O port. It 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) Control mode

It functions as external interrupt request input, clock/buzzer output, and serial interface data I/O, and clock I/O.

(a) INTP6

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.

(b) PCLBUZ0

This is a clock/buzzer output pin.

(c) SI20

This is a serial data input pin of serial interface CSI20.

(d) SO20

This is a serial data output pin of serial interface CSI20.

(e) $\overline{\text{SCK20}}$

This is a serial clock I/O pin of serial interface CSI20.

(f) TxD2

This is a serial data output pin of serial interface UART2.

(g) RxD2

This is a serial data input pin of serial interface UART2.

(h) SDA20

This is a serial data I/O pin of serial interface for simplified I^2C.

(i) SCL20

This is a serial clock I/O pin of serial interface for simplified I^2C.

2.2.13 AV_{REF}

This is the A/D converter reference voltage input pin and the positive power supply pin of P20 - P27 and A/D converter.

When P20/AN10-P27/AN17 are used as the analog port pins, make the potential of AV_{REF} be such that 1.8 V ≤ AV_{REF} ≤ V_{DD}. When either one of P20/AN10-P27/AN17are used as the digital port or when the A/D converter is not used, make AV_{REF} the same potential as EV_{DD} or V_{DD}.

2.2.14 AV_{SS}

This is the ground potential pin of A/D converter, P20 - P27. Even when the A/D converter is not used, always use this pin with the same potential as EV_{SS} and V_{SS}.

2.2.15 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

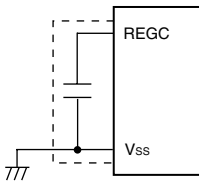
When the external reset pin is not used, connect this pin directly or via a resistor to EV_{DD}.

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

2.2.16 REGC

This is the pin for connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect this pin <R> to V_{SS} via a capacitor (0.47 - 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.

2.2.17 V_{DD}, EV_{DD}

V_{DD} is the positive power supply pin for P121 - P124 and pins other than ports (excluding the $\overline{\text{RESET}}$ and FLMD0 pins).

EV_{DD} is the positive power supply pin for ports other than P20 - P27, P121 - P124 as well as for the $\overline{\text{RESET}}$ and FLMD0 pins.

2.2.18 V_{SS}, EV_{SS}

V_{SS} is the ground potential pin for P121 - P124 and pins other than ports (excluding the $\overline{\text{RESET}}$, FLMD0 pins).

EV_{SS} is the ground potential pin for ports other than P20 - P27, P121 - P124 as well as for the $\overline{\text{RESET}}$ and FLMD0 pins.

2.2.19 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 **23.5 (1) Back ground event control register**). To externally pull it down, 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 - 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 - 200 kΩ.

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. See **Figure 2-1. Pin I/O Circuit List** for the I/O circuit of each type.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P02/SO10/TxD1	5-AG	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P03/SI10/RxD1/SDA10	5-AN		
P04/$\overline{SCK10}$/SCL10			
P10/$\overline{SCK00}$			
P11/SI00/RxD0			
P12/SO00/TxD0	5-AG		
P13/TxD3			
P14/RxD3	8-R		
P15/RTCDIV/RTCCL	5-AG		
P16/TI01/TO01/INTP5	8-R		
P17/TI02/TO02			
P20/ANI0 - P27/ANI7[Note]	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
P31/TI03/TO03/INTP4	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P40/TOOL0			<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.
P42/TI04/TO04	8-R		
P43			
P50/INTP1			
P51/INTP2			
P52/TO00	5-AG		
P53/TI00	8-R		

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

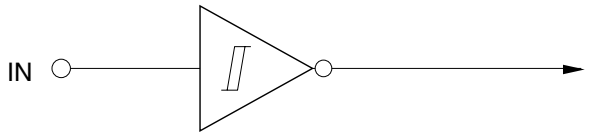
Table 2-2. Connection of Unused Pins (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P60/SCL0	13-R	I/O	Input: Connect to EV_{SS}. Output: Set the port output latch to 0 and leave these pins open via low-level output.
P61/SDA0			
P62, P63	13-P		
P70/KR0 - P73//KR3	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P74/KR4/INTP8 - P77/KR7/INTP11			
P110	8-R		
P111	5-AG		
P120/INTP0/EXLVI	8-R		
P121/X1[Note]	37-B	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK[Note]			
P123/XT1[Note]			
P124/XT2[Note]			
P130	3-C	Output	Leave open.
P140/PCLBUZ0/ITNP6	8-R	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P142/$\overline{SCK20}$/SCL20	5-AN		
P143/SI20/RxD2/SDA20			
P144/SO20/TxD2	5-AG		
AV_{REF}	–	–	<When one or more of P20 - P27 are set as a digital port> Make this pin the same potential as EV_{DD} or V_{DD}. <When all of P20 - P27 are set as analog ports> Make this pin to have a potential where 1.8 V ≤ AV_{REF} ≤ V_{DD}.
AV_{SS}	–	–	Make this pin the same potential as EV_{SS} or V_{SS}.
FLMD0	2-W	–	Leave open or connect to V_{SS} via a resistor of 100 kΩ or more.
$\overline{RESET}$	2	Input	Connect directly or via a resistor to EV_{DD}.
<R> REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).
USBP	USB	I/O	Connect to EV_{SS}.
USBM	USB	I/O	Connect to EV_{SS}.
USBPUC	3-C	Output	Leave open.

Notes 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.

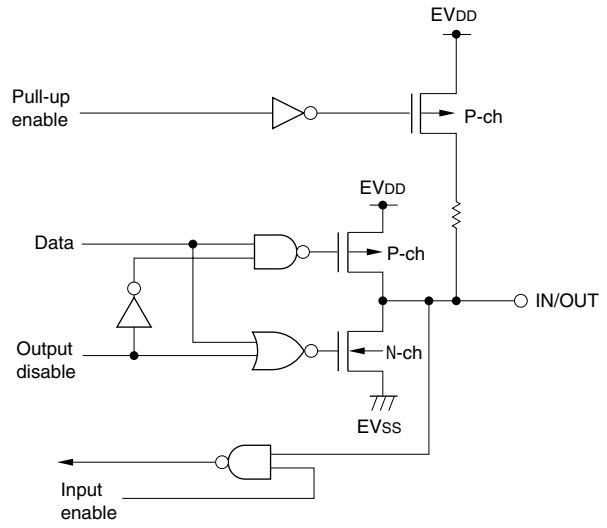
Figure 2-1. Pin I/O Circuit List (1/2)

Type 2

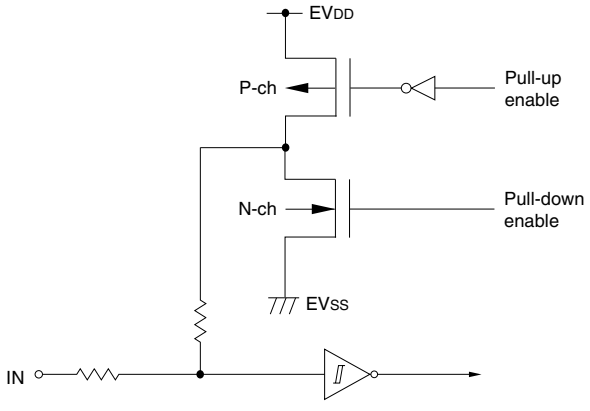


Schmitt-triggered input with hysteresis characteristics

Type 5-AG

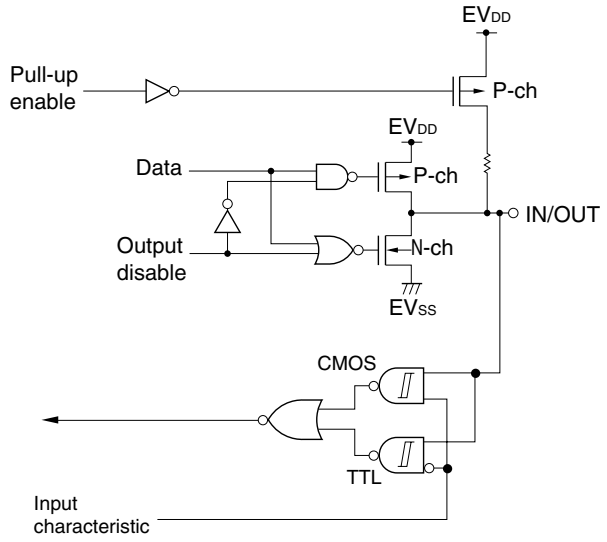


Type 2-W

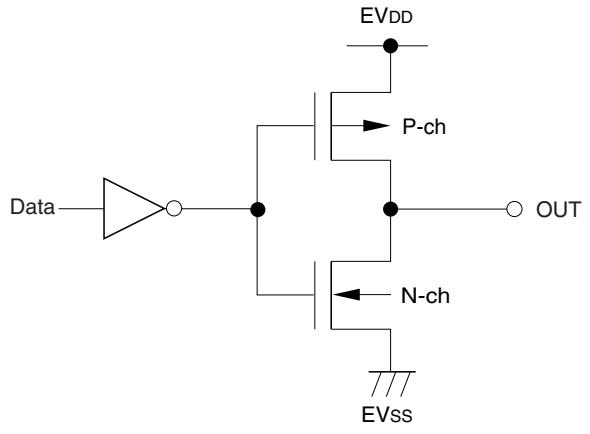


Schmitt-triggered input with hysteresis characteristics

Type 5-AN



Type 3-C



Type 8-R

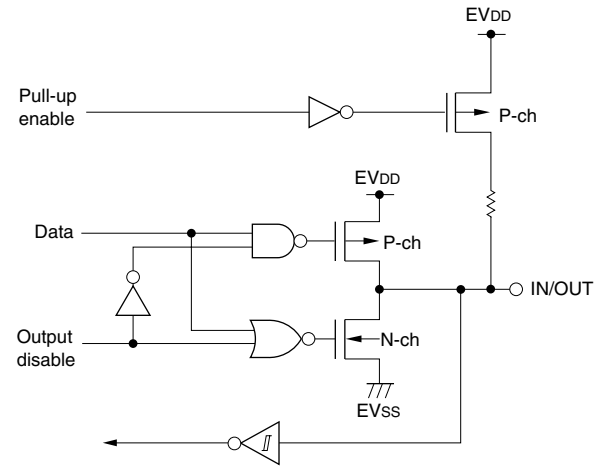
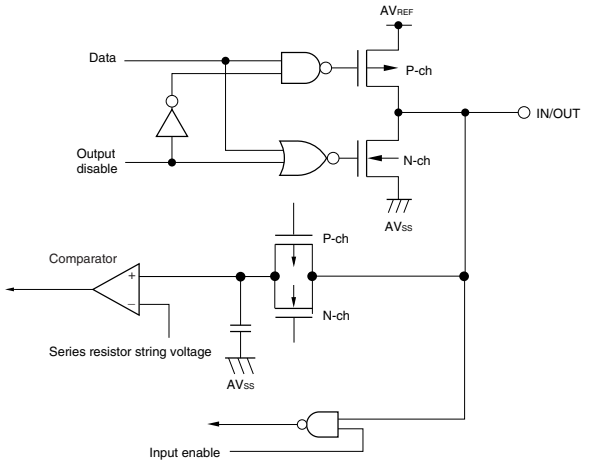
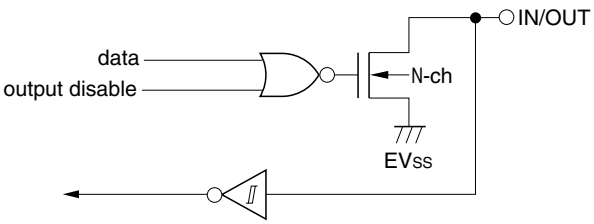


Figure 2-1. Pin I/O Circuit List (2/2)

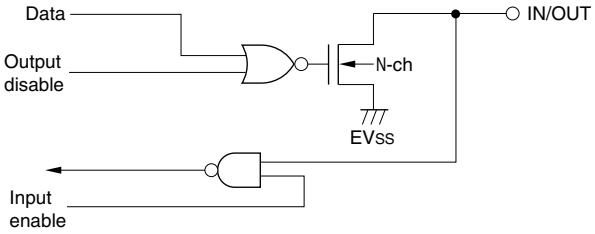
Type 11-G



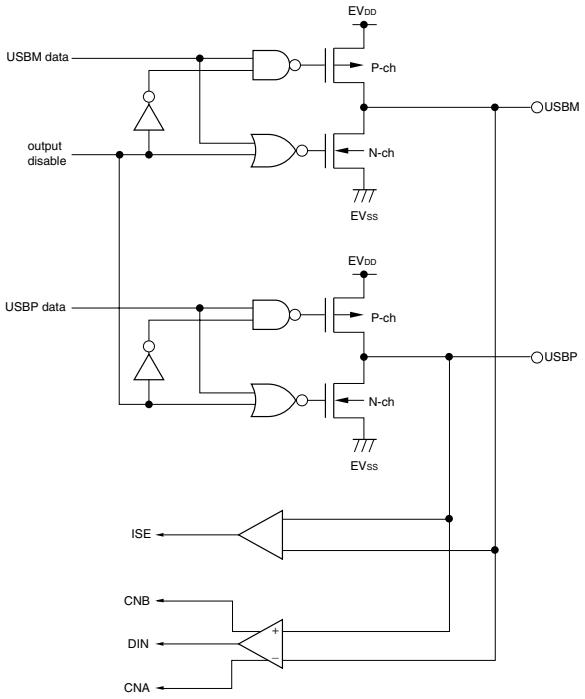
Type 13-R



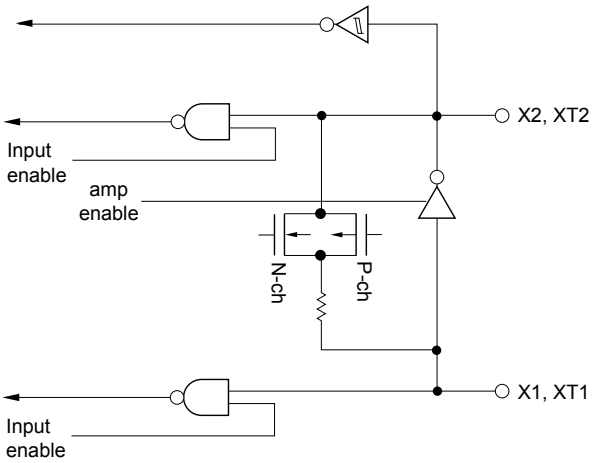
Type 13-P



USB



Type 37-B

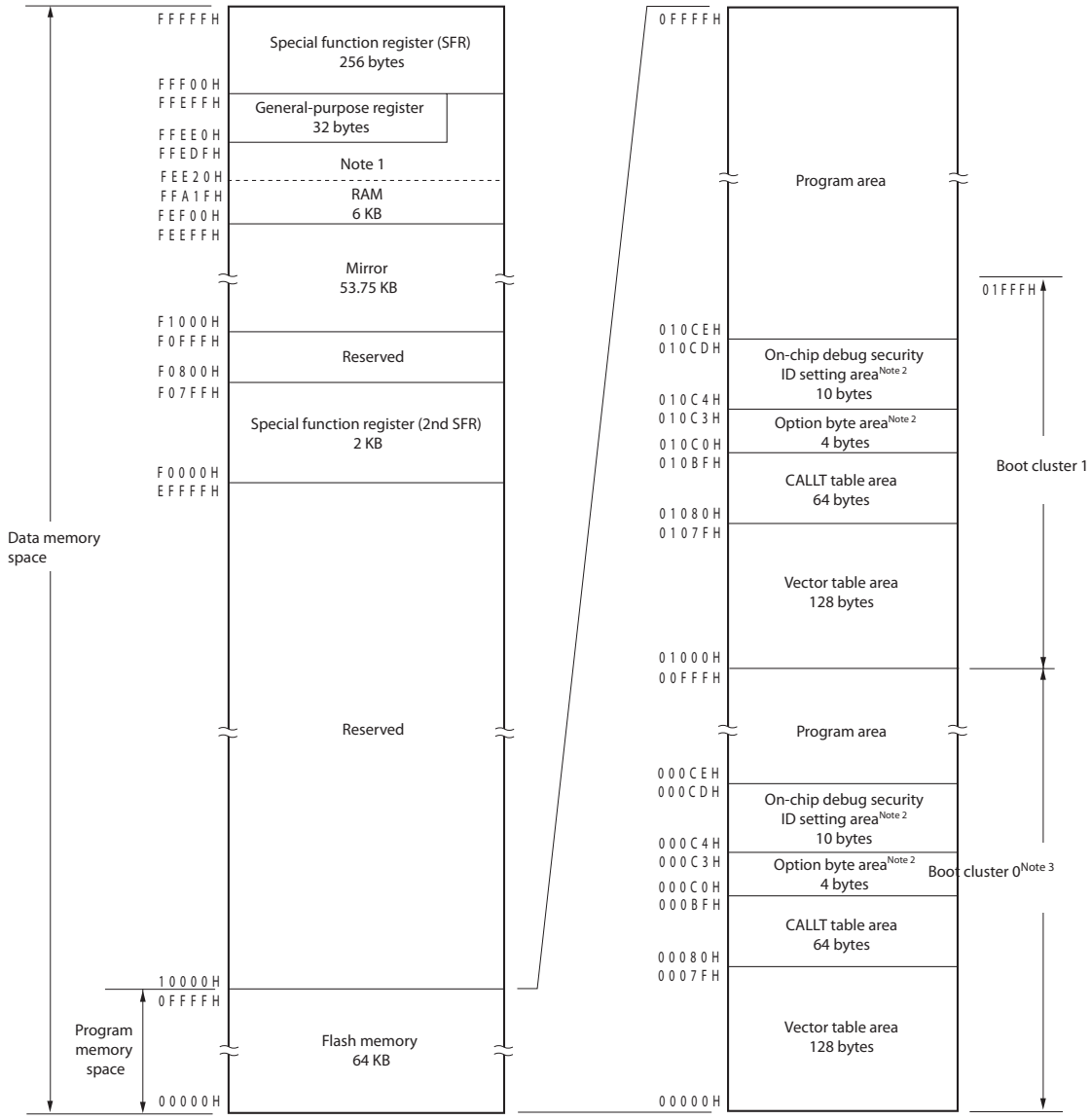


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the **78K0R/KC3-L, 78K0R/KE3-L** can access a 1 MB memory space. Figures 3-1 - 3-3 show the memory maps.

Figure 3-1 . Memory Map (μPD78F1022)



Note. 1. Use of the area FFE20H - FFEDFH is prohibited when using the self-programming function, since this area is used for self-programming library.

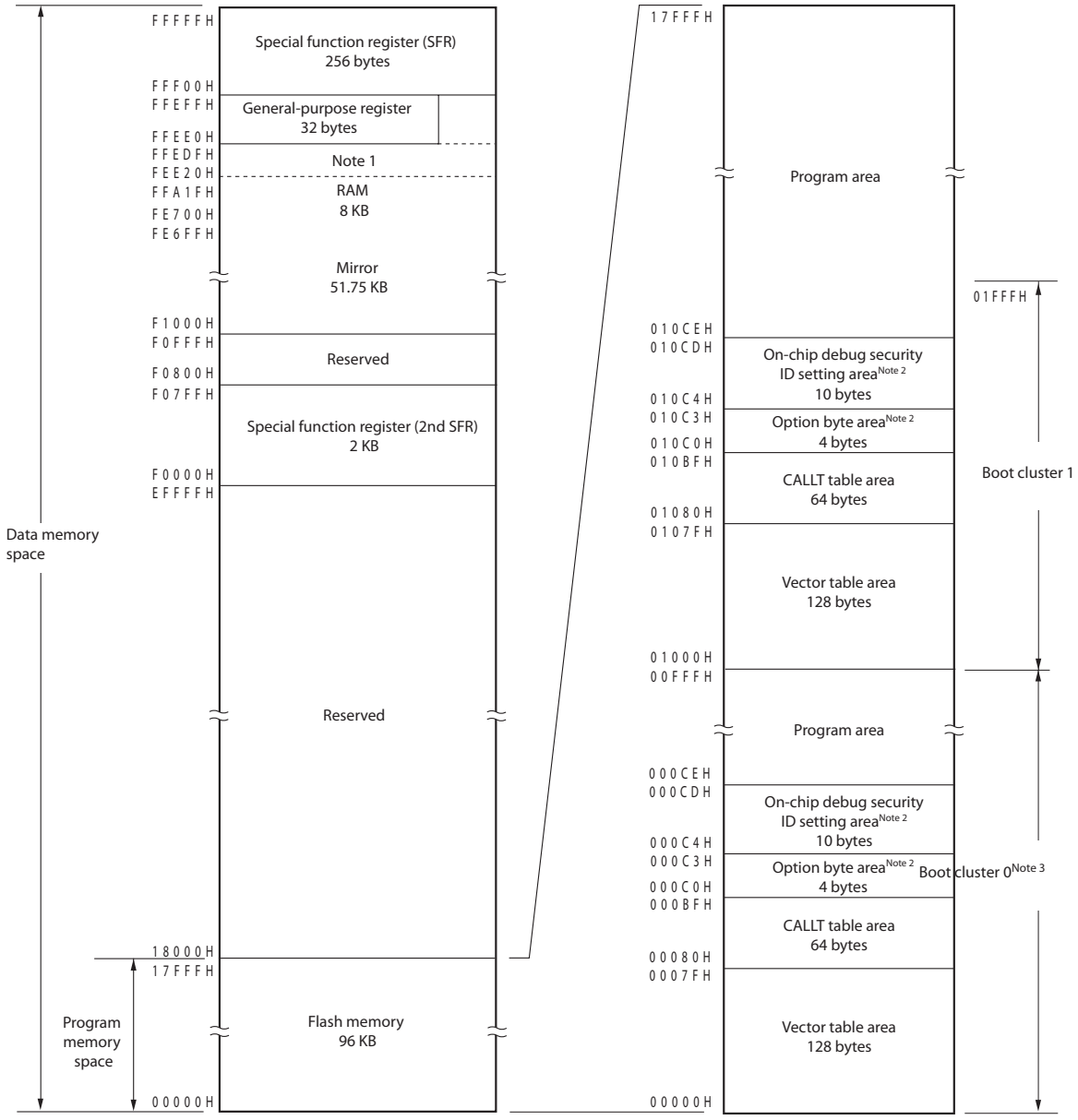
2. When boot swap is not used: Set the option bytes to 000C0H - 000C3H, and the on-chip debug security IDs to 000C4H - 000CDH.

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

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.7 Security Setting**).

<R> **Remark** Instructions can be executed from the RAM area excluding the general-purpose register area.

Figure 3-2 . Memory Map (μPD78F1023, 78F1025)



Notes 1. Use of the area FFE20H - FFEDFH is prohibited when using the self-programming function, since this area is used for self-programming library.

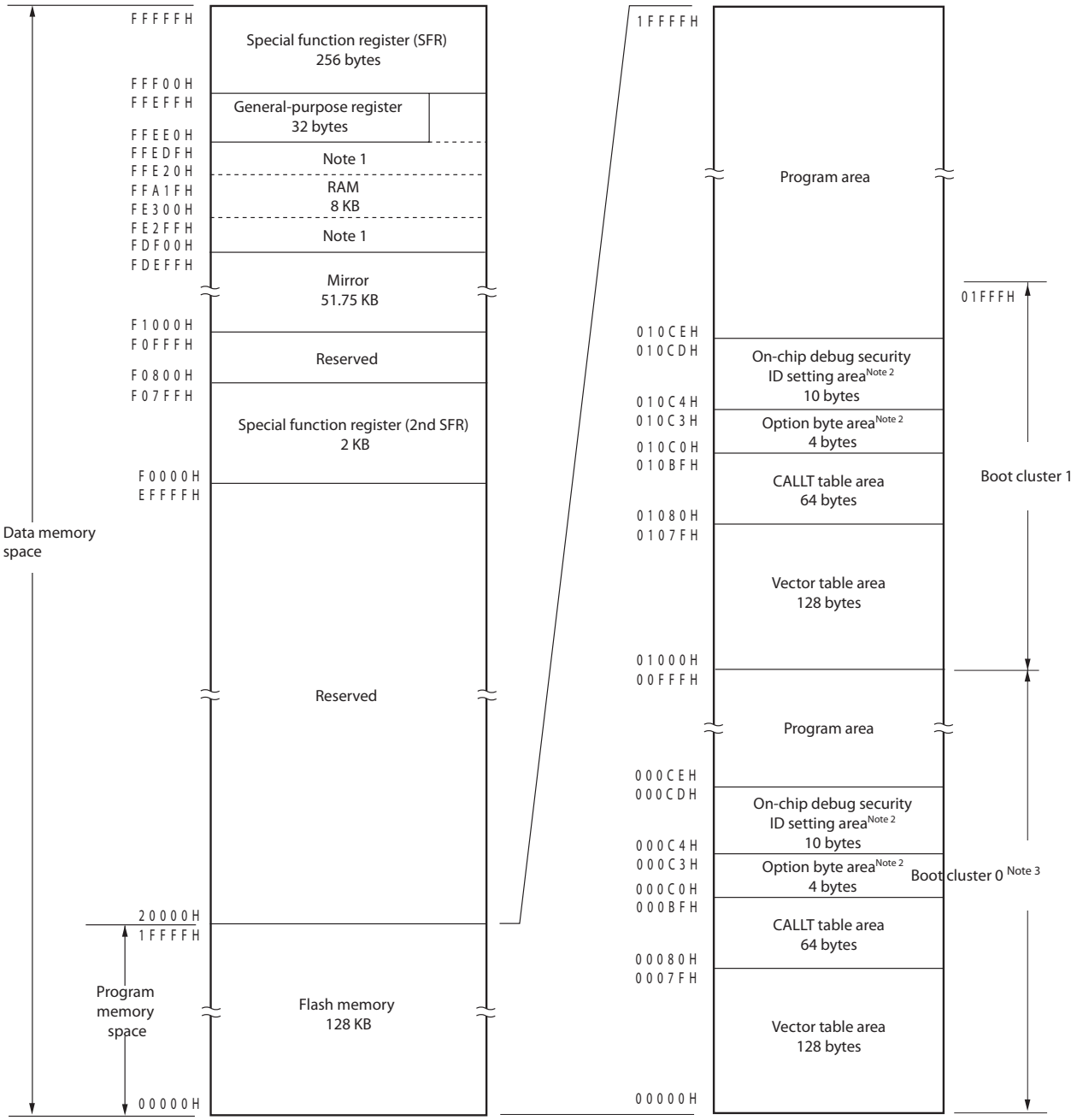
2. When boot swap is not used: Set the option bytes to 000C0H - 000C3H, and the on-chip debug security IDs to 000C4H - 000CDH.

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

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.7 Security Setting**).

<R> **Remark** Instructions can be executed from the RAM area excluding the general-purpose register area.

Figure 3-3. Memory Map (μPD78F1024, 78F1026)



Notes 1. Use of the area FFE20H - FFEDFH and FDF00H - FE2FFH are prohibited when using the self-programming function, since this area is used for self-programming library.

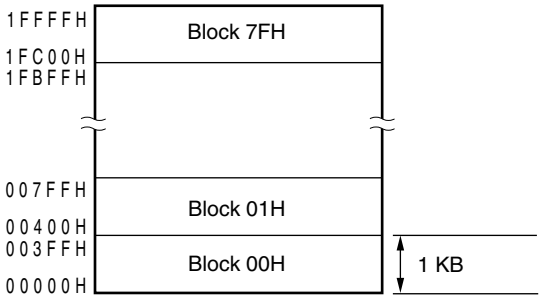
2. When boot swap is not used: Set the option bytes to 000C0H - 000C3H, and the on-chip debug security IDs to 000C4H - 000CDH.

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

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.7 Security Setting**).

<R> **Remark** Instructions can be executed from the RAM area excluding the general-purpose register area.

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

Remark μPD78F1022: Block numbers 00H - 3FH
μPD78F1023, 78F1025: Block numbers 00H - 5FH
μPD78F1024, 78F1026: Block numbers 00H - 7FH

3.1.1 Internal program memory space

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

78K0R/KC3-L, KE3-L products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F1022	Flash memory	65536 × 8 bits (00000H - 0FFFFH)
μPD78F1023, 78F1025		98304 × 8 bits (00000H - 17FFFH)
μPD78F1024, 78F1026		131072 × 8 bits (00000H - 1FFFFH)

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

(1) Vector table area

The 128-byte area 00000H - 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 - 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	0002EH	INTTM01
		00030H	INTTM02
00002H	INTDBG	00032H	INTTM03
00004H	INTWDTI	00034H	INTAD
00006H	INTLVI	00036H	INTRTC
00008H	INTP0	00038H	INTRTCI
0000AH	INTP1	0003AH	INTKR
0000CH	INTP2	0003CH	INTST2/INTCSI20/INTIIC20[note]
00010H	INTP4	0003EH	INTP6
00012H	INTP5	00042H	INTTM04
00014H	INTST3	00044H	INTTM05
00016H	INTSR3	00046H	INTTM06
00018H	INTSRE3	00048H	INTTM07
0001AH	INTDMA0	0004AH	INTSR2[note]
0001CH	INTDMA1	0004EH	INTP8[note]
0001EH	INTST0/INTCSI00	00050H	INTP9[note]
00020H	INTSR0	00052H	INTP10[note]
00022H	INTSRE0	00054H	INTP11[note]
00024H	INTST1/INTCSI10/INTIIC10	0005CH	INTSRE2[note]
00026H	INTSR1	0005EH	INTSUB
00028H	INTSRE1	00060H	INTRSUM
0002AH	INTIICA	0007EH	BRK
0002CH	INTTM00		

Note **78K0R/KE3-L** Only

(2) CALLT instruction table area

The 64-byte area 00080H - 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 - 0FFFFH (because an address code is of 2 bytes).
To use the boot swap function, set a CALLT instruction table also at 01080H - 010BFH.

(3) Option byte area

A 4-byte area of 000C0H - 000C3H can be used as an option byte area. Set the option byte at 010C0H - 010C3H when the boot swap is used. For details, see **CHAPTER 22 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H - 000CDH and 010C4H - 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 - 000CDH when the boot swap is not used and at 000C4H - 000CDH and 010C4H - 010CDH when the boot swap is used. For details, see **CHAPTER 24 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The μPD78F1022 mirror the data flash area of 00000H - 0FFFFH, F0000H - FFFFFH.

The μPD78F1023, 78F1024, 78F1025 and 78F1026 mirror the data flash area of 00000H - 0FFFFH or 10000H - 1FFFFH or F0000H - FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

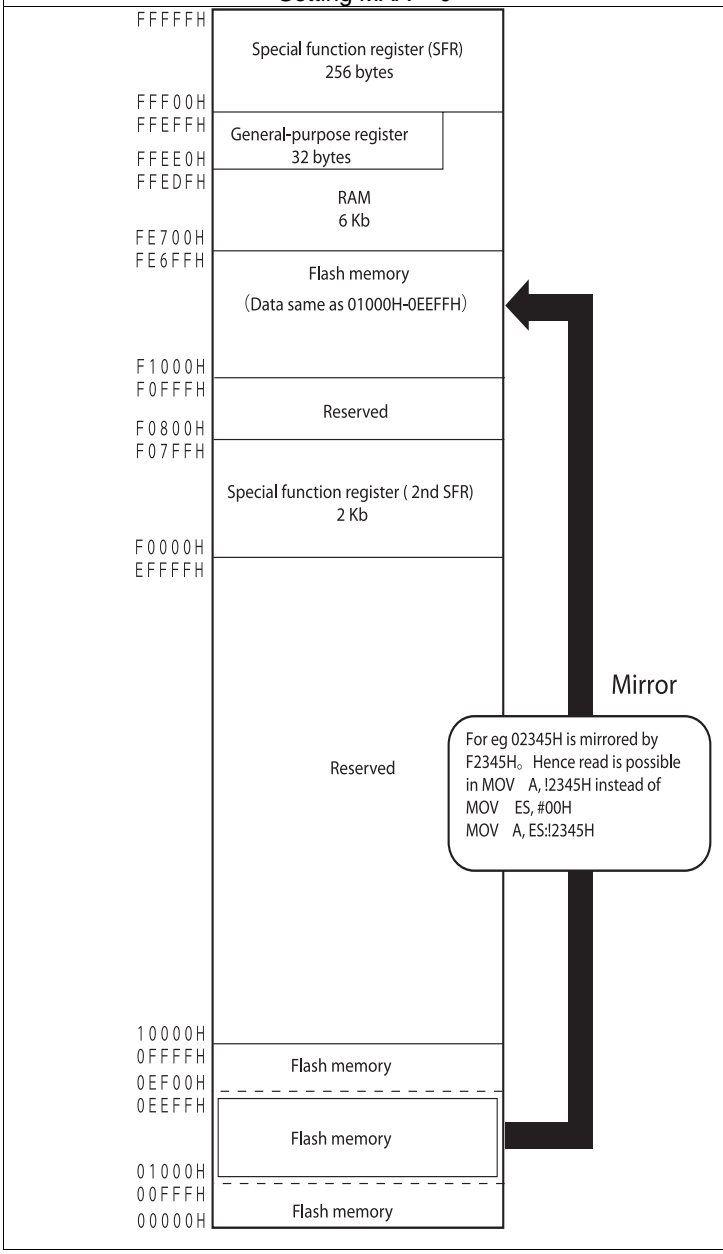
By reading data from F0000H - FFFFFH, an instruction that does not have the ES registers as an operand can be used, and thus the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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

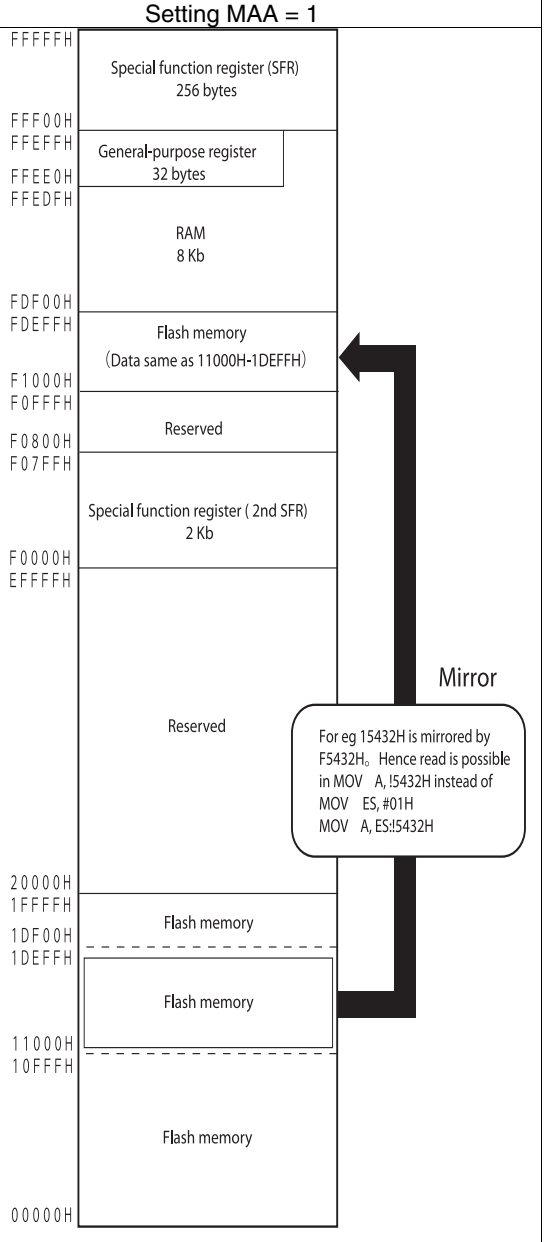
The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Example 1 μPD78F1022
(Flash memory: 64 KB, RAM: 6 KB)
Setting MAA = 0



Example 2 μPD78F1024, 78F1026
(Flash memory: 128 KB, RAM: 8 KB)
Setting MAA = 1



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

PMC register is described below.

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

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

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

Reset signal generation sets this register to 00H.

Figure 3-4. Format of **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 – FFFFFH
0	00000H - 0FFFFH is mirrored to F0000H – FFFFFH
1	10000H - 1FFFFH is mirrored to F0000H – FFFFFH

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

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

3. When the μPD78F1022 is used, be sure to set bit 0 (MAA) to 0 (default value).

3.1.3 Internal data memory space

78K0R/KC3-L, 78K0R/KE3-L products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
• PD78F1022	6144×8 Bit(FE700H-FFEFFH)
• PD78F1023, 78F1024, 78F1025, 78F1026	8192×8 Bit(FDF00H-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 - 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 - FFEFFH) space for fetching instructions or as a stack area.

2. While using the self-programming function, the area of FFE20H - FFEFFH cannot be used as a stack memory. Furthermore, the areas of FDF00H - FE2FFH also cannot be used as stack memories with the μPD78F1023, 78F1024, 78F1025, 78F1026 respectively.

3.1.4 Special function register (SFR) area

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

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

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

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

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

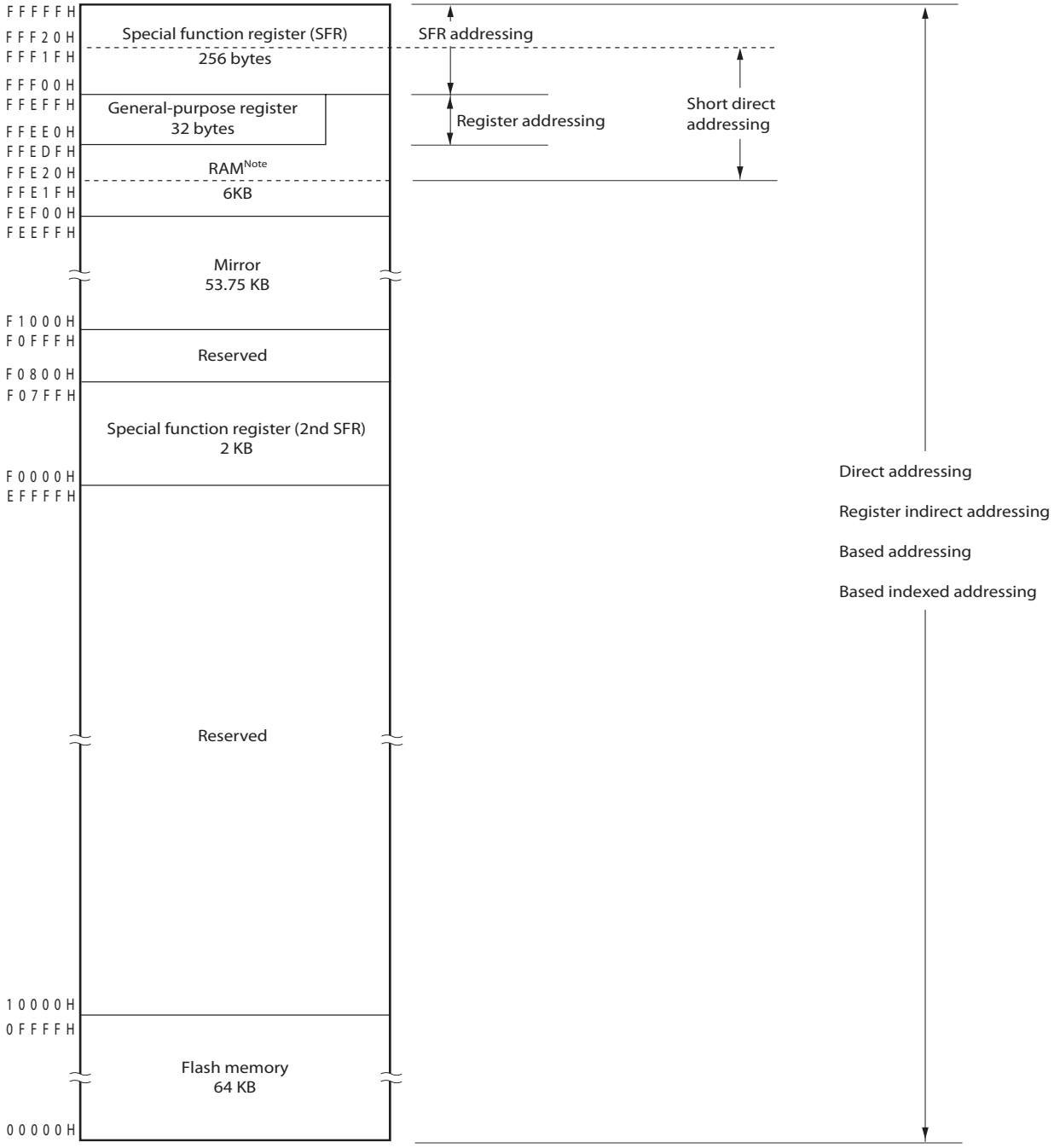
Caution Do not access addresses to which the 2nd SFR is not assigned.

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.

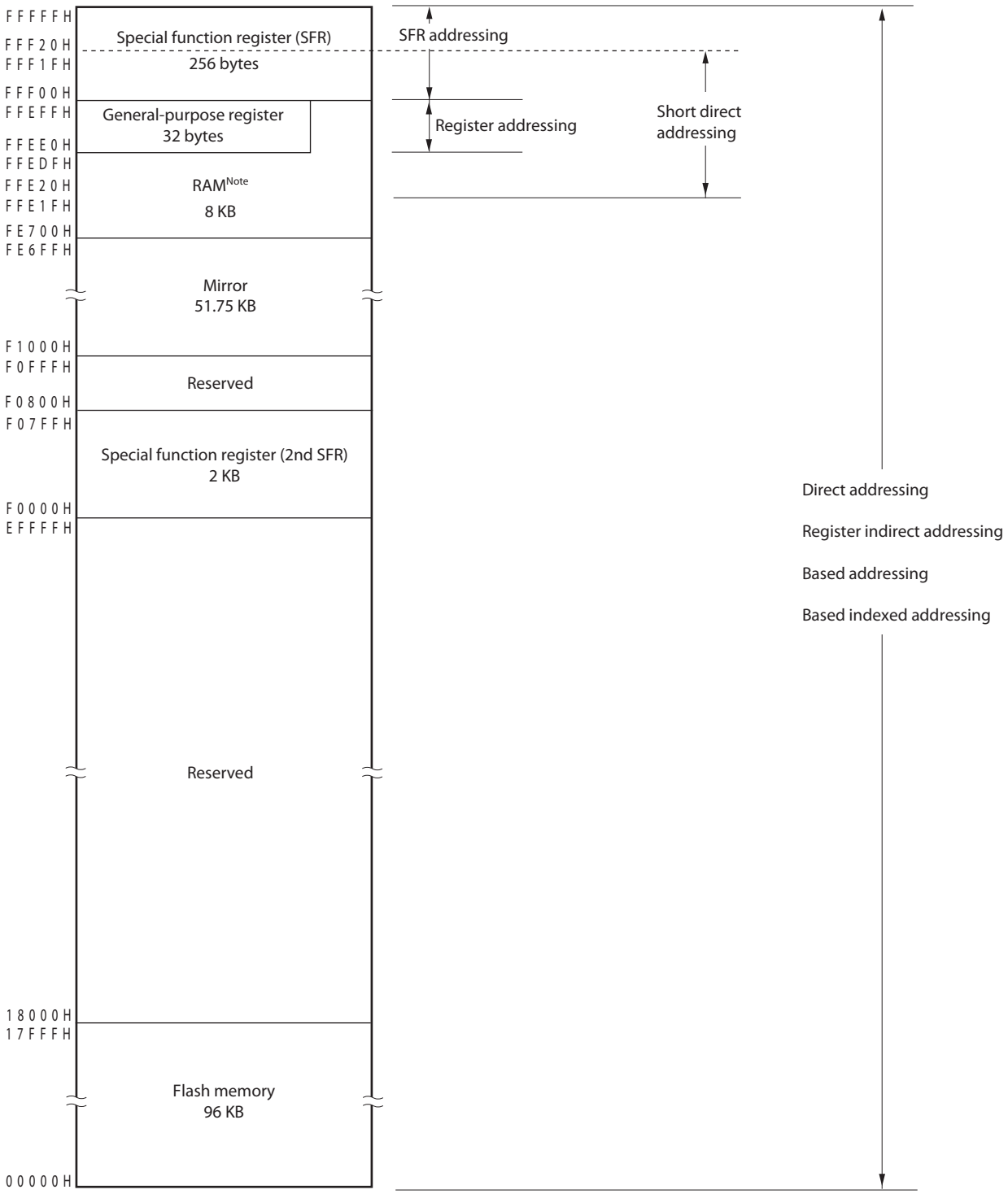
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the **78K0R/KC3-L, KE3-L**, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function 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 (μPD78F1022)



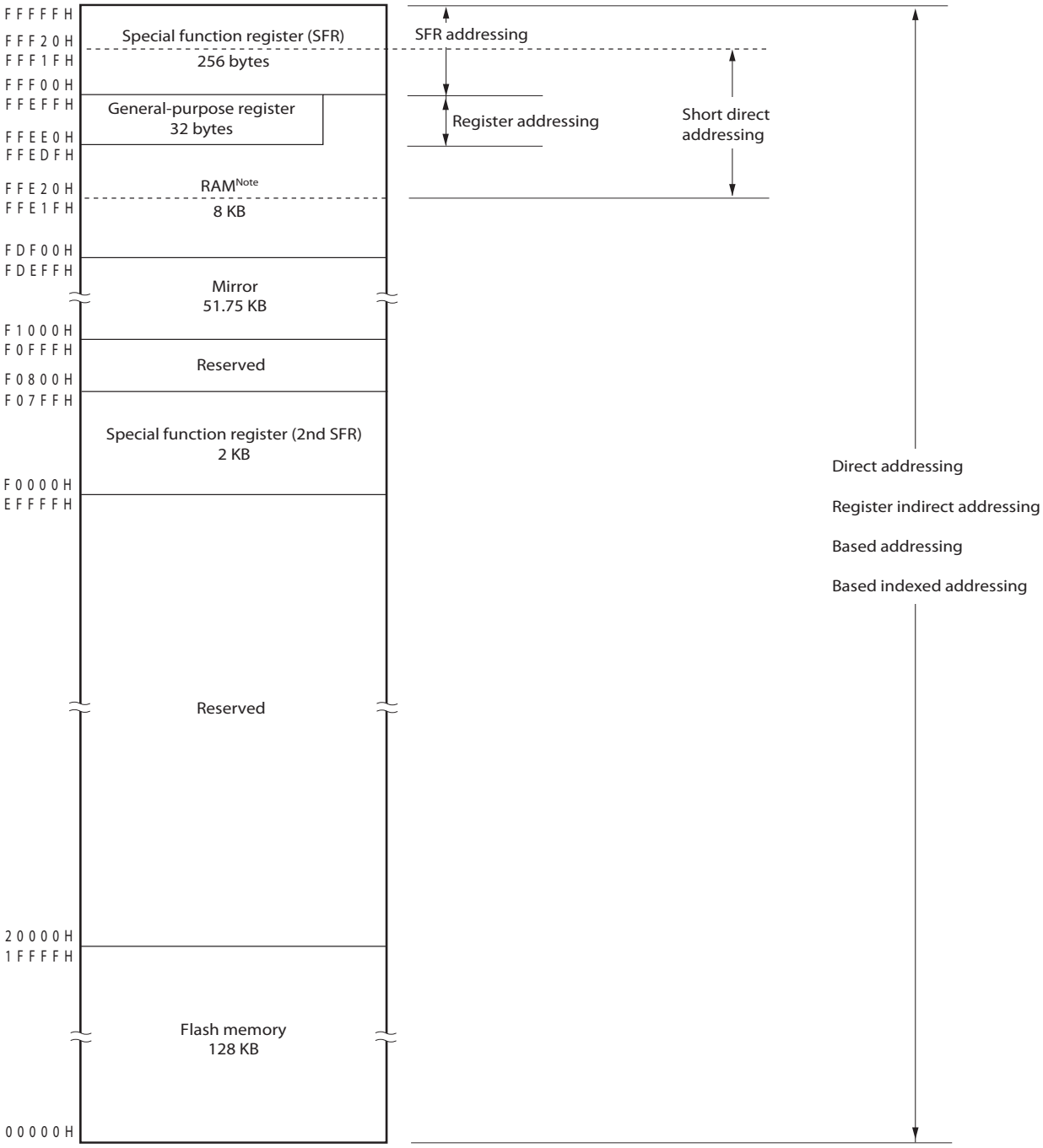
Note While using the self-programming function, the area FFE20H to FFEDFH cannot be used as stack memory.

Figure 3-6. Correspondence Between Data Memory and Addressing (μPD78F1023, 78F1025)



Note While using the self-programming function, the area FFE20H to FFEDFH and FDF00H to FE2FFH cannot be used as stack memory.

Figure 3-7 Correspondence Between Data Memory and Addressing (μPD78F1024, 78F1026)



Note While using the self-programming function, the area FFE20H to FFEDFH and FDF00H to FE2FFH cannot be used as stack memory.

3.2 Processor Registers

The **78K0R/KC3-L, KE3-L** products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

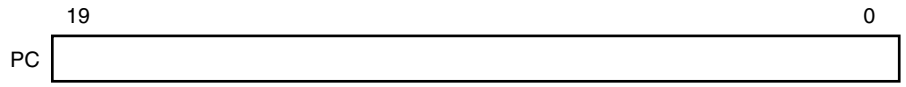
The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.

When a branch instruction is executed, immediate data and register contents are set.

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

Figure 3-8. 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 vector interrupt request acknowledgment 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 **15.3 (3)**) cannot 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.

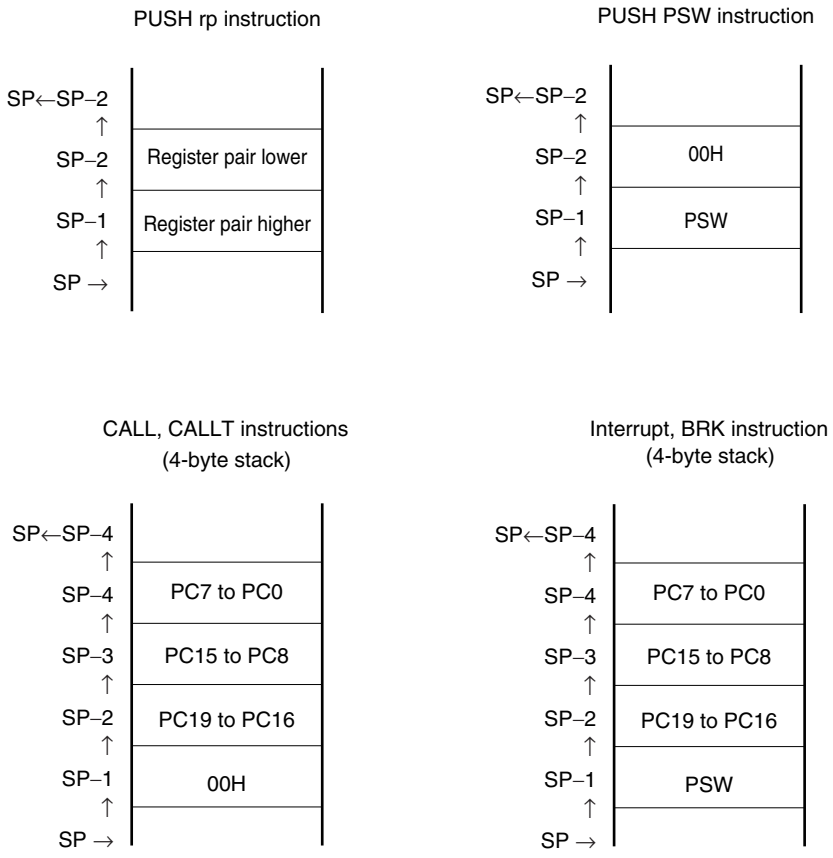
Each stack operation saves data as shown in Figure 3-11.

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 - FFEFFH**) space as a stack area.

3. While using the self-programming function, the area of **FFE20H - FFEFFH** cannot be used as a stack memory. Furthermore, the areas of **FDF00H - FE2FFH** also cannot be used as stack memories with the μPD78F1023, 78F1024, 78F1025, 78F1026 respectively.

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 - 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).

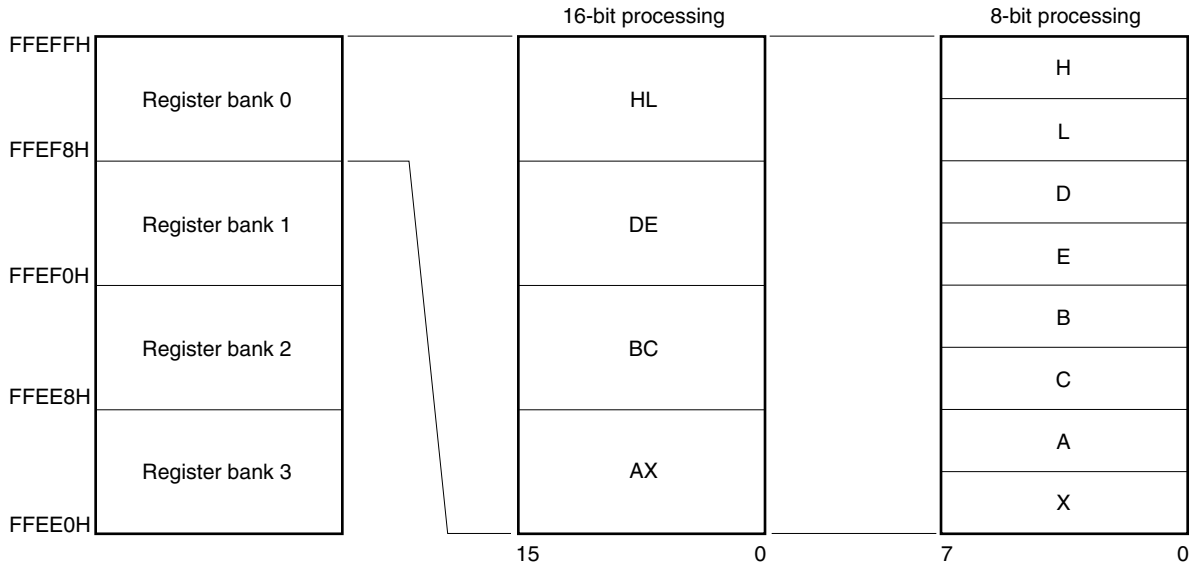
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

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

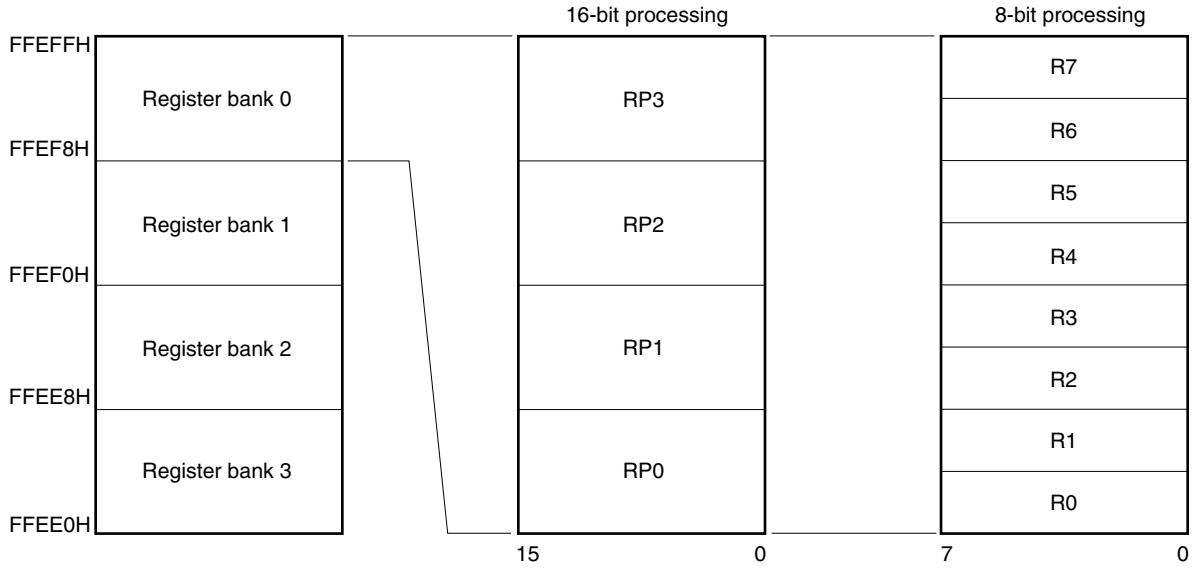
<R> **2. While using the self-programming function, the area FFE20H to FFEFFH cannot be used as stack memory. Furthermore, the area of FDF00H to FE2FFH also cannot be used with the μPD78F1023, 78F1024, 78F1025, and 78F1026, respectively.**

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 - 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	78K0R/KC3-L	78K0R/KE3-L
					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	√	√
FFF07H	Port register 7	P7		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	√	√
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	10 bit A/D Conversion Result Register	ADCR		R	–	–	√	0000H	√	√
FFF1FH	8-bit A/D conversion result register	ADCRH		R	–	√	–	00H	√	√
FFF20H	Port mode register 0	PM0		R/W	√	√	–	FFH	√	√
FFF21H	Port mode register 1	PM1		R/W	√	√	–	FFH	√	√
FFF22H	Port mode register 2	PM2		R/W	√	√	–	FFH	√	√
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	√	√
FFF27H	Port mode register 7	PM7		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	√	√

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L, 78K0R/KF3-L
					1 Bit	8 Bit	16 Bit			
FFF30H	A/D Conversion Mode Register	ADM		R/W	√	√	–	00H	√	√
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H	√	√
FFF37H	Key return mode register	KRM		R/W	√	√	–	00H	√	√
FFF38H	External interrupt rising edge enable register0	EGP0		R/W	√	√	–	00H	√	√
FFF39H	External interrupt falling edge enable register0	EGN0		R/W	√	√	–	00H	√	√
FFF3AH	External interrupt rising edge enable register1	EGP1		R/W	√	√	–	00H	–	√
FFF3BH	External interrupt falling edge enable register1	EGN1		R/W	√	√	–	00H	–	√
FFF3CH	Input switch control register	ISC		R/W	√	√	–	00H	√	√
FFF3EH	Timer input select register 0	TIS0		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	√	√
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H	√	√
FFF65H										
FFF66H	Timer data register 03	TDR03		R/W	–	–	√	0000H	√	√
FFF67H										
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										

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
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	√	√
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 stable time counter function register	OSTC	R	√	√	–	00H	√	√
FFFA3H	Oscillation stable time selection register	OSTS	R/W	–	√	–	07H	√	√
FFFA4H	Clock control register	CKC	R/W	√	√	–	09H	√	√
FFFA5H	Clock output selection register	CKS0	R/W	√	√	–	00H	√	√
FFFA8H	Reset control flag register	RESF	R	–	√	–	00H note1	√	√
FFFA9H	Low-voltage detection register	LVIM	R/W	√	√	–	00H note2	√	√
FFFAAH	Low-voltage detection level select register	LVIS	R/W	√	√	–	0EH note3	√	√
FFFABH	Watch dog timer enable register	WDTE	R/W	–	√	–	1A/9A note4	√	√

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.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
					1 Bit	8 Bit	16 Bit			
FFFB0H	DMA SFR Address register0	DSA0		R/W	–	√	–	00H	√	√
FFFB1H	DMA SFR Address register1	DSA1		R/W	–	√	–	00H	√	√
FFFB2H	DMA RAM Address register0L	DRA0L	DRA0	R/W	–	√	√	√	√	√
FFFB3H	DMA RAM Address register0H	DRA0H		R/W	–	√		√	√	√
FFFB4H	DMA RAM Address register1L	DRA1L	DRA1	R/W	–	√	√	00H	√	√
FFFB5H	DMA RAM Address register1H	DRA1H		R/W	–	√		00H	√	√
FFFB6H	DMA Byte count register0L	DBC0L	DBC0	R/W	–	√	√	00H	√	√
FFFB7H	DMA Byte count register0H	DBC0H		R/W	–	√		00H	√	√
FFFB8H	DMA Byte count register1L	DBC1L	DBC1	R/W	–	√	√	00H	√	√
FFFB9H	DMA Byte count register1H	DBC1H		R/W	–	√		00H	√	√
FFFBAH	DMA Mode control register0	DMC0		R/W	√	√	–	00H	√	√
FFFBBH	DMA Mode control register1	DMC1		R/W	√	√	–	00H	√	√
FFFBCH	DMA Operation control register0	DRC0		R/W	√	√	–	00H	√	√
FFFBDH	DMA Operation control register1	DRC1		R/W	√	√	–	00H	√	√
FFFBEH	Background event control register	BECTL		R/W	√	√	–	00H	√	√
FFFC0H	–	PFCMD[note]		–	–	–	–	Undefined	√	√
FFFC2H	–	PFS[note]		–	–	–	–	00H	√	√
FFFC4H	–	FLPMC[note]		–	–	–	–	08H	√	√
FFFD0H	Interrupt request flag register2L	IF2L	IF2	R/W	√	√	√	00H	√	√
FFFD1H	Interrupt request flag register2H	IF2H		R/W	√	√		00H	√	√
FFFD4H	Interrupt mask flag register2L	MK2L	MK2	R/W	√	√	√	FFH	√	√
FFFD5H	Interrupt mask flag register2H	MK2H		R/W	√	√		FFH	√	√
FFFD8H	Priority specification flag register02L	PR02L	PR02	R/W	√	√	√	FFH	√	√
FFFD9H	Priority specification flag register02H	PR02H		R/W	√	√		FFH	√	√
FFFDCH	Priority specification flag register12L	PR12L	PR12	R/W	√	√	√	FFH	√	√
FFFDDH	Priority specification flag register12H	PR12H		R/W	√	√		FFH	√	√
FFFE0H	Interrupt request flag register0L	IF0L	IF0	R/W	√	√	√	00H	√	√
FFFE1H	Interrupt request flag register0H	IF0H		R/W	√	√		00H	√	√
FFFE2H	Interrupt request flag register1L	IF1L	IF1	R/W	√	√	√	00H	√	√
FFFE3H	Interrupt request flag register1H	IF1H		R/W	√	√		00H	√	√
FFFE4H	Interrupt mask flag register0L	MK0L	MK0	R/W	√	√	√	FFH	√	√
FFFE5H	Interrupt mask flag register0H	MK0H		R/W	√	√		FFH	√	√
FFFE6H	Interrupt mask flag register1L	MK1L	MK1	R/W	√	√	√	FFH	√	√
FFFE7H	Interrupt mask flag register1H	MK1H		R/W	√	√		FFH	√	√
FFFE8H	Priority specification flag register00L	PR00L	PR00	R/W	√	√	√	FFH	√	√
FFFE9H	Priority specification flag register00H	PR00H		R/W	√	√		FFH	√	√

Note 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	78K0R/KC3-L	78K0R/KE3-L
					1 Bit	8 Bit	16 Bit			
FFFEAH	Priority specification flag register01L	PR01L	PR01	R/W	√	√	√	FFH	√	√
FFFEBH	Priority specification flag register01H	PR01H		R/W	√	√		FFH	√	√
FFFECH	Priority specification flag register10L	PR10L	PR10	R/W	√	√	√	FFH	√	√
FFFEDH	Priority specification flag register10H	PR10H		R/W	√	√		FFH	√	√
FFFEEH	Priority specification flag register11L	PR11L	PR11	R/W	√	√	√	FFH	√	√
FFFEFH	Priority specification flag register11H	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										
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H	√	√

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 - F07FFH area. SFRs other than those in the SFR area (FFF00H - 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 a 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 the 2nd SFR is 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 (1/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
					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	√	√
F0037H	Pull-up resistor option register 7	PU7		R/W	√	√	–	00H	√	√
F003BH	Pull-up resistor option register 11	PU11		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	√	√
F0040H	Port input mode register 0	PIM0		R/W	√	√	–	00H	√	√
F0041H	Port input mode register 1	PIM1		R/W	√	√	–	00H	√	√
F004EH	Port input mode register 14	PIM14		R/W	√	√	–	00H	–	√
F0050H	Port output mode register 0	POM0		R/W	√	√	–	00H	√	√
F0051H	Port output mode register 1	POM1		R/W	√	√	–	00H	√	√
F005EH	Port output mode register 14	POM14		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	√	√
F00E2H	Multiplication/division data register C (H)	MDCH		R	–	–	√	0000H	√	√
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	–	√	–	00H	√	√
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H	√	√
F0101H		–			–	–			√	√
<R> F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H	√	√
<R> 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		–			–	–			√	√
<R> F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H	√	√
<R> F010BH		–			–	–			√	√
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H	√	√
F010DH		–			–	–			√	√

Table 3-6. Extended SFR (2nd SFR) List (2/21)

	Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
						1 Bit	8 Bit	16 Bit			
	F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H	√	√
	F010FH		–			–	–			√	√
	F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H	√	√
	F0111H										
<R>	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										
<R>	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		–			–	–			√	√
<R>	F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H	–	√
<R>	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 (3/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
					1 Bit	8 Bit	16 Bit			
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	–	√	√	0000H	√	√
F014DH		–			–	–			√	√
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	SCR10		R/W	–	–	√	0087H	–	√
F0159H	setting register 10				–					
F015AH	Serial communication operation	SCR11		R/W	–	–	√	0087H	–	√
F015BH	setting register 11									
F015CH	Serial communication operation	SCR12		R/W	–	–	√	0087H	√	√
F015DH	setting register 12				–					
F015EH	Serial communication operation	SCR13		R/W	–	–	√	0087H	√	√
F015FH	setting register 13									
F0160H	Serial channel enable status	SE1L	SE1	R	√	√	√	0000H	√	√
F0161H	register 1	–			–	–				
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										

Table 3-6. Extended SFR (2nd SFR) List (4/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
					1 Bit	8 Bit	16 Bit			
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH	√	√
F0187H										
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH	√	√
F0189H										
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH	√	√
F018BH										
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH	√	√
F018DH										
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH	√	√
F018FH										
F0190H	Timer mode register 00	TMR00		R/W	–	–	√	0000H	√	√
F0191H										
F0192H	Timer mode register 01	TMR01		R/W	–	–	√	0000H	√	√
F0193H										
F0194H	Timer mode register 02	TMR02		R/W	–	–	√	0000H	√	√
F0195H										
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H	√	√
F0197H										
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H	√	√
F0199H										
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H	√	√
F019BH										
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H	√	√
F019DH										
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H	√	√
F019FH										
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H	√	√
F01A1H		–			–	–				
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H	√	√
F01A3H		–			–	–				
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H	√	√
F01A5H		–			–	–				
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H	√	√
F01A7H		–			–	–				
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H	√	√
F01A9H		–			–	–				
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H	√	√
F01ABH		–			–	–				
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H	√	√
F01ADH		–			–	–				

Table 3-6. Extended SFR (2nd SFR) List (5/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
					1-bit	8-bit	16-bit			
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H	√	√
F01AFH		–			–	–			√	√
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H	√	√
F01B1H		–			–	–			√	√
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H	√	√
F01B3H		–			–	–			√	√
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H	√	√
F01B5H		–			–	–			√	√
F01B6H	Timer clock select register 0	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		–			–	–			√	√
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	√	√

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

Table 3-6 Extended SFR (2nd SFR) List (6/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F0540H	UF0 EP0NAK Register	UF0E0N	R/W	–	√	–	00H	√	√
F0541H	UF0 EP0NAKALL Register	UF0E0NA	R/W	–	√	–	00H	√	√
F0542H	UF0 EPNAK Register	UF0EN	R/W	–	√	–	00H	√	√
F0543H	UF0 EPNAK Mask Register	UF0ENM	R/W	–	√	–	00H	√	√
F0544H	UF0 SNDSIE Register	UF0SDS	R/W	–	√	–	00H	√	√
F0545H	UF0 CLR Request Register	UF0CLR	R	–	√	–	00H	√	√
F0546H	UF0 SET Request Register	UF0SET	R	–	√	–	00H	√	√
F0547H	UF0 EP Status 0 Register	UF0EPS0	R	–	√	–	00H	√	√
F0548H	UF0 EP Status 1 Register	UF0EPS1	R	–	√	–	00H	√	√
F0549H	UF0 EP Status 2 Register	UF0EPS2	R	–	√	–	00H	√	√
F0550H	UF0 INT Status 0 Register	UF0IS0	R	–	√	–	00H	√	√
F0551H	UF0 INT Status 1 Register	UF0IS1	R	–	√	–	00H	√	√
F0552H	UF0 INT Status 2 Register	UF0IS2	R	–	√	–	00H	√	√
F0553H	UF0 INT Status 3 Register	UF0IS3	R	–	√	–	00H	√	√
F0554H	UF0 INT Status 4 Register	UF0IS4	R	–	√	–	00H	√	√
F0557H	UF0 INT Mask 0 Register	UF0IM0	R/W	–	√	–	00H	√	√
F0558H	UF0 INT Mask 1 Register	UF0IM1	R/W	–	√	–	00H	√	√
F0559H	UF0 INT Mask 2 Register	UF0IM2	R/W	–	√	–	00H	√	√
F055AH	UF0 INT Mask 3 Register	UF0IM3	R/W	–	√	–	00H	√	√
F055BH	UF0 INT Mask 4 Register	UF0IM4	R/W	–	√	–	00H	√	√
F055EH	UF0 INT Clear 0 Register	UF0IC0	W	–	√	–	FFH	√	√
F055FH	UF0 INT Clear 1 Register	UF0IC1	W	–	√	–	FFH	√	√
F0560H	UF0 INT Clear 2 Register	UF0IC2	W	–	√	–	FFH	√	√
F0561H	UF0 INT Clear 3 Register	UF0IC3	W	–	√	–	FFH	√	√
F0562H	UF0 INT Clear 4 Register	UF0IC4	W	–	√	–	FFH	√	√
F0570H	UF0 FIFO Clear 0 Register	UF0FIC0	W	–	√	–	00H	√	√
F0571H	UF0 FIFO Clear 1 Register	UF0FIC1	W	–	√	–	00H	√	√
F0575H	UF0 Data End Register	UF0DEND	R/W	–	√	–	00H	√	√
F0577H	UF0 GPR Register	UF0GPR	R/W	–	√	–	00H	√	√
F057AH	UF0 Mode Control Register	UF0MODC	R/W	–	√	–	00H	√	√
F057CH	UF0 Mode Status Register	UF0MODS	R	–	√	–	00H	√	√
F0580H	UF0 EP0 Lead Register	UF0E0R	R	–	√	–	Undefined	√	√
F0581H	UF0 EP0 Length Register	UF0E0L	R	–	√	–	00H	√	√
F0582H	UF0 EP0 Set up Register	UF0E0ST	R	–	√	–	00H	√	√
F0583H	UF0 EP0 Light Register	UF0E0W	W	–	√	–	00H	√	√
F0584H	UF0 Bulk Out 1 Register	UF0BO1	R	–	√	–	Undefined	√	√
F0585H	UF0 Bulk Out 1 Length Register	UF0BO1L	R	–	√	–	00H	√	√
F0586H	UF0 Bulk Out 2 Register	UF0BO2	R	–	√	–	Undefined	√	√
F0587H	UF0 Bulk Out 2 Length Register	UF0BO2L	R	–	√	–	00H	√	√
F0588H	UF0 Bulk In 1 Register	UF0BI1	W	–	√	–	00H	√	√

Table 3-6 Extended SFR (2nd SFR) List (8/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F0589H	UF0 Bulk In 2 Register	UF0BI2	W	–	√	–	00H	√	√
F058AH	UF0 Interrupt 1 Register	UF0INT1	W	–	√	–	00H	√	√
F058BH	UF0 Interrupt 2 Register	UF0INT2	W	–	√	–	00H	√	√
F059CH	PLLControl Register	PLLC	R/W	√	√	–	01H	√	√
F059DH	USB Clock Control Register	UCKC	R/W	√	√	–	00H	√	√
F059EH	Buffer Control Register	UF0BC	R/W	–	√	–	00H	√	√
F05A2H	Device Status Register L	UF0DSTL	R/W	–	√	–	00H	√	√
F05A6H	EP0 Status Register L	UF0E0SL	R/W	–	√	–	00H	√	√
F05A8H	EP1 Status Register L	UF0E1SL	R/W	–	√	–	00H	√	√
F05AAH	EP2 Status Register L	UF0E2SL	R/W	–	√	–	00H	√	√
F05ACH	EP3 Status Register L	UF0E3SL	R/W	–	√	–	00H	√	√
F05AEH	EP4 Status Register L	UF0E4SL	R/W	–	√	–	00H	√	√
F05B4H	EP7 Status Register L	UF0E7SL	R/W	–	√	–	00H	√	√
F05B6H	EP8 Status Register L	UF0E8SL	R/W	–	√	–	00H	√	√
F05C0H	Address Register	UF0ADRS	R	–	√	–	00H	√	√
F05C1H	Configuration Register	UF0CNF	R	–	√	–	00H	√	√
F05C2H	Interface 0 Register	UF0IF0	R	–	√	–	00H	√	√
F05C3H	Interface 1 Register	UF0IF1	R	–	√	–	00H	√	√
F05C4H	Interface 2 Register	UF0IF2	R	–	√	–	00H	√	√
F05C5H	Interface 3 Register	UF0IF3	R	–	√	–	00H	√	√
F05C6H	Interface 4 Register	UF0IF4	R	–	√	–	00H	√	√
F05D0H	Descriptor Length Register	UF0DSCL	R/W	–	√	–	00H	√	√
F05D1H	Device Descriptor Register 0	UF0DD0	R/W	–	√	–	Undefined	√	√
F05D2H	Device Descriptor Register 1	UF0DD1	R/W	–	√	–	Undefined	√	√
F05D3H	Device Descriptor Register 2	UF0DD2	R/W	–	√	–	Undefined	√	√
F05D4H	Device Descriptor Register 3	UF0DD3	R/W	–	√	–	Undefined	√	√
F05D5H	Device Descriptor Register 4	UF0DD4	R/W	–	√	–	Undefined	√	√
F05D6H	Device Descriptor Register 5	UF0DD5	R/W	–	√	–	Undefined	√	√
F05D7H	Device Descriptor Register 6	UF0DD6	R/W	–	√	–	Undefined	√	√
F05D8H	Device Descriptor Register 7	UF0DD7	R/W	–	√	–	Undefined	√	√
F05D9H	Device Descriptor Register 8	UF0DD8	R/W	–	√	–	Undefined	√	√
F05DAH	Device Descriptor Register 9	UF0DD9	R/W	–	√	–	Undefined	√	√
F05DBH	Device Descriptor Register 10	UF0DD10	R/W	–	√	–	Undefined	√	√
F05DCH	Device Descriptor Register 11	UF0DD11	R/W	–	√	–	Undefined	√	√
F05DDH	Device Descriptor Register 12	UF0DD12	R/W	–	√	–	Undefined	√	√
F05DEH	Device Descriptor Register 13	UF0DD13	R/W	–	√	–	Undefined	√	√
F05DFH	Device Descriptor Register 14	UF0DD14	R/W	–	√	–	Undefined	√	√
F05E0H	Device Descriptor Register 15	UF0DD15	R/W	–	√	–	Undefined	√	√
F05E1H	Device Descriptor Register 16	UF0DD16	R/W	–	√	–	Undefined	√	√
F05E2H	Device Descriptor Register 17	UF0DD17	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (8/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F05E3H	Configuration/ Interface / End Point Descriptor Register 0	UF0CIE0	R/W	–	√	–	Undefined	√	√
F05E4H	Configuration/ Interface / End Point Descriptor Register 1	UF0CIE1	R/W	–	√	–	Undefined	√	√
F05E5H	Configuration/ Interface / End Point Descriptor Register 2	UF0CIE2	R/W	–	√	–	Undefined	√	√
F05E6H	Configuration/ Interface / End Point Descriptor Register 3	UF0CIE3	R/W	–	√	–	Undefined	√	√
F05E7H	Configuration/ Interface / End Point Descriptor Register 4	UF0CIE4	R/W	–	√	–	Undefined	√	√
F05E8H	Configuration/ Interface / End Point Descriptor Register 5	UF0CIE5	R/W	–	√	–	Undefined	√	√
F05E9H	Configuration/ Interface / End Point Descriptor Register 6	UF0CIE6	R/W	–	√	–	Undefined	√	√
F05EAH	Configuration/ Interface / End Point Descriptor Register 7	UF0CIE7	R/W	–	√	–	Undefined	√	√
F05EBH	Configuration/ Interface / End Point Descriptor Register 8	UF0CIE8	R/W	–	√	–	Undefined	√	√
F05ECH	Configuration/ Interface / End Point Descriptor Register 9	UF0CIE9	R/W	–	√	–	Undefined	√	√
F05EDH	Configuration/ Interface / End Point Descriptor Register 10	UF0CIE10	R/W	–	√	–	Undefined	√	√
F05EEH	Configuration/ Interface / End Point Descriptor Register 11	UF0CIE11	R/W	–	√	–	Undefined	√	√
F05EFH	Configuration/ Interface / End Point Descriptor Register 12	UF0CIE12	R/W	–	√	–	Undefined	√	√
F05F0H	Configuration/ Interface / End Point Descriptor Register 13	UF0CIE13	R/W	–	√	–	Undefined	√	√
F05F1H	Configuration/ Interface / End Point Descriptor Register 14	UF0CIE14	R/W	–	√	–	Undefined	√	√
F05F2H	Configuration/ Interface / End Point Descriptor Register 15	UF0CIE15	R/W	–	√	–	Undefined	√	√
F05F3H	Configuration/ Interface / End Point Descriptor Register 16	UF0CIE16	R/W	–	√	–	Undefined	√	√
F05F4H	Configuration/ Interface / End Point Descriptor Register 17	UF0CIE17	R/W	–	√	–	Undefined	√	√
F05F5H	Configuration/ Interface / End Point Descriptor Register 18	UF0CIE18	R/W	–	√	–	Undefined	√	√
F05F6H	Configuration/ Interface / End Point Descriptor Register 19	UF0CIE19	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (9/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F05F7H	Configuration/ Interface / End Point Descriptor Register 20	UF0CIE20	R/W	–	√	–	Undefined	√	√
F05F8H	Configuration/ Interface / End Point Descriptor Register 21	UF0CIE21	R/W	–	√	–	Undefined	√	√
F05F9H	Configuration/ Interface / End Point Descriptor Register 22	UF0CIE22	R/W	–	√	–	Undefined	√	√
F05FAH	Configuration/ Interface / End Point Descriptor Register 23	UF0CIE23	R/W	–	√	–	Undefined	√	√
F05FBH	Configuration/ Interface / End Point Descriptor Register 24	UF0CIE24	R/W	–	√	–	Undefined	√	√
F05FCH	Configuration/ Interface / End Point Descriptor Register 25	UF0CIE25	R/W	–	√	–	Undefined	√	√
F05FDH	Configuration/ Interface / End Point Descriptor Register 26	UF0CIE26	R/W	–	√	–	Undefined	√	√
F05FEH	Configuration/ Interface / End Point Descriptor Register 27	UF0CIE27	R/W	–	√	–	Undefined	√	√
F05FFH	Configuration/ Interface / End Point Descriptor Register 28	UF0CIE28	R/W	–	√	–	Undefined	√	√
F0600H	Configuration/ Interface / End Point Descriptor Register 29	UF0CIE29	R/W	–	√	–	Undefined	√	√
F0601H	Configuration/ Interface / End Point Descriptor Register 30	UF0CIE30	R/W	–	√	–	Undefined	√	√
F0602H	Configuration/ Interface / End Point Descriptor Register 31	UF0CIE31	R/W	–	√	–	Undefined	√	√
F0603H	Configuration/ Interface / End Point Descriptor Register 32	UF0CIE32	R/W	–	√	–	Undefined	√	√
F0604H	Configuration/ Interface / End Point Descriptor Register 33	UF0CIE33	R/W	–	√	–	Undefined	√	√
F0605H	Configuration/ Interface / End Point Descriptor Register 34	UF0CIE34	R/W	–	√	–	Undefined	√	√
F0606H	Configuration/ Interface / End Point Descriptor Register 35	UF0CIE35	R/W	–	√	–	Undefined	√	√
F0607H	Configuration/ Interface / End Point Descriptor Register 36	UF0CIE36	R/W	–	√	–	Undefined	√	√
F0608H	Configuration/ Interface / End Point Descriptor Register 37	UF0CIE37	R/W	–	√	–	Undefined	√	√
F0609H	Configuration/ Interface / End Point Descriptor Register 38	UF0CIE38	R/W	–	√	–	Undefined	√	√
F060AH	Configuration/ Interface / End Point Descriptor Register 39	UF0CIE39	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (10/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F060BH	Configuration/ Interface / End Point Descriptor Register 40	UF0CIE40	R/W	–	√	–	Undefined	√	√
F060CH	Configuration/ Interface / End Point Descriptor Register 41	UF0CIE41	R/W	–	√	–	Undefined	√	√
F060DH	Configuration/ Interface / End Point Descriptor Register 42	UF0CIE42	R/W	–	√	–	Undefined	√	√
F060EH	Configuration/ Interface / End Point Descriptor Register 43	UF0CIE43	R/W	–	√	–	Undefined	√	√
F060FH	Configuration/ Interface / End Point Descriptor Register 44	UF0CIE44	R/W	–	√	–	Undefined	√	√
F0610H	Configuration/ Interface / End Point Descriptor Register 45	UF0CIE45	R/W	–	√	–	Undefined	√	√
F0611H	Configuration/ Interface / End Point Descriptor Register 46	UF0CIE46	R/W	–	√	–	Undefined	√	√
F0612H	Configuration/ Interface / End Point Descriptor Register 47	UF0CIE47	R/W	–	√	–	Undefined	√	√
F0613H	Configuration/ Interface / End Point Descriptor Register 48	UF0CIE48	R/W	–	√	–	Undefined	√	√
F0614H	Configuration/ Interface / End Point Descriptor Register 49	UF0CIE49	R/W	–	√	–	Undefined	√	√
F0615H	Configuration/ Interface / End Point Descriptor Register 50	UF0CIE50	R/W	–	√	–	Undefined	√	√
F0616H	Configuration/ Interface / End Point Descriptor Register 51	UF0CIE51	R/W	–	√	–	Undefined	√	√
F0617H	Configuration/ Interface / End Point Descriptor Register 52	UF0CIE52	R/W	–	√	–	Undefined	√	√
F0618H	Configuration/ Interface / End Point Descriptor Register 53	UF0CIE53	R/W	–	√	–	Undefined	√	√
F0619H	Configuration/ Interface / End Point Descriptor Register 54	UF0CIE54	R/W	–	√	–	Undefined	√	√
F061AH	Configuration/ Interface / End Point Descriptor Register 55	UF0CIE55	R/W	–	√	–	Undefined	√	√
F061BH	Configuration/ Interface / End Point Descriptor Register 56	UF0CIE56	R/W	–	√	–	Undefined	√	√
F061CH	Configuration/ Interface / End Point Descriptor Register 57	UF0CIE57	R/W	–	√	–	Undefined	√	√
F061DH	Configuration/ Interface / End Point Descriptor Register 58	UF0CIE58	R/W	–	√	–	Undefined	√	√
F061EH	Configuration/ Interface / End Point Descriptor Register 59	UF0CIE59	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (11/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F061FH	Configuration/ Interface / End Point Descriptor Register 60	UF0CIE60	R/W	–	√	–	Undefined	√	√
F0620H	Configuration/ Interface / End Point Descriptor Register 61	UF0CIE61	R/W	–	√	–	Undefined	√	√
F0621H	Configuration/ Interface / End Point Descriptor Register 62	UF0CIE62	R/W	–	√	–	Undefined	√	√
F0622H	Configuration/ Interface / End Point Descriptor Register 63	UF0CIE63	R/W	–	√	–	Undefined	√	√
F0623H	Configuration/ Interface / End Point Descriptor Register 64	UF0CIE64	R/W	–	√	–	Undefined	√	√
F0624H	Configuration/ Interface / End Point Descriptor Register 65	UF0CIE65	R/W	–	√	–	Undefined	√	√
F0625H	Configuration/ Interface / End Point Descriptor Register 66	UF0CIE66	R/W	–	√	–	Undefined	√	√
F0626H	Configuration/ Interface / End Point Descriptor Register 67	UF0CIE67	R/W	–	√	–	Undefined	√	√
F0627H	Configuration/ Interface / End Point Descriptor Register 68	UF0CIE68	R/W	–	√	–	Undefined	√	√
F0628H	Configuration/ Interface / End Point Descriptor Register 69	UF0CIE69	R/W	–	√	–	Undefined	√	√
F0629H	Configuration/ Interface / End Point Descriptor Register 70	UF0CIE70	R/W	–	√	–	Undefined	√	√
F062AH	Configuration/ Interface / End Point Descriptor Register 71	UF0CIE71	R/W	–	√	–	Undefined	√	√
F062BH	Configuration/ Interface / End Point Descriptor Register 72	UF0CIE72	R/W	–	√	–	Undefined	√	√
F062CH	Configuration/ Interface / End Point Descriptor Register 73	UF0CIE73	R/W	–	√	–	Undefined	√	√
F062DH	Configuration/ Interface / End Point Descriptor Register 74	UF0CIE74	R/W	–	√	–	Undefined	√	√
F062EH	Configuration/ Interface / End Point Descriptor Register 75	UF0CIE75	R/W	–	√	–	Undefined	√	√
F062FH	Configuration/ Interface / End Point Descriptor Register 76	UF0CIE76	R/W	–	√	–	Undefined	√	√
F0630H	Configuration/ Interface / End Point Descriptor Register 77	UF0CIE77	R/W	–	√	–	Undefined	√	√
F0631H	Configuration/ Interface / End Point Descriptor Register 78	UF0CIE78	R/W	–	√	–	Undefined	√	√
F0632H	Configuration/ Interface / End Point Descriptor Register 79	UF0CIE79	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (12/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F0633H	Configuration/ Interface / End Point Descriptor Register 80	UF0CIE80	R/W	–	√	–	Undefined	√	√
F0634H	Configuration/ Interface / End Point Descriptor Register 81	UF0CIE81	R/W	–	√	–	Undefined	√	√
F0635H	Configuration/ Interface / End Point Descriptor Register 82	UF0CIE82	R/W	–	√	–	Undefined	√	√
F0636H	Configuration/ Interface / End Point Descriptor Register 83	UF0CIE83	R/W	–	√	–	Undefined	√	√
F0637H	Configuration/ Interface / End Point Descriptor Register 84	UF0CIE84	R/W	–	√	–	Undefined	√	√
F0638H	Configuration/ Interface / End Point Descriptor Register 85	UF0CIE85	R/W	–	√	–	Undefined	√	√
F0639H	Configuration/ Interface / End Point Descriptor Register 86	UF0CIE86	R/W	–	√	–	Undefined	√	√
F063AH	Configuration/ Interface / End Point Descriptor Register 87	UF0CIE87	R/W	–	√	–	Undefined	√	√
F063BH	Configuration/ Interface / End Point Descriptor Register 88	UF0CIE88	R/W	–	√	–	Undefined	√	√
F063CH	Configuration/ Interface / End Point Descriptor Register 89	UF0CIE89	R/W	–	√	–	Undefined	√	√
F063DH	Configuration/ Interface / End Point Descriptor Register 90	UF0CIE90	R/W	–	√	–	Undefined	√	√
F063EH	Configuration/ Interface / End Point Descriptor Register 91	UF0CIE91	R/W	–	√	–	Undefined	√	√
F063FH	Configuration/ Interface / End Point Descriptor Register 92	UF0CIE92	R/W	–	√	–	Undefined	√	√
F0640H	Configuration/ Interface / End Point Descriptor Register 93	UF0CIE93	R/W	–	√	–	Undefined	√	√
F0641H	Configuration/ Interface / End Point Descriptor Register 94	UF0CIE94	R/W	–	√	–	Undefined	√	√
F0642H	Configuration/ Interface / End Point Descriptor Register 95	UF0CIE95	R/W	–	√	–	Undefined	√	√
F0643H	Configuration/ Interface / End Point Descriptor Register 96	UF0CIE96	R/W	–	√	–	Undefined	√	√
F0644H	Configuration/ Interface / End Point Descriptor Register 97	UF0CIE97	R/W	–	√	–	Undefined	√	√
F0645H	Configuration/ Interface / End Point Descriptor Register 98	UF0CIE98	R/W	–	√	–	Undefined	√	√
F0646H	Configuration/ Interface / End Point Descriptor Register 99	UF0CIE99	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (13/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F0647H	Configuration/ Interface / End Point Descriptor Register 100	UF0CIE100	R/W	–	√	–	Undefined	√	√
F0648H	Configuration/ Interface / End Point Descriptor Register 101	UF0CIE101	R/W	–	√	–	Undefined	√	√
F0649H	Configuration/ Interface / End Point Descriptor Register 102	UF0CIE102	R/W	–	√	–	Undefined	√	√
F064AH	Configuration/ Interface / End Point Descriptor Register 103	UF0CIE103	R/W	–	√	–	Undefined	√	√
F064BH	Configuration/ Interface / End Point Descriptor Register 104	UF0CIE104	R/W	–	√	–	Undefined	√	√
F064CH	Configuration/ Interface / End Point Descriptor Register 105	UF0CIE105	R/W	–	√	–	Undefined	√	√
F064DH	Configuration/ Interface / End Point Descriptor Register 106	UF0CIE106	R/W	–	√	–	Undefined	√	√
F064EH	Configuration/ Interface / End Point Descriptor Register 107	UF0CIE107	R/W	–	√	–	Undefined	√	√
F064FH	Configuration/ Interface / End Point Descriptor Register 108	UF0CIE108	R/W	–	√	–	Undefined	√	√
F0650H	Configuration/ Interface / End Point Descriptor Register 109	UF0CIE109	R/W	–	√	–	Undefined	√	√
F0651H	Configuration/ Interface / End Point Descriptor Register 110	UF0CIE110	R/W	–	√	–	Undefined	√	√
F0652H	Configuration/ Interface / End Point Descriptor Register 111	UF0CIE111	R/W	–	√	–	Undefined	√	√
F0653H	Configuration/ Interface / End Point Descriptor Register 112	UF0CIE112	R/W	–	√	–	Undefined	√	√
F0654H	Configuration/ Interface / End Point Descriptor Register 113	UF0CIE113	R/W	–	√	–	Undefined	√	√
F0655H	Configuration/ Interface / End Point Descriptor Register 114	UF0CIE114	R/W	–	√	–	Undefined	√	√
F0656H	Configuration/ Interface / End Point Descriptor Register 115	UF0CIE115	R/W	–	√	–	Undefined	√	√
F0657H	Configuration/ Interface / End Point Descriptor Register 116	UF0CIE116	R/W	–	√	–	Undefined	√	√
F0658H	Configuration/ Interface / End Point Descriptor Register 117	UF0CIE117	R/W	–	√	–	Undefined	√	√
F0659H	Configuration/ Interface / End Point Descriptor Register 118	UF0CIE118	R/W	–	√	–	Undefined	√	√
F065AH	Configuration/ Interface / End Point Descriptor Register 119	UF0CIE119	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (14/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F065BH	Configuration/ Interface / End Point Descriptor Register 120	UF0CIE120	R/W	–	√	–	Undefined	√	√
F065CH	Configuration/ Interface / End Point Descriptor Register 121	UF0CIE121	R/W	–	√	–	Undefined	√	√
F065DH	Configuration/ Interface / End Point Descriptor Register 122	UF0CIE122	R/W	–	√	–	Undefined	√	√
F065EH	Configuration/ Interface / End Point Descriptor Register 123	UF0CIE123	R/W	–	√	–	Undefined	√	√
F065FH	Configuration/ Interface / End Point Descriptor Register 124	UF0CIE124	R/W	–	√	–	Undefined	√	√
F0660H	Configuration/ Interface / End Point Descriptor Register 125	UF0CIE125	R/W	–	√	–	Undefined	√	√
F0661H	Configuration/ Interface / End Point Descriptor Register 126	UF0CIE126	R/W	–	√	–	Undefined	√	√
F0662H	Configuration/ Interface / End Point Descriptor Register 127	UF0CIE127	R/W	–	√	–	Undefined	√	√
F0663H	Configuration/ Interface / End Point Descriptor Register 128	UF0CIE128	R/W	–	√	–	Undefined	√	√
F0664H	Configuration/ Interface / End Point Descriptor Register 129	UF0CIE129	R/W	–	√	–	Undefined	√	√
F0665H	Configuration/ Interface / End Point Descriptor Register 130	UF0CIE130	R/W	–	√	–	Undefined	√	√
F0666H	Configuration/ Interface / End Point Descriptor Register 131	UF0CIE131	R/W	–	√	–	Undefined	√	√
F0667H	Configuration/ Interface / End Point Descriptor Register 132	UF0CIE132	R/W	–	√	–	Undefined	√	√
F0668H	Configuration/ Interface / End Point Descriptor Register 133	UF0CIE133	R/W	–	√	–	Undefined	√	√
F0669H	Configuration/ Interface / End Point Descriptor Register 134	UF0CIE134	R/W	–	√	–	Undefined	√	√
F066AH	Configuration/ Interface / End Point Descriptor Register 135	UF0CIE135	R/W	–	√	–	Undefined	√	√
F066BH	Configuration/ Interface / End Point Descriptor Register 136	UF0CIE136	R/W	–	√	–	Undefined	√	√
F066CH	Configuration/ Interface / End Point Descriptor Register 137	UF0CIE137	R/W	–	√	–	Undefined	√	√
F066DH	Configuration/ Interface / End Point Descriptor Register 138	UF0CIE138	R/W	–	√	–	Undefined	√	√
F066EH	Configuration/ Interface / End Point Descriptor Register 139	UF0CIE139	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (15/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F066FH	Configuration/ Interface / End Point Descriptor Register 140	UF0CIE140	R/W	–	√	–	Undefined	√	√
F0670H	Configuration/ Interface / End Point Descriptor Register 141	UF0CIE141	R/W	–	√	–	Undefined	√	√
F0671H	Configuration/ Interface / End Point Descriptor Register 142	UF0CIE142	R/W	–	√	–	Undefined	√	√
F0672H	Configuration/ Interface / End Point Descriptor Register 143	UF0CIE143	R/W	–	√	–	Undefined	√	√
F0673H	Configuration/ Interface / End Point Descriptor Register 144	UF0CIE144	R/W	–	√	–	Undefined	√	√
F0674H	Configuration/ Interface / End Point Descriptor Register 145	UF0CIE145	R/W	–	√	–	Undefined	√	√
F0675H	Configuration/ Interface / End Point Descriptor Register 146	UF0CIE146	R/W	–	√	–	Undefined	√	√
F0676H	Configuration/ Interface / End Point Descriptor Register 147	UF0CIE147	R/W	–	√	–	Undefined	√	√
F0677H	Configuration/ Interface / End Point Descriptor Register 148	UF0CIE148	R/W	–	√	–	Undefined	√	√
F0678H	Configuration/ Interface / End Point Descriptor Register 149	UF0CIE149	R/W	–	√	–	Undefined	√	√
F0679H	Configuration/ Interface / End Point Descriptor Register 150	UF0CIE150	R/W	–	√	–	Undefined	√	√
F067AH	Configuration/ Interface / End Point Descriptor Register 151	UF0CIE151	R/W	–	√	–	Undefined	√	√
F067BH	Configuration/ Interface / End Point Descriptor Register 152	UF0CIE152	R/W	–	√	–	Undefined	√	√
F067CH	Configuration/ Interface / End Point Descriptor Register 153	UF0CIE153	R/W	–	√	–	Undefined	√	√
F067DH	Configuration/ Interface / End Point Descriptor Register 154	UF0CIE154	R/W	–	√	–	Undefined	√	√
F067EH	Configuration/ Interface / End Point Descriptor Register 155	UF0CIE155	R/W	–	√	–	Undefined	√	√
F067FH	Configuration/ Interface / End Point Descriptor Register 156	UF0CIE156	R/W	–	√	–	Undefined	√	√
F0680H	Configuration/ Interface / End Point Descriptor Register 157	UF0CIE157	R/W	–	√	–	Undefined	√	√
F0681H	Configuration/ Interface / End Point Descriptor Register 158	UF0CIE158	R/W	–	√	–	Undefined	√	√
F0682H	Configuration/ Interface / End Point Descriptor Register 159	UF0CIE159	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (16/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F0683H	Configuration/ Interface / End Point Descriptor Register 160	UF0CIE160	R/W	–	√	–	Undefined	√	√
F0684H	Configuration/ Interface / End Point Descriptor Register 161	UF0CIE161	R/W	–	√	–	Undefined	√	√
F0685H	Configuration/ Interface / End Point Descriptor Register 162	UF0CIE162	R/W	–	√	–	Undefined	√	√
F0686H	Configuration/ Interface / End Point Descriptor Register 163	UF0CIE163	R/W	–	√	–	Undefined	√	√
F0687H	Configuration/ Interface / End Point Descriptor Register 164	UF0CIE164	R/W	–	√	–	Undefined	√	√
F0688H	Configuration/ Interface / End Point Descriptor Register 165	UF0CIE165	R/W	–	√	–	Undefined	√	√
F0689H	Configuration/ Interface / End Point Descriptor Register 166	UF0CIE166	R/W	–	√	–	Undefined	√	√
F068AH	Configuration/ Interface / End Point Descriptor Register 167	UF0CIE167	R/W	–	√	–	Undefined	√	√
F068BH	Configuration/ Interface / End Point Descriptor Register 168	UF0CIE168	R/W	–	√	–	Undefined	√	√
F068CH	Configuration/ Interface / End Point Descriptor Register 169	UF0CIE169	R/W	–	√	–	Undefined	√	√
F068DH	Configuration/ Interface / End Point Descriptor Register 170	UF0CIE170	R/W	–	√	–	Undefined	√	√
F068EH	Configuration/ Interface / End Point Descriptor Register 171	UF0CIE171	R/W	–	√	–	Undefined	√	√
F068FH	Configuration/ Interface / End Point Descriptor Register 172	UF0CIE172	R/W	–	√	–	Undefined	√	√
F0690H	Configuration/ Interface / End Point Descriptor Register 173	UF0CIE173	R/W	–	√	–	Undefined	√	√
F0691H	Configuration/ Interface / End Point Descriptor Register 174	UF0CIE174	R/W	–	√	–	Undefined	√	√
F0692H	Configuration/ Interface / End Point Descriptor Register 175	UF0CIE175	R/W	–	√	–	Undefined	√	√
F0693H	Configuration/ Interface / End Point Descriptor Register 176	UF0CIE176	R/W	–	√	–	Undefined	√	√
F0694H	Configuration/ Interface / End Point Descriptor Register 177	UF0CIE177	R/W	–	√	–	Undefined	√	√
F0695H	Configuration/ Interface / End Point Descriptor Register 178	UF0CIE178	R/W	–	√	–	Undefined	√	√
F0696H	Configuration/ Interface / End Point Descriptor Register 179	UF0CIE179	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (17/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F0697H	Configuration/ Interface / End Point Descriptor Register 180	UF0CIE180	R/W	–	√	–	Undefined	√	√
F0698H	Configuration/ Interface / End Point Descriptor Register 181	UF0CIE181	R/W	–	√	–	Undefined	√	√
F0699H	Configuration/ Interface / End Point Descriptor Register 182	UF0CIE182	R/W	–	√	–	Undefined	√	√
F069AH	Configuration/ Interface / End Point Descriptor Register 183	UF0CIE183	R/W	–	√	–	Undefined	√	√
F069BH	Configuration/ Interface / End Point Descriptor Register 184	UF0CIE184	R/W	–	√	–	Undefined	√	√
F069CH	Configuration/ Interface / End Point Descriptor Register 185	UF0CIE185	R/W	–	√	–	Undefined	√	√
F069DH	Configuration/ Interface / End Point Descriptor Register 186	UF0CIE186	R/W	–	√	–	Undefined	√	√
F069EH	Configuration/ Interface / End Point Descriptor Register 187	UF0CIE187	R/W	–	√	–	Undefined	√	√
F069FH	Configuration/ Interface / End Point Descriptor Register 188	UF0CIE188	R/W	–	√	–	Undefined	√	√
F06A0H	Configuration/ Interface / End Point Descriptor Register 189	UF0CIE189	R/W	–	√	–	Undefined	√	√
F06A1H	Configuration/ Interface / End Point Descriptor Register 190	UF0CIE190	R/W	–	√	–	Undefined	√	√
F06A2H	Configuration/ Interface / End Point Descriptor Register 191	UF0CIE191	R/W	–	√	–	Undefined	√	√
F06A3H	Configuration/ Interface / End Point Descriptor Register 192	UF0CIE192	R/W	–	√	–	Undefined	√	√
F06A4H	Configuration/ Interface / End Point Descriptor Register 193	UF0CIE193	R/W	–	√	–	Undefined	√	√
F06A5H	Configuration/ Interface / End Point Descriptor Register 194	UF0CIE194	R/W	–	√	–	Undefined	√	√
F06A6H	Configuration/ Interface / End Point Descriptor Register 195	UF0CIE195	R/W	–	√	–	Undefined	√	√
F06A7H	Configuration/ Interface / End Point Descriptor Register 196	UF0CIE196	R/W	–	√	–	Undefined	√	√
F06A8H	Configuration/ Interface / End Point Descriptor Register 197	UF0CIE197	R/W	–	√	–	Undefined	√	√
F06A9H	Configuration/ Interface / End Point Descriptor Register 198	UF0CIE198	R/W	–	√	–	Undefined	√	√
F06AAH	Configuration/ Interface / End Point Descriptor Register 199	UF0CIE199	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (18/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F06ABH	Configuration/ Interface / End Point Descriptor Register 200	UF0CIE200	R/W	–	√	–	Undefined	√	√
F06ACH	Configuration/ Interface / End Point Descriptor Register 201	UF0CIE201	R/W	–	√	–	Undefined	√	√
F06ADH	Configuration/ Interface / End Point Descriptor Register 202	UF0CIE202	R/W	–	√	–	Undefined	√	√
F06AEH	Configuration/ Interface / End Point Descriptor Register 203	UF0CIE203	R/W	–	√	–	Undefined	√	√
F06AFH	Configuration/ Interface / End Point Descriptor Register 204	UF0CIE204	R/W	–	√	–	Undefined	√	√
F06B0H	Configuration/ Interface / End Point Descriptor Register 205	UF0CIE205	R/W	–	√	–	Undefined	√	√
F06B1H	Configuration/ Interface / End Point Descriptor Register 206	UF0CIE206	R/W	–	√	–	Undefined	√	√
F06B2H	Configuration/ Interface / End Point Descriptor Register 207	UF0CIE207	R/W	–	√	–	Undefined	√	√
F06B3H	Configuration/ Interface / End Point Descriptor Register 208	UF0CIE208	R/W	–	√	–	Undefined	√	√
F06B4H	Configuration/ Interface / End Point Descriptor Register 209	UF0CIE209	R/W	–	√	–	Undefined	√	√
F06B5H	Configuration/ Interface / End Point Descriptor Register 210	UF0CIE210	R/W	–	√	–	Undefined	√	√
F06B6H	Configuration/ Interface / End Point Descriptor Register 211	UF0CIE211	R/W	–	√	–	Undefined	√	√
F06B7H	Configuration/ Interface / End Point Descriptor Register 212	UF0CIE212	R/W	–	√	–	Undefined	√	√
F06B8H	Configuration/ Interface / End Point Descriptor Register 213	UF0CIE213	R/W	–	√	–	Undefined	√	√
F06B9H	Configuration/ Interface / End Point Descriptor Register 214	UF0CIE214	R/W	–	√	–	Undefined	√	√
F06BAH	Configuration/ Interface / End Point Descriptor Register 215	UF0CIE215	R/W	–	√	–	Undefined	√	√
F06BBH	Configuration/ Interface / End Point Descriptor Register 216	UF0CIE216	R/W	–	√	–	Undefined	√	√
F06BCH	Configuration/ Interface / End Point Descriptor Register 217	UF0CIE217	R/W	–	√	–	Undefined	√	√
F06BDH	Configuration/ Interface / End Point Descriptor Register 218	UF0CIE218	R/W	–	√	–	Undefined	√	√
F06BEH	Configuration/ Interface / End Point Descriptor Register 219	UF0CIE219	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (19/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F06BFH	Configuration/ Interface / End Point Descriptor Register 220	UF0CIE220	R/W	–	√	–	Undefined	√	√
F06C0H	Configuration/ Interface / End Point Descriptor Register 221	UF0CIE221	R/W	–	√	–	Undefined	√	√
F06C1H	Configuration/ Interface / End Point Descriptor Register 222	UF0CIE222	R/W	–	√	–	Undefined	√	√
F06C2H	Configuration/ Interface / End Point Descriptor Register 223	UF0CIE223	R/W	–	√	–	Undefined	√	√
F06C3H	Configuration/ Interface / End Point Descriptor Register 224	UF0CIE224	R/W	–	√	–	Undefined	√	√
F06C4H	Configuration/ Interface / End Point Descriptor Register 225	UF0CIE225	R/W	–	√	–	Undefined	√	√
F06C5H	Configuration/ Interface / End Point Descriptor Register 226	UF0CIE226	R/W	–	√	–	Undefined	√	√
F06C6H	Configuration/ Interface / End Point Descriptor Register 227	UF0CIE227	R/W	–	√	–	Undefined	√	√
F06C7H	Configuration/ Interface / End Point Descriptor Register 228	UF0CIE228	R/W	–	√	–	Undefined	√	√
F06C8H	Configuration/ Interface / End Point Descriptor Register 229	UF0CIE229	R/W	–	√	–	Undefined	√	√
F06C9H	Configuration/ Interface / End Point Descriptor Register 230	UF0CIE230	R/W	–	√	–	Undefined	√	√
F06CAH	Configuration/ Interface / End Point Descriptor Register 231	UF0CIE231	R/W	–	√	–	Undefined	√	√
F06CBH	Configuration/ Interface / End Point Descriptor Register 232	UF0CIE232	R/W	–	√	–	Undefined	√	√
F06CCH	Configuration/ Interface / End Point Descriptor Register 233	UF0CIE233	R/W	–	√	–	Undefined	√	√
F06CDH	Configuration/ Interface / End Point Descriptor Register 234	UF0CIE234	R/W	–	√	–	Undefined	√	√
F06CEH	Configuration/ Interface / End Point Descriptor Register 235	UF0CIE235	R/W	–	√	–	Undefined	√	√
F06CFH	Configuration/ Interface / End Point Descriptor Register 236	UF0CIE236	R/W	–	√	–	Undefined	√	√
F06D0H	Configuration/ Interface / End Point Descriptor Register 237	UF0CIE237	R/W	–	√	–	Undefined	√	√
F06D1H	Configuration/ Interface / End Point Descriptor Register 238	UF0CIE238	R/W	–	√	–	Undefined	√	√
F06D2H	Configuration/ Interface / End Point Descriptor Register 239	UF0CIE239	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (20/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F06D3H	Configuration/ Interface / End Point Descriptor Register 240	UF0CIE240	R/W	–	√	–	Undefined	√	√
F06D4H	Configuration/ Interface / End Point Descriptor Register 241	UF0CIE241	R/W	–	√	–	Undefined	√	√
F06D5H	Configuration/ Interface / End Point Descriptor Register 242	UF0CIE242	R/W	–	√	–	Undefined	√	√
F06D6H	Configuration/ Interface / End Point Descriptor Register 243	UF0CIE243	R/W	–	√	–	Undefined	√	√
F06D7H	Configuration/ Interface / End Point Descriptor Register 244	UF0CIE244	R/W	–	√	–	Undefined	√	√
F06D8H	Configuration/ Interface / End Point Descriptor Register 245	UF0CIE245	R/W	–	√	–	Undefined	√	√
F06D9H	Configuration/ Interface / End Point Descriptor Register 246	UF0CIE246	R/W	–	√	–	Undefined	√	√
F06DAH	Configuration/ Interface / End Point Descriptor Register 247	UF0CIE247	R/W	–	√	–	Undefined	√	√
F06DBH	Configuration/ Interface / End Point Descriptor Register 248	UF0CIE248	R/W	–	√	–	Undefined	√	√
F06DCH	Configuration/ Interface / End Point Descriptor Register 249	UF0CIE249	R/W	–	√	–	Undefined	√	√
F06DDH	Configuration/ Interface / End Point Descriptor Register 250	UF0CIE250	R/W	–	√	–	Undefined	√	√
F06DEH	Configuration/ Interface / End Point Descriptor Register 251	UF0CIE251	R/W	–	√	–	Undefined	√	√
F06DFH	Configuration/ Interface / End Point Descriptor Register 252	UF0CIE252	R/W	–	√	–	Undefined	√	√
F06E0H	Configuration/ Interface / End Point Descriptor Register 253	UF0CIE253	R/W	–	√	–	Undefined	√	√
F06E1H	Configuration/ Interface / End Point Descriptor Register 254	UF0CIE254	R/W	–	√	–	Undefined	√	√
F06E2H	Configuration/ Interface / End Point Descriptor Register 255	UF0CIE255	R/W	–	√	–	Undefined	√	√

Table 3-6 Extended SFR (2nd SFR) List (21/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	78K0R/KC3-L	78K0R/KE3-L
				1 Bit	8 Bit	16 Bit			
F06F0H	Active Interface Number Register	UF0AIFN	R/W	–	√	–	00H	√	√
F06F1H	Active Alternative Setting Register	UF0AAS	R/W	–	√	–	00H	√	√
F06F2H	Alternative Setting Status Register	UF0ASS	R	–	√	–	00H	√	√
F06F3H	End Point1 Interface Mapping Register	UF0E1IM	R/W	–	√	–	00H	√	√
F06F4H	End Point2 Interface Mapping Register	UF0E2IM	R/W	–	√	–	00H	√	√
F06F5H	End Point3 Interface Mapping Register	UF0E3IM	R/W	–	√	–	00H	√	√
F06F6H	End Point4 Interface Mapping Register	UF0E4IM	R/W	–	√	–	00H	√	√
F06F9H	End Point7 Interface Mapping Register	UF0E7IM	R/W	–	√	–	00H	√	√
F06FAH	End Point8 Interface Mapping Register	UF0E8IM	R/W	–	√	–	00H	√	√

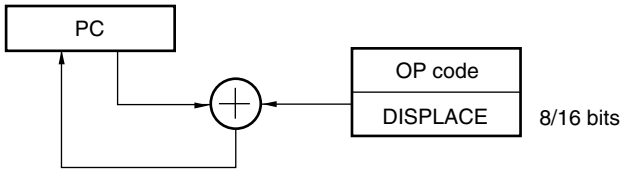
3.3 Instruction Address Addressing

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

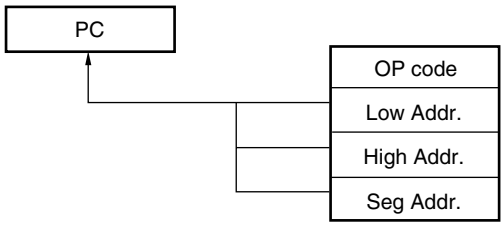
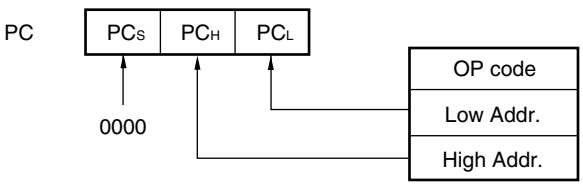


Figure 3-16. Example of CALL !addr16/BR !addr16



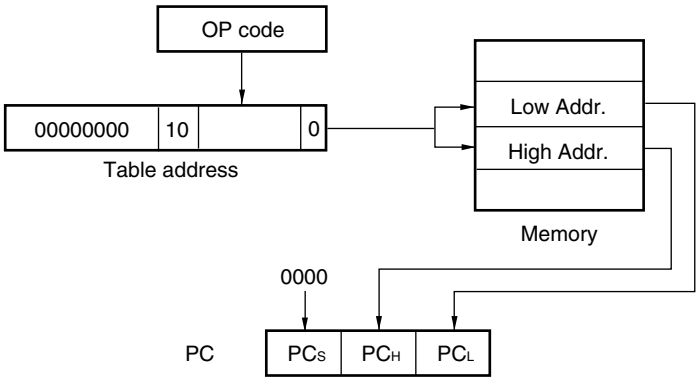
3.3.3 Table indirect addressing

[Function]

Table indirect addressing specifies a table address in the CALLT table area (0080H - 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 - 0FFFFH.

Figure 3-17. Outline of Table Indirect Addressing

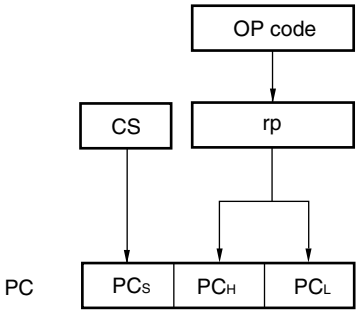


3.3.4 Register direct addressing

[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]

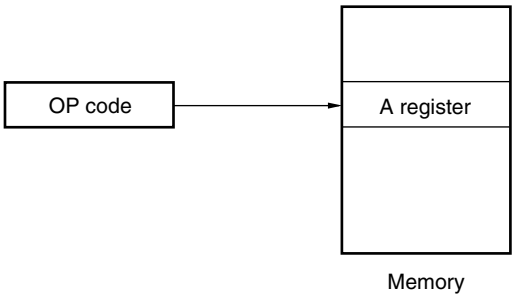
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[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

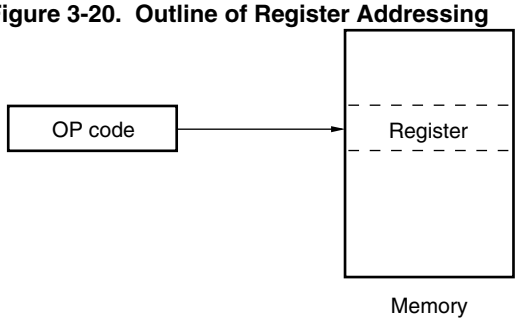
[Function]

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 - 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

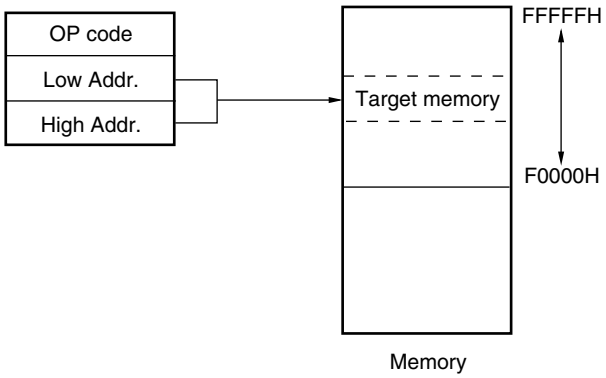
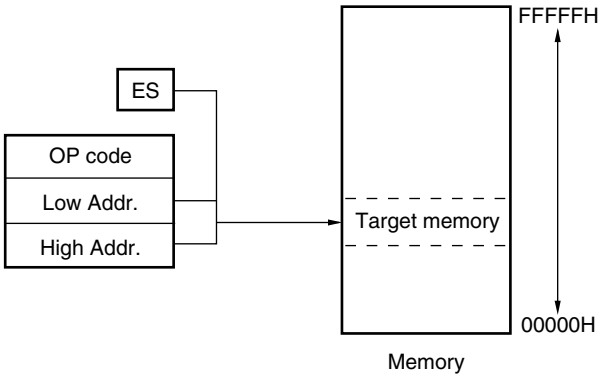


Figure 3-22. Example of ES:ADDR16



3.4.4 Short direct addressing

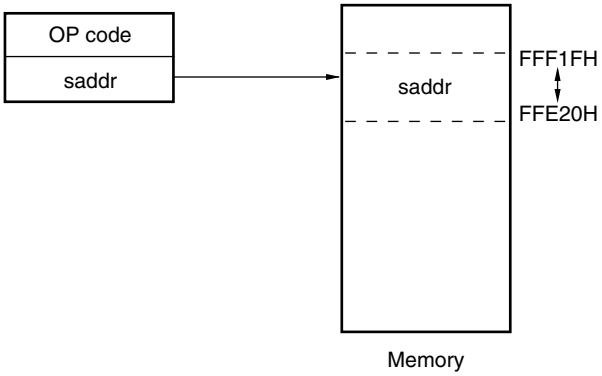
[Function]

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 - FFF1FH.

[Operand format]

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

Figure 3-23. Outline of Short Direct Addressing



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

Regardless of whether 16-bit or 20-bit immediate data is used, addresses within the space from FFE20H - FFF1FH are specified for the memory.

3.4.5 SFR addressing

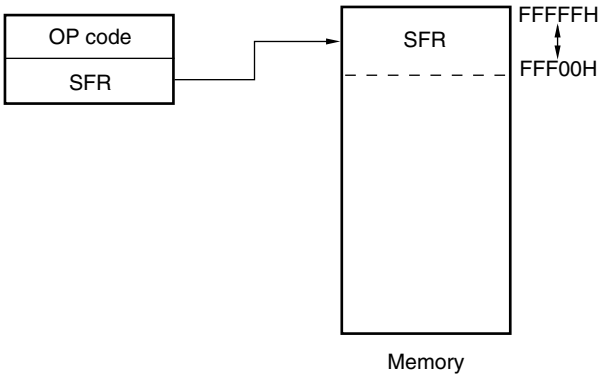
[Function]

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 - FFFFFH.

[Operand format]

Identifier	Description
SFR	SFR name
SFRP	16-bit-manipulable 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 - FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

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

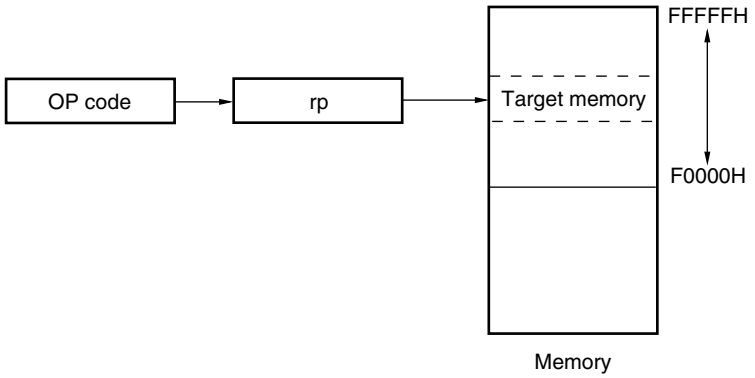
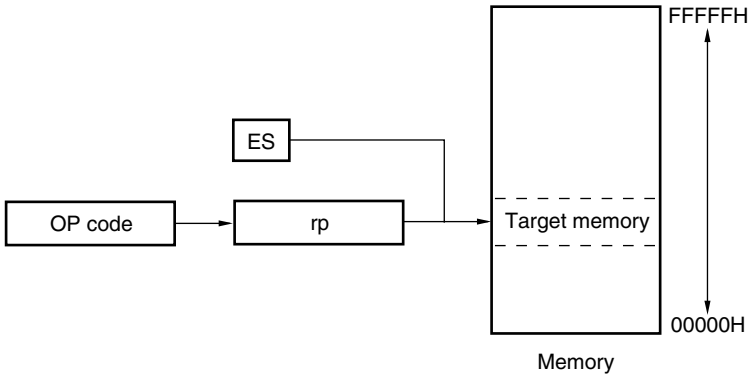


Figure 3-26. Example of ES:[DE], ES:[HL]



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 - FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H - FFFFFH is specifiable)
–	word[BC] (only the space from F0000H - 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]

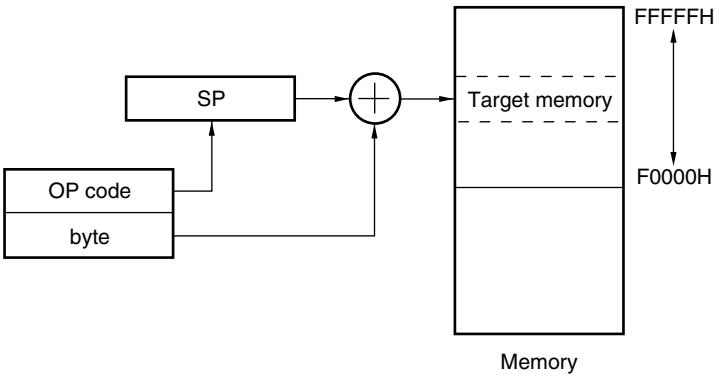


Figure 3-28. Example of [HL + byte], [DE + byte]

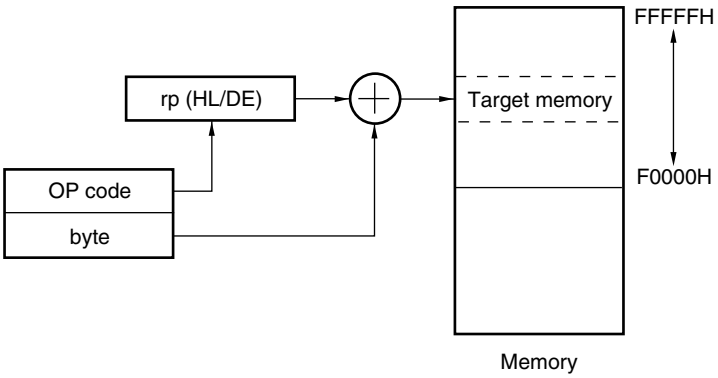


Figure 3-29. Example of word[B], word[C]

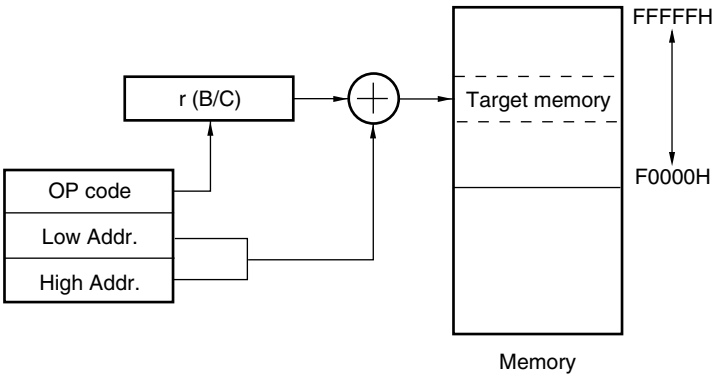


Figure 3-30. Example of word[BC]

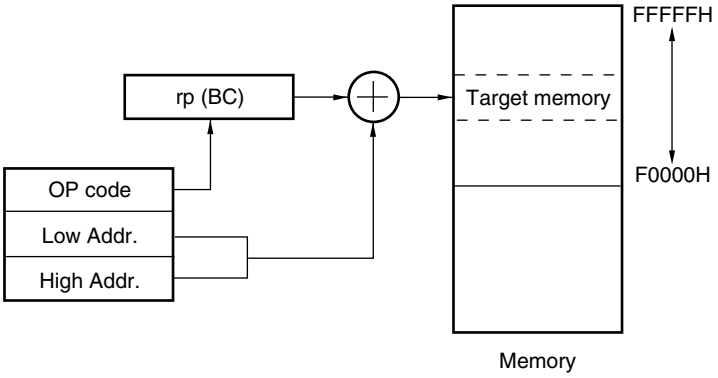


Figure 3-31. Example of ES:[HL + byte], ES:[DE + byte]

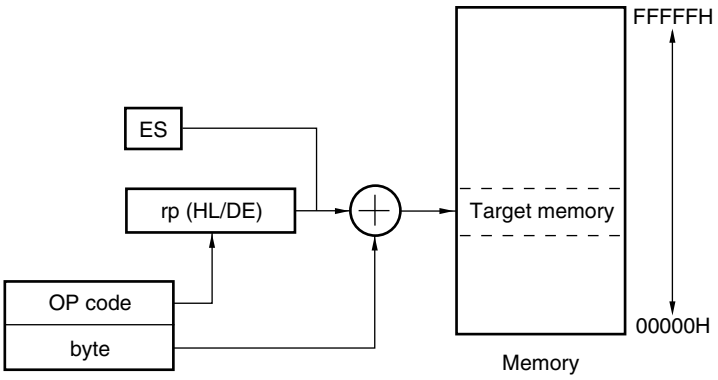


Figure 3-32. Example of ES:word[B], ES:word[C]

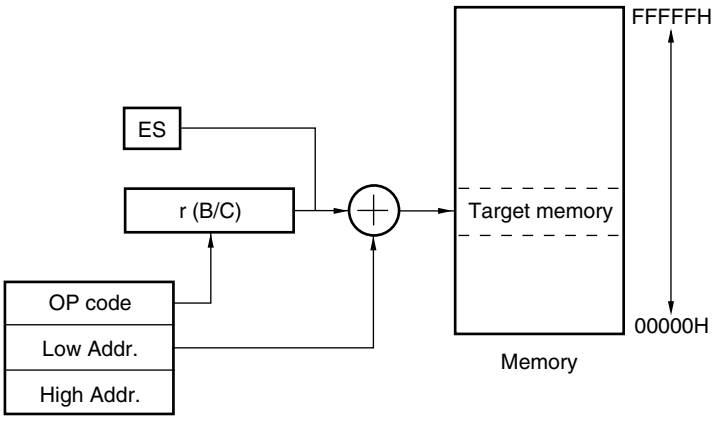
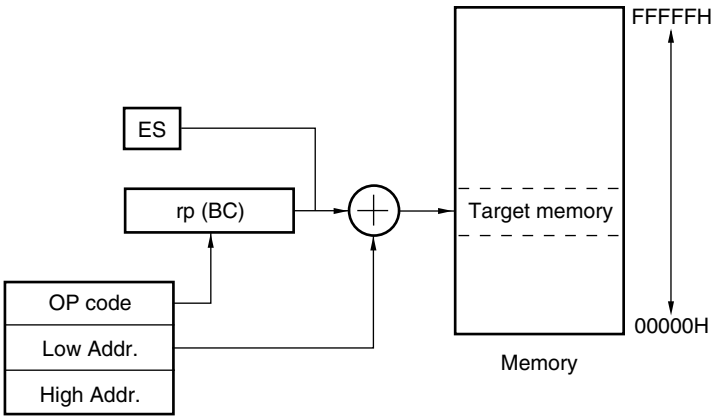


Figure 3-33. Example of ES:word[BC]



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 - 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]

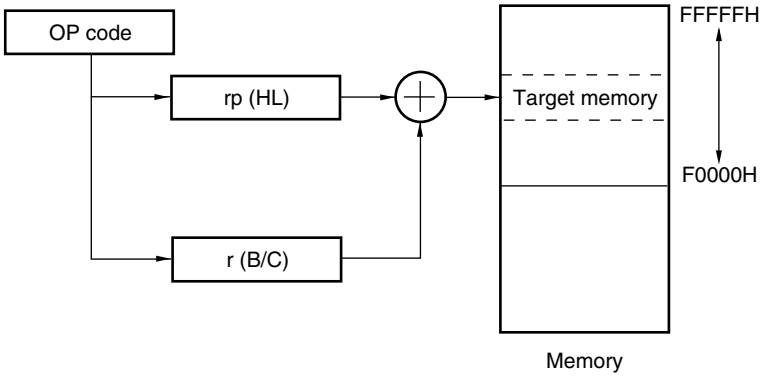
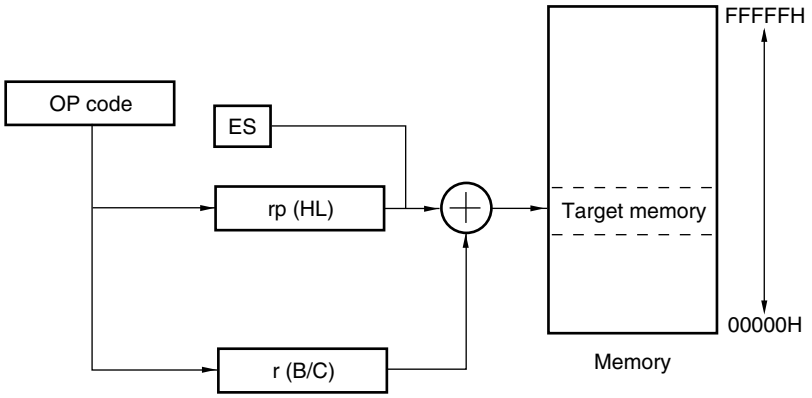


Figure 3-35. Example of ES:[HL+B], ES:[HL+C]



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 three types of pin I/O buffer power supplies: AV_{REF}, 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_{REF}	P20 - P27
EV_{DD}	• Port pins other than P20 - P27 • RESET and FLMD0 pins
V_{DD}	• P121 - P124 • Non-port pins (excluding $\overline{\text{RESET}}$ and FLMD0 pins)

78K0R/KC3-L, 78K0R/KE3-L products are provided with the digital I/O ports, which enable variety of control operations. The functions of each port are shown in Table 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)

KE3-L	KC3-L	Function Name	I/O	Function	After Reset	Alternate Function
√	√	P02	I/O	Port 0. I/O port. Input of P03, P04 can be set to TTL input buffer. Output of P02 - P04 can be set to N-ch open-drain output (V_{DD} tolerance). 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	SO10/TxD1
√	√	P03				SI10/RxD1/SDA10
√	√	P04				$\overline{SCK10}$/SCL10
√	√	P10	I/O	Port 1. I/O port. Input of P10, P11 can be set to TTL input buffer. Output of P10, P12 can be set to N-ch open-drain output (V_{DD} tolerance). 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	$\overline{SCK00}$
√	√	P11				SI00/RxD0
√	√	P12				SO00/TxD0
√	√	P13				TxD3
√	√	P14				RxD3
√	√	P15				RTCDIV/RTCCL
√	√	P16				TI01/TO01/INTP5
√	√	P17				TI02/TO02
√	√	P20 - P27	I/O	Port 2. I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0 - ANI7
√	√	P31	I/O	Port 3. 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/TO03/INTP4
√	√	P40[Note]	I/O	Port 4. 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
√	×	P42				TI04/TO04
√	×	P43				-
√	√	P50	I/O	Port 5. 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	INTP1
√	√	P51				INTP2
√	×	P52				TO00
√	×	P53				TI00
√	√	P60	I/O	Port 6. I/O port. Output of P60 - P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCL0
√	√	P61				SDA0
√	√	P62, P63				-

Note If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in **2.2.5 P40 - P43 (port 4)**).

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

KE3-L	KC3-L	Function Name	I/O	Function	After Reset	Alternate Function
√	√	P70 - P73	I/O	Port 7. 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	KR0 - KR3
√	×	P74 - P77				KR4/INTP8 - KR7/INTP11
√	×	P110, 111	I/O	Port 11. 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	–
√	√	P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. 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	I/O	Port 14. I/O port. Input of P142, P143 can be set to TTL input buffer. Output of P142 - P144 can be set to the N-ch open-drain output (V_{DD} tolerance). 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	PCLBUZ0/INTP6
√	×	P142				$\overline{SCK20}$/SCL20
√	×	P143				SI20/RxD2/SDA20
√	×	P144				SO20/TxD2

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	• 78K0R/KC3-L (48 Pin product) Port mode registers (PM0 - PM7, PM12, PM14) Port registers (P0 - P7, P12, P14) Pull-up resistor option registers (PU0, PU1, PU3 – PU5, PU7, PU12, PU14) Port input mode registers (PIM0, PIM1) Port output mode registers (POM0, POM1) A/D port configuration register (ADPC) • 78K0R/KE3-L (64 Pin product) Port mode registers (PM0 - PM7,PM11, PM12, PM14) Port registers (P0 - P7, P11 - P14) Pull-up resistor option registers (PU0, PU1, PU3 – PU5, PU7,PU11, PU12, PU14) Port input mode registers (PIM0, PIM1, PIM14) Port output mode registers (POM0, POM1, POM14) A/D port configuration register (ADPC)
Port	• 78K0R/KC3-L (48 Pin product) Total: 39 (CMOS I/O: 30, CMOS input: 4, N-ch open drain I/O: 4), USB Buffer Control: 1 • 78K0R/KE3-L (64 Pin product) Total: 53 (CMOS I/O: 43, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4), USB Buffer Control: 1
Pull-up resistor	• 78K0R/KC3-L (48 Pin product) Total: 22 • 78K0R/KE3-L (64 Pin product) Total: 35

4.2.1 Port 0

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P02/SO10/TxD1	√	√
P03/SI10/RxD1/SDA10	√	√
P04/SCK10/SCL10	√	√

It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P02 - P04 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P03, P04 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P02 - P04 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

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

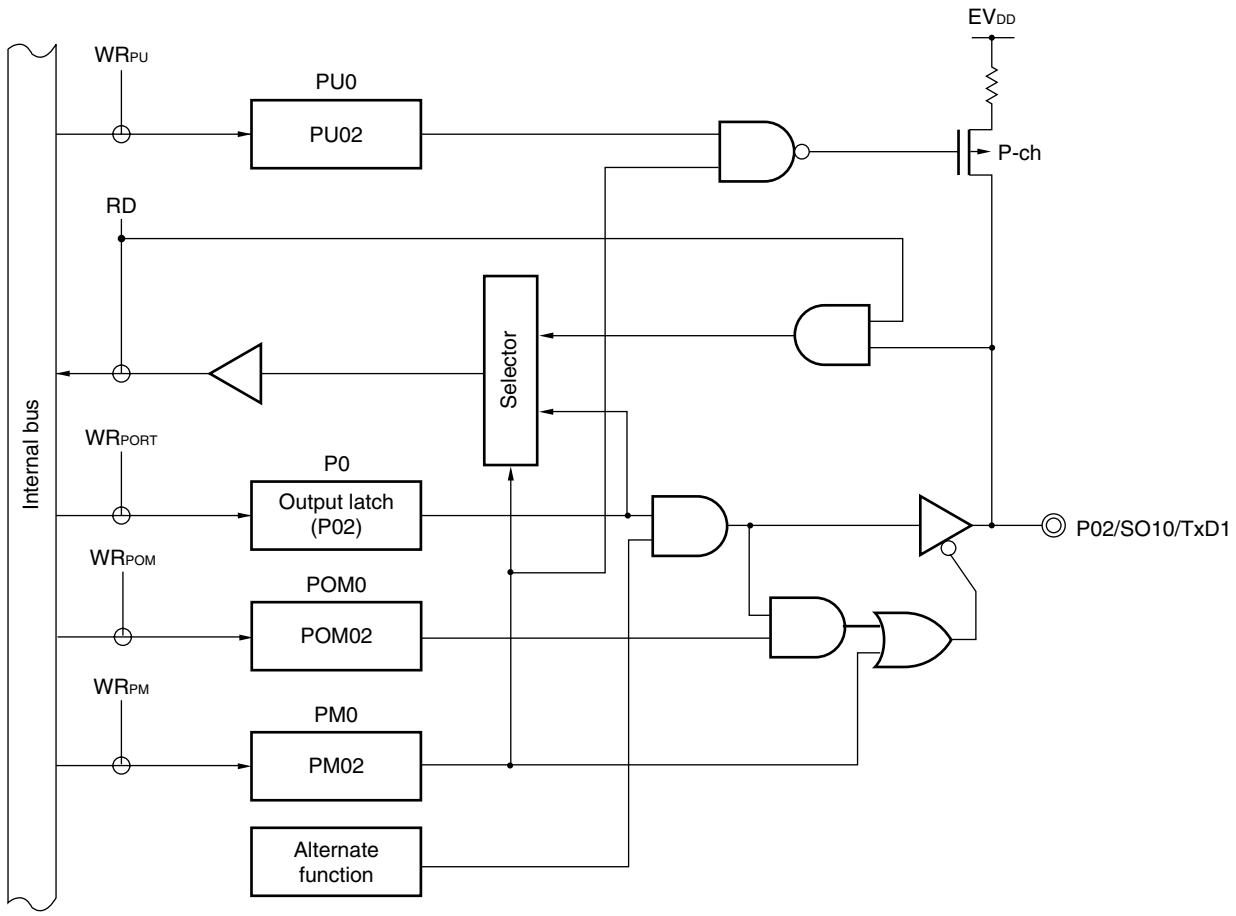
Reset signal generation sets port 0 to input mode.

Figures 4-1, 4-2 show block diagrams of port 0.

Cautions To use P02/SO10/TxD1, P03/SI10/RxD1/SDA10, or P04/$\overline{\text{SCK10}}$/SCL10 as a general-purpose port, note the serial array unit 0 setting. For details, refer to the following tables.

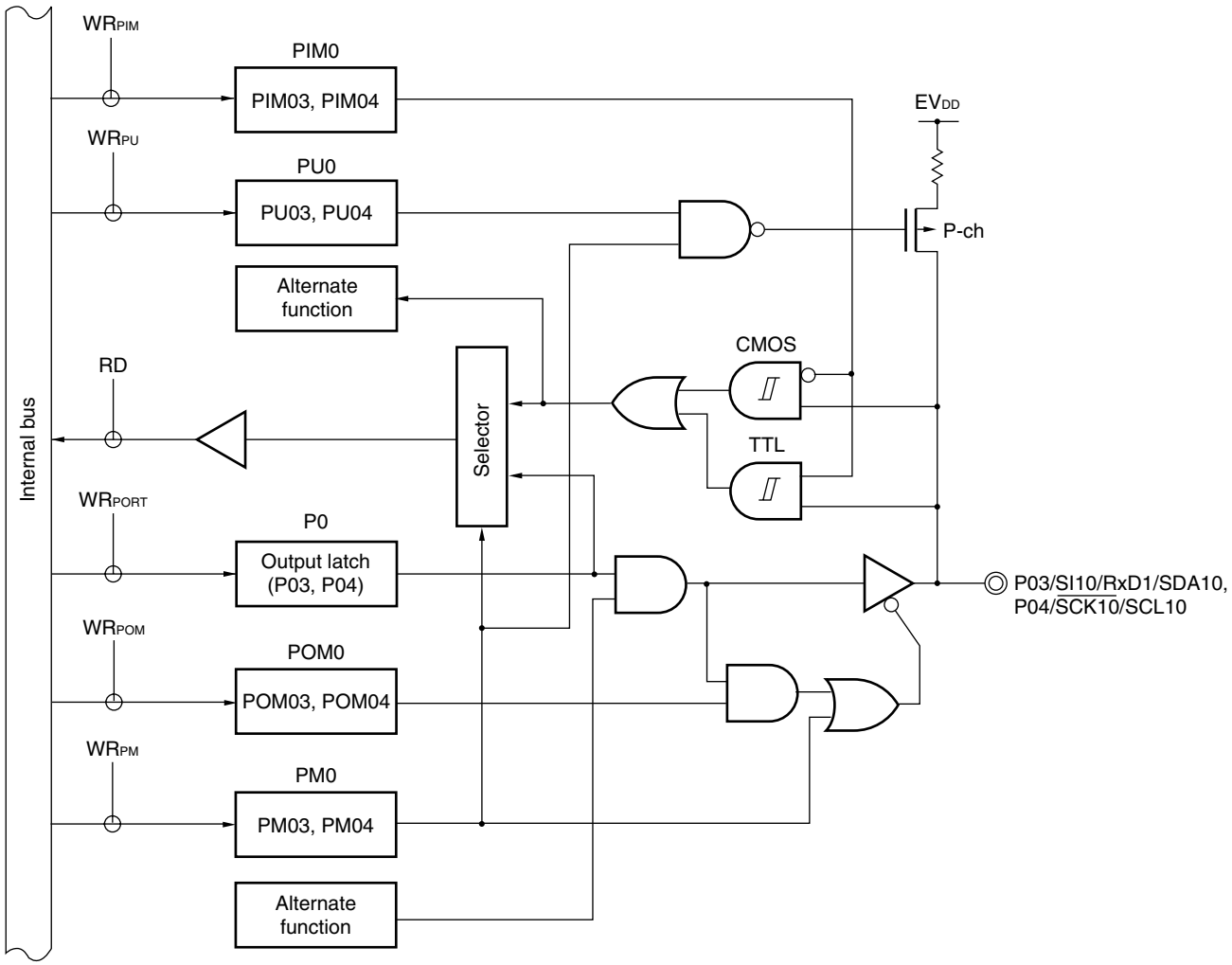
- **Table 11-7 Relationship between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10)**
- **Table 11-8 Relationship between Register Settings and Pins (Channel 3 of Unit 0: UART1 Reception)**

Figure 4-1. Block Diagram of P02



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
POM0: Port output mode register 0
RD: Read signal
WR××: Write signal

Figure 4-2. Block Diagram of P03, P04



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PIM0: Port input mode register 0
POM0: Port output mode register 0
RD: Read signal
WR××: Write signal

4.2.2 Port 1

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P10/ SCK00	√	√
P11/SI00/RxD0	√	√
P12/SO00/TxD0	√	√
P13/TxD3	√	√
P14/RxD3	√	√
P15/RTCDIV/RTCCL	√	√
P16/TI01/TO01/INTP5	√	√
P17/TI02/TO02	√	√

It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 - P17 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 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, P12 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 external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and real-time counter clock output.

Reset signal generation sets port 1 to input mode.

Figure 4-3 - 4-9 show block diagrams of port 1.

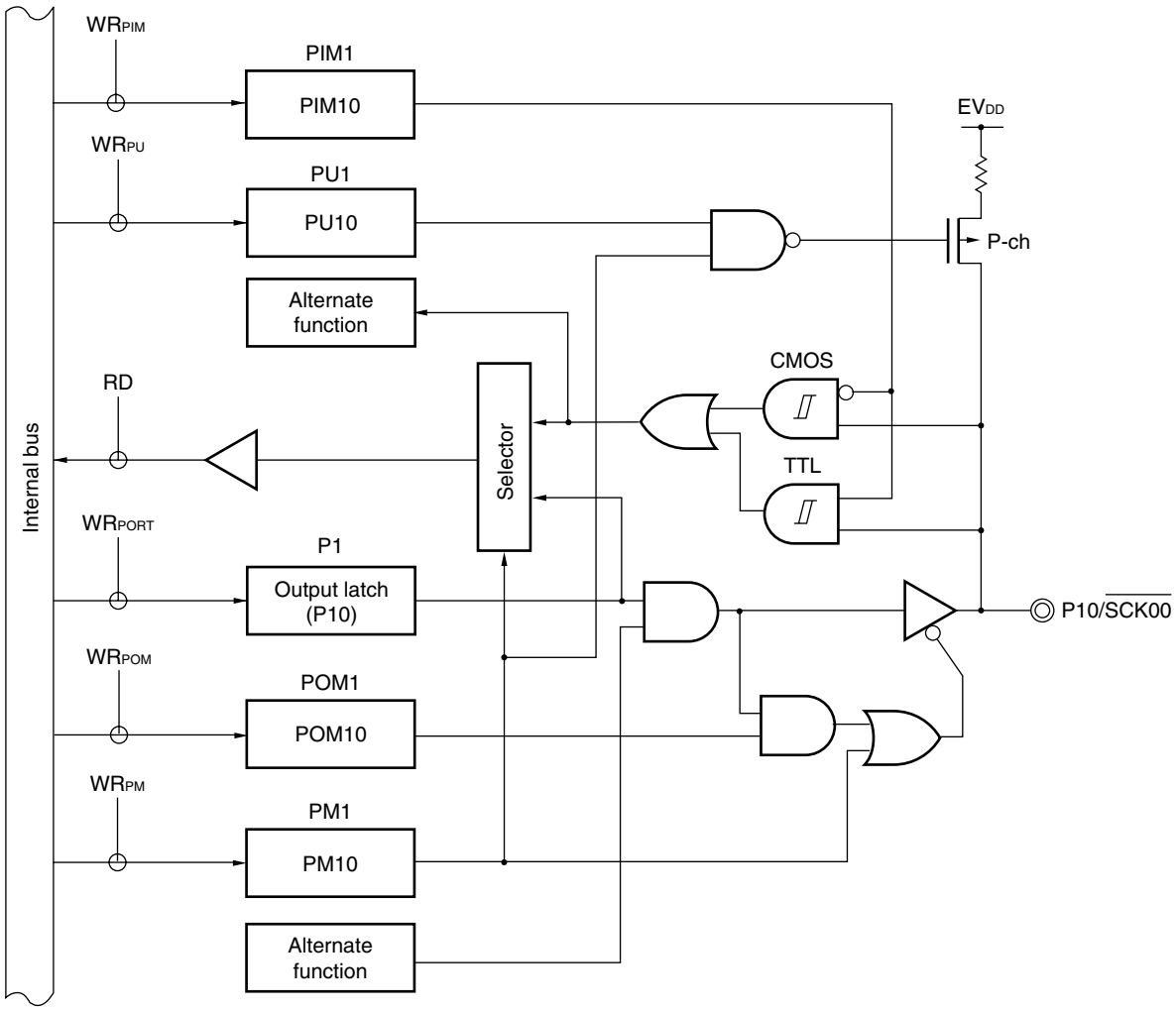
Cautions 1. To use P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, or P12/SO00/TxD0, P13/TxD3, P14/RxD3 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables.

- **Table 11-5 Relationship between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission)**
- **Table 11-6 Relationship between Register Settings and Pins (Channel 1 of Unit 0: UART0 Reception)**
- **Table 11-11 Relationship between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission)**
- **Table 11-12 Relationship between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception)**

2. To use P16/TI01/TO01/INTP5 or P17/TI02/TO02 as a general-purpose port, set bits 1 and 2 (TO01, TO02) of timer output register 0 (TO0) and bits 1 and 2 (TOE01, TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

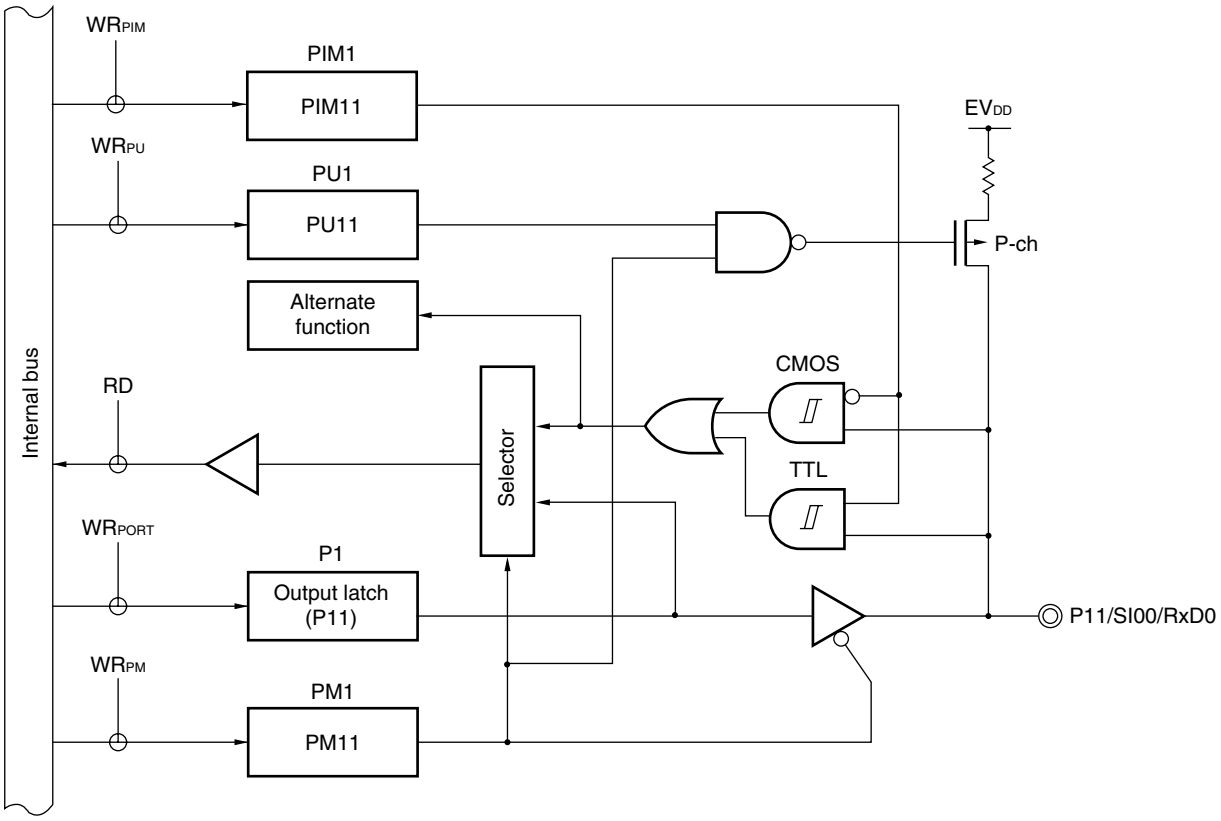
3. To use P15/RTCDIV/RTCCL as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0) and bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2) to "0", which is the same as their default status settings.

Figure 4-3. Block Diagram of P10



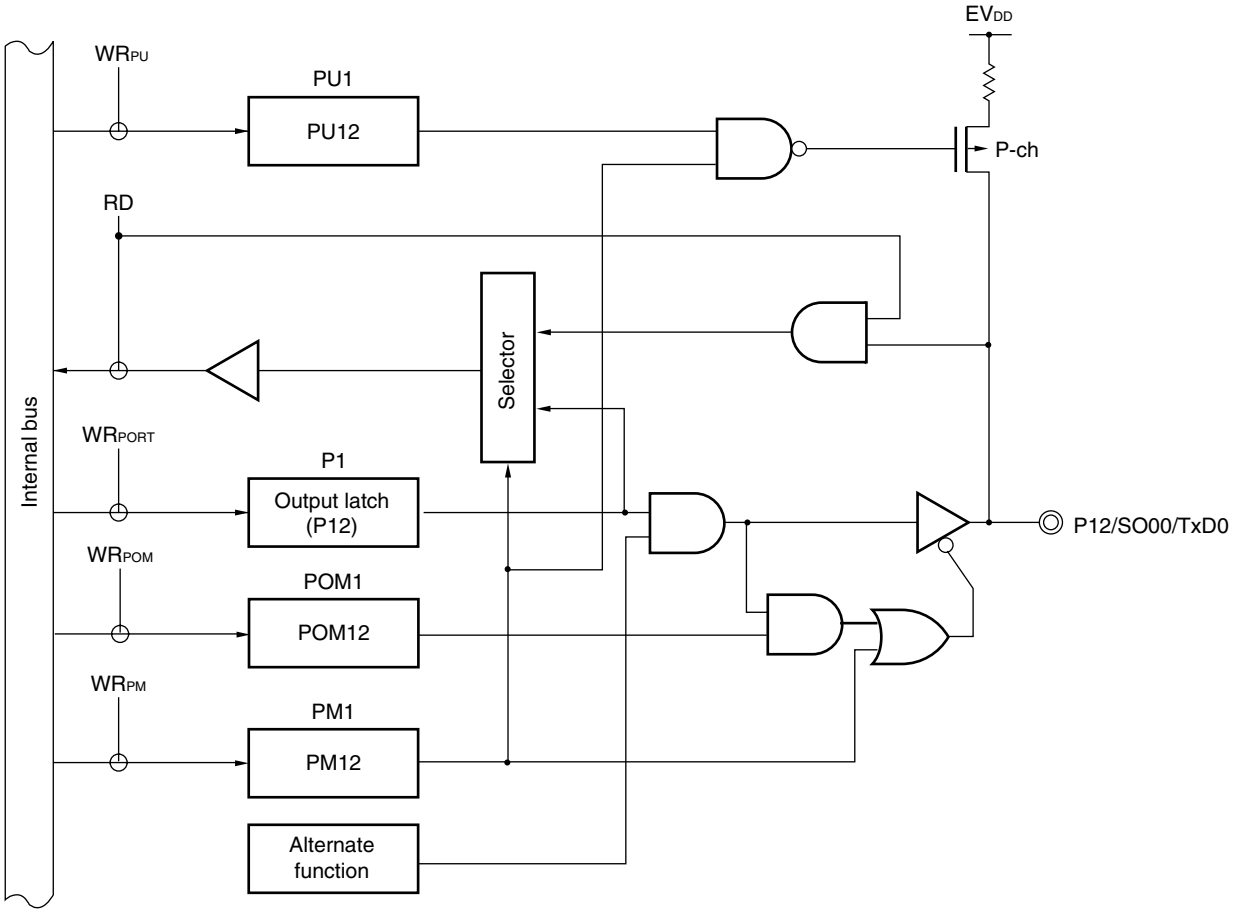
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P11



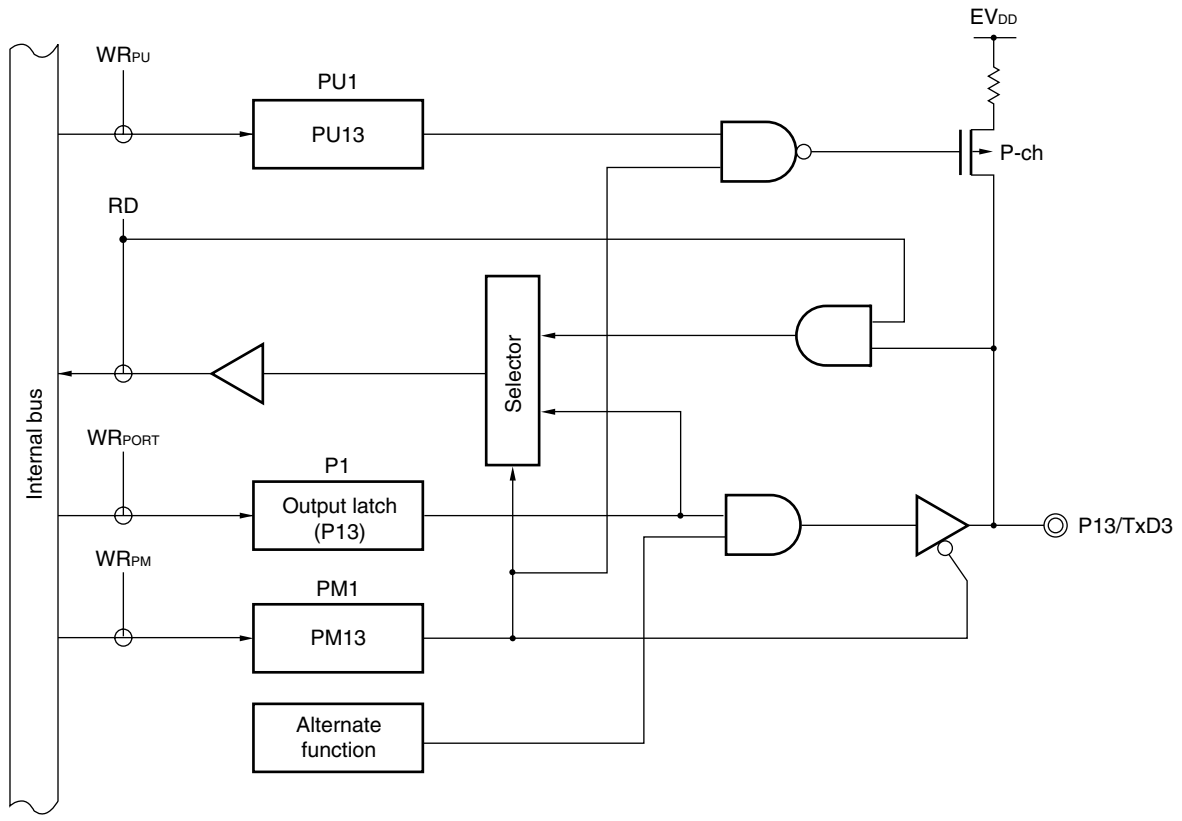
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
RD: Read signal
WR××: Write signal

Figure 4-5. Block Diagram of P12



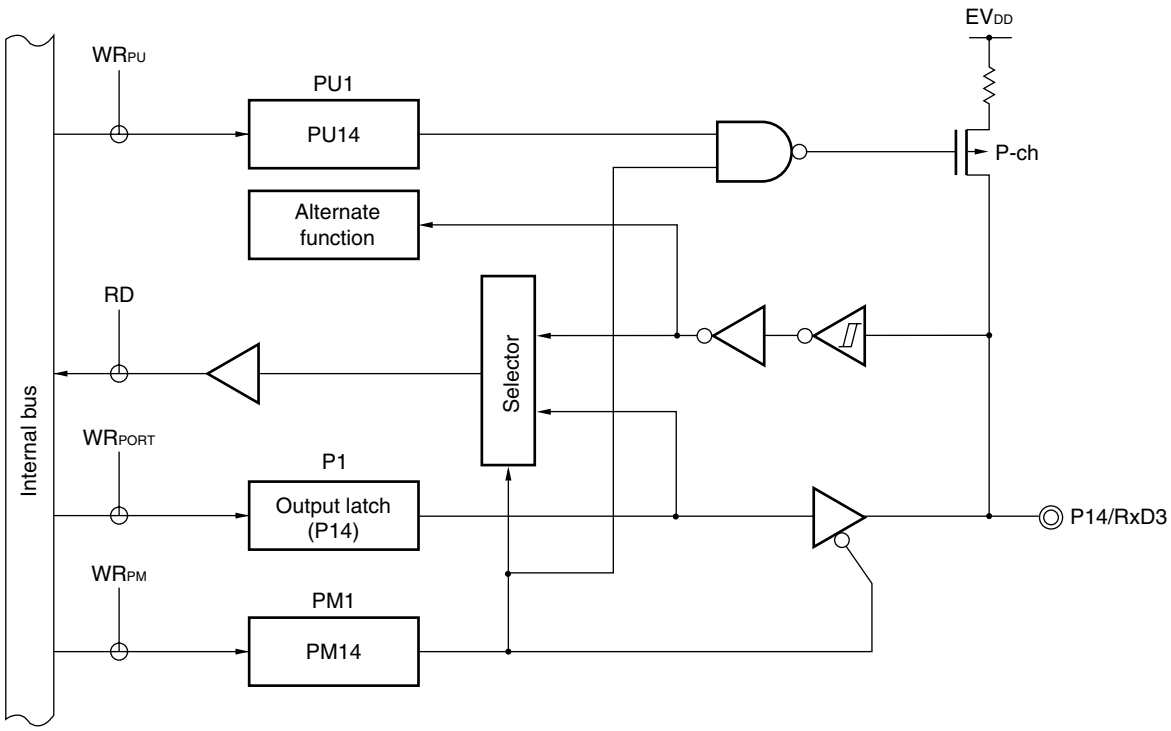
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-6. Block Diagram of P13



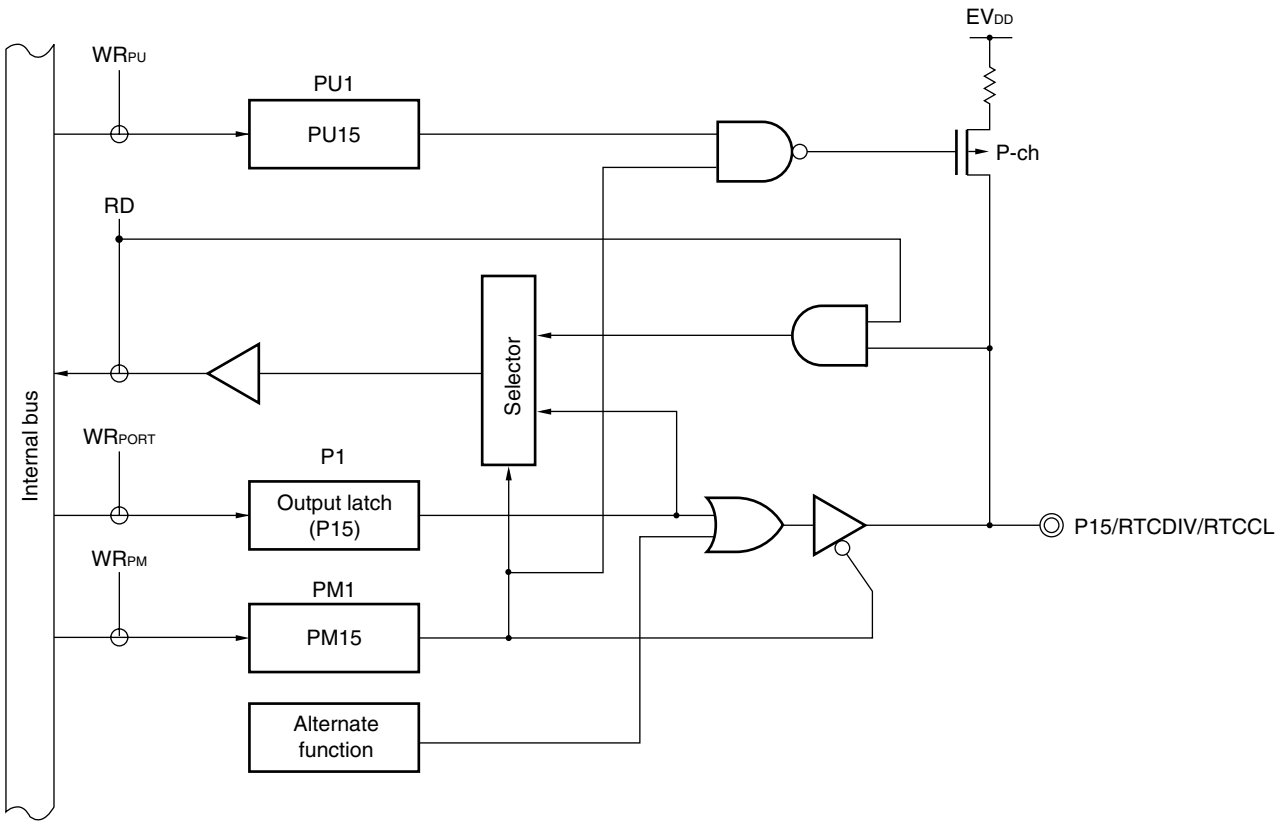
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-7. Block Diagram of P14



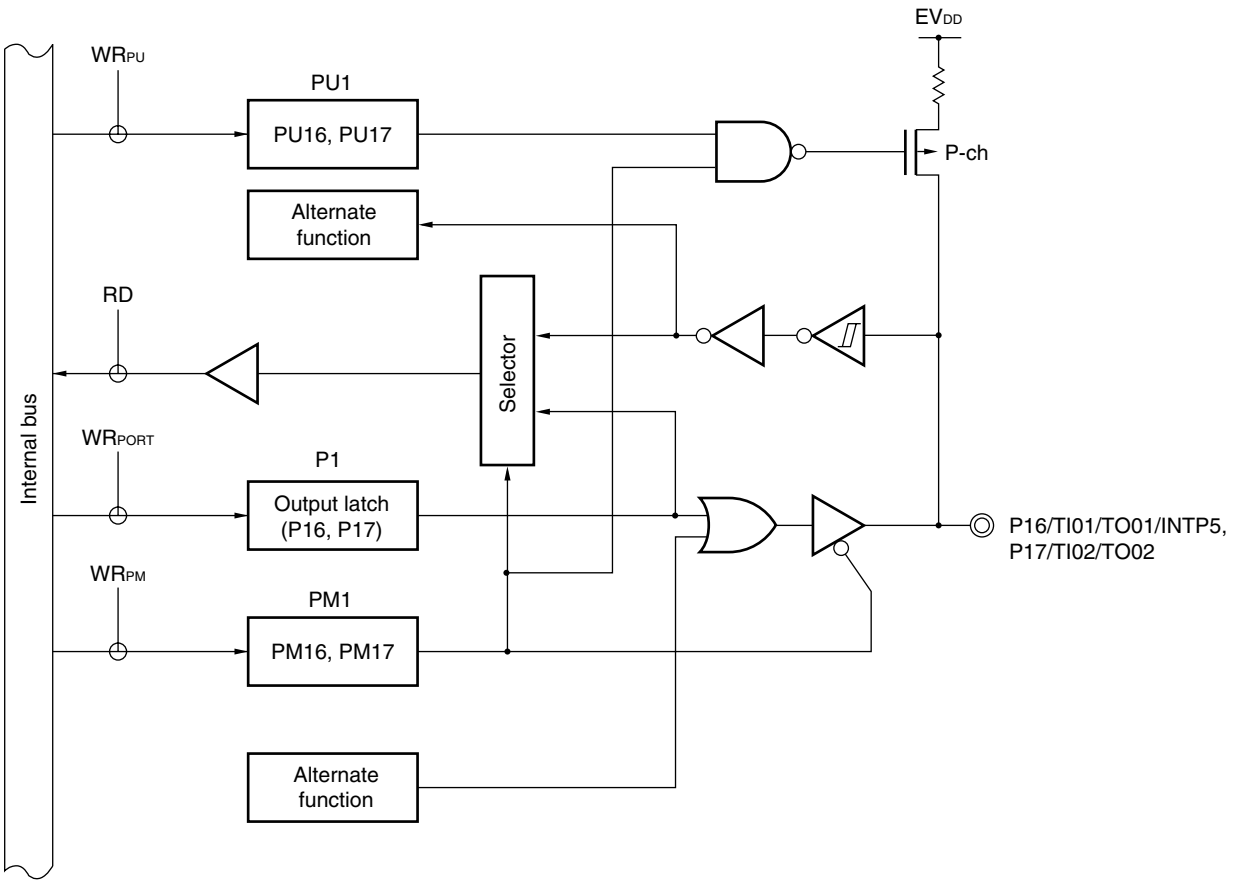
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-8. Block Diagram of P15



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

Figure 4-9. Block Diagram of P16, P17



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

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P20/ANI0	√	√
P21/ANI1	√	√
P22/ANI2	√	√
P23/ANI3	√	√
P24/ANI4	√	√
P25/ANI5	√	√
P26/ANI6	√	√
P27/ANI7	√	√

It is an I/O port with an output latch. It 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.

To use P20/ANI0 - P27/ANI7 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 - P27/ANI7 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 - P27/ANI7 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using PM2. Use these pins starting from the upper bit.

Table 4-4. Setting Functions of P20/ANI0 - P27/ANI7 Pins

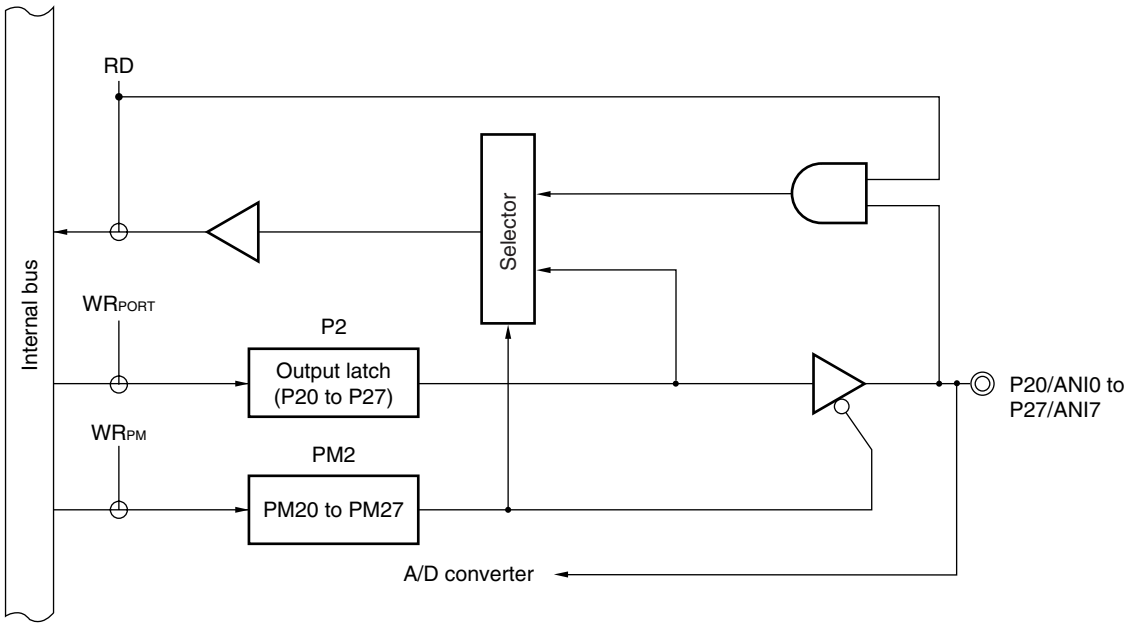
ADPC	PM2	ADS	P20/ANI0 - P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

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

Figure 4-10 shows a block diagram of port 2.

Caution See 2.2.13 AV_{REF} for the voltage to be applied to the AV_{REF} pin when using port 2 as a digital I/O.

Figure 4-10. Block Diagram of P20 - P27



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

4.2.4 Port 3

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P31/TI03/TO03/INTP4	√	√

It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 and P31 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).

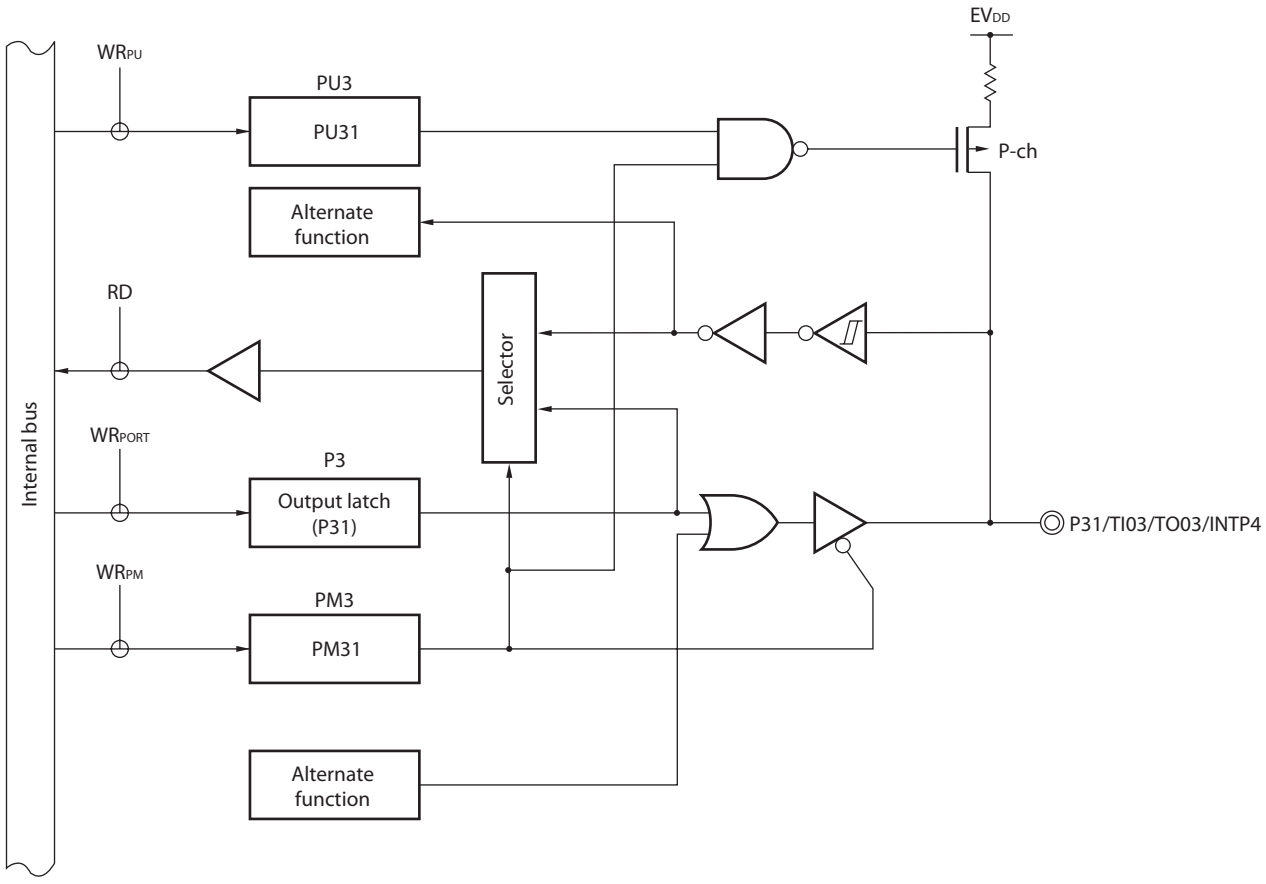
This port can also be used for external interrupt request input, timer I/O.

Reset signal generation sets it to input mode.

Figure 4-11 shows block a diagram of port 3.

Cautions To use P31/TI03/TO03/INTP4 as a general-purpose port, set bit 3 (TO03) of timer output register 0 (TO0) and bit 3 (TOE03) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 4-11. Block Diagram of P31



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

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P40/TOOL0	√	√
P41/TOOL1	√	√
P42/TI04/TO04	–	√
P43	–	√

It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 - P43 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, clock output, and timer I/O.

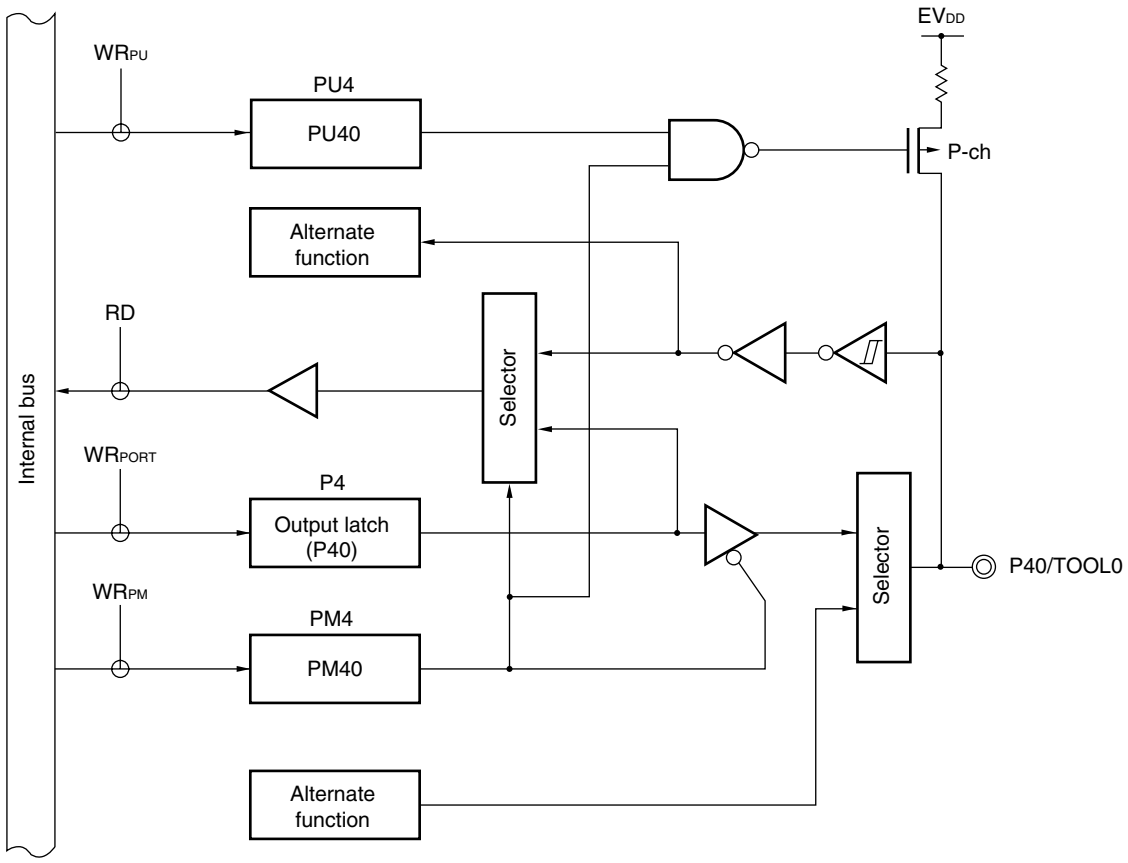
Reset signal generation sets port 4 to input mode.

Figures 4-12 - 4-15 show block diagrams of port 4.

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

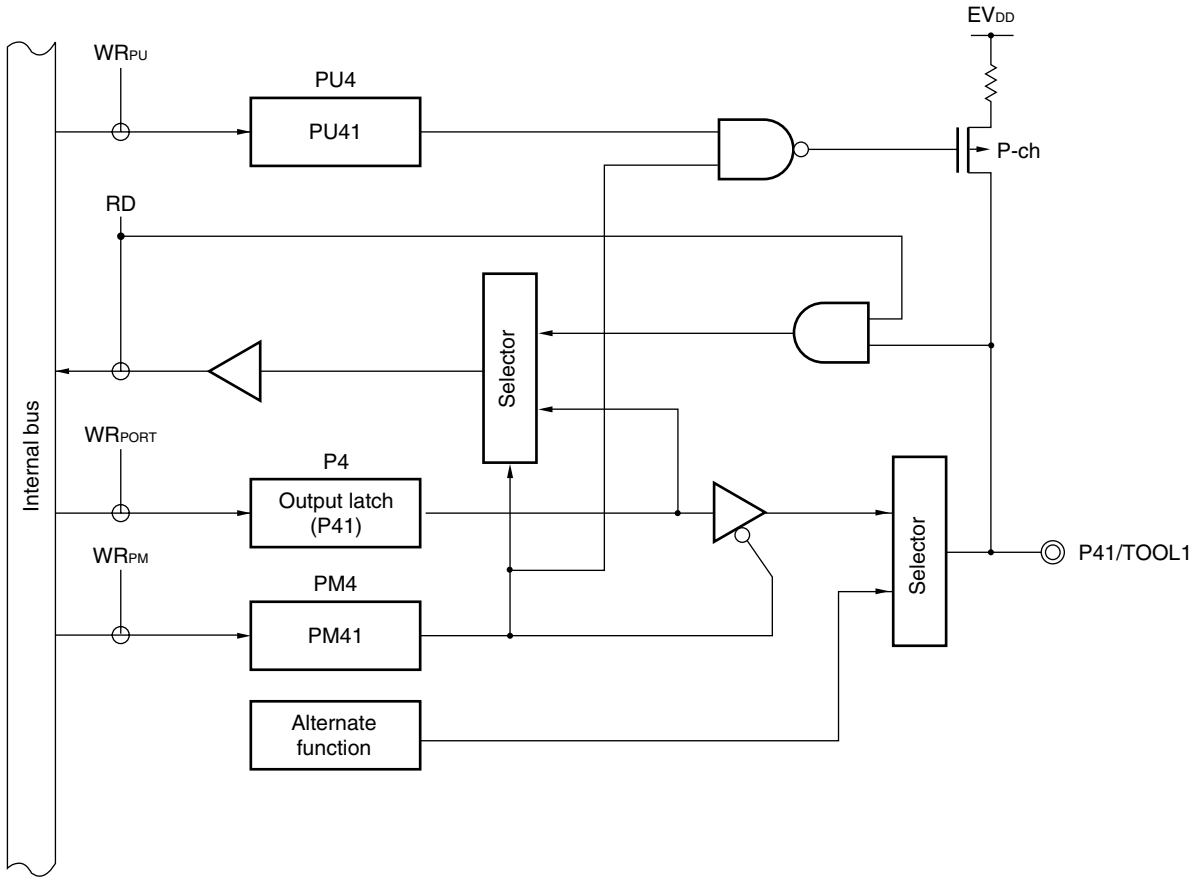
Cautions 1. When a tool is connected, the P40 pin cannot be used as a port pin.
When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger.
1-line mode: can be used as a port (P41).
2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).
2. To use P42/TI04/TO04 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.

Figure 4-12. Block Diagram of P40



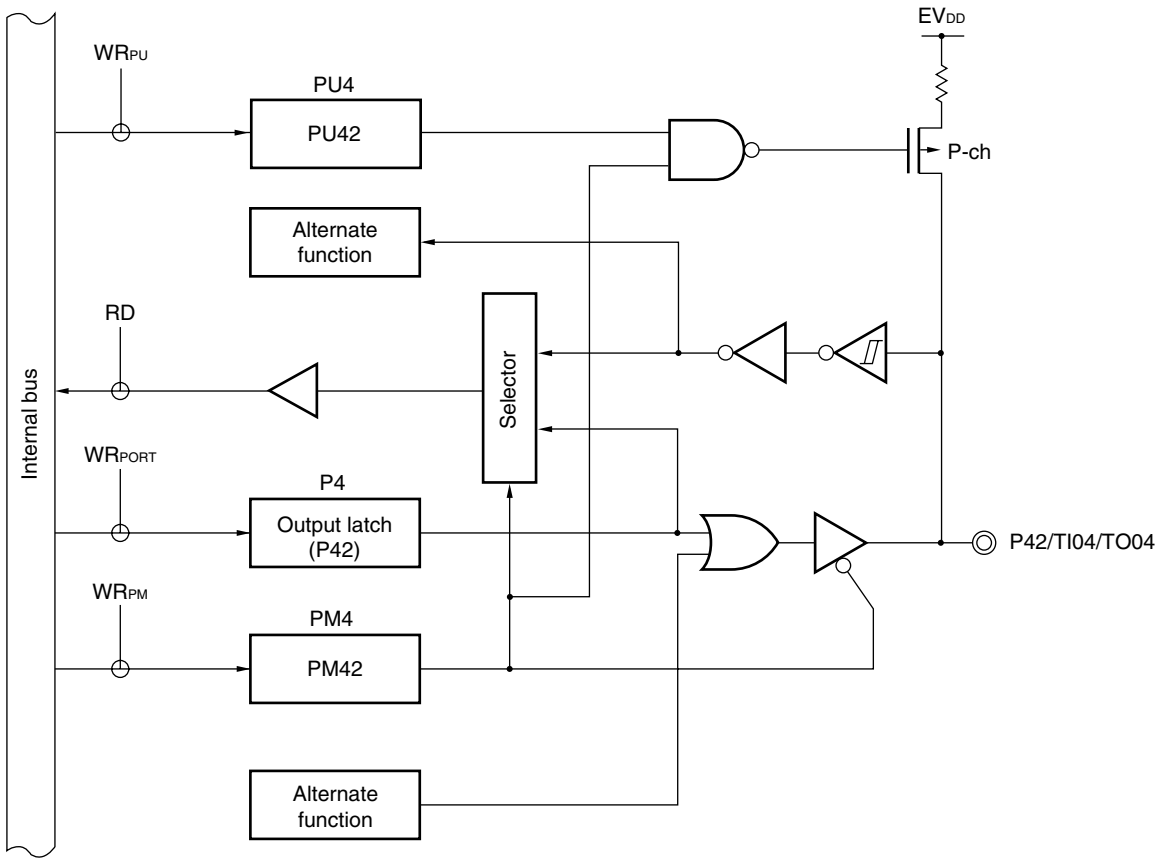
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-13. Block Diagram of P41



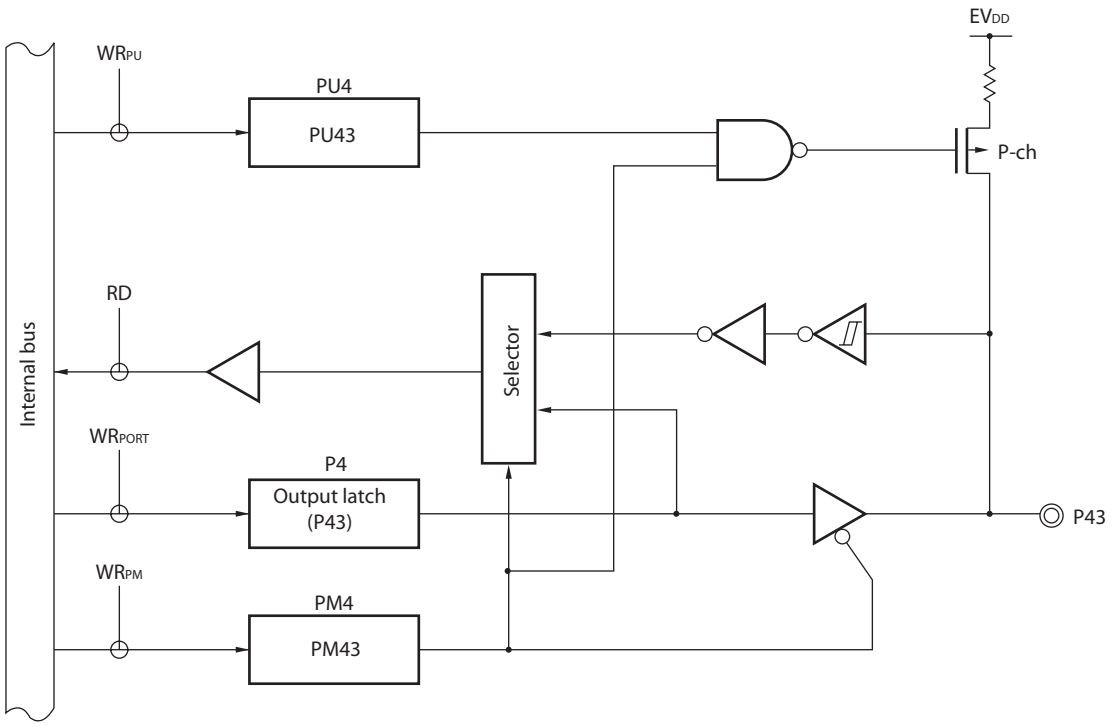
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-14. Block Diagram of P42



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

Figure 4-15. Block Diagram of P43



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

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P50/INTP1	√	√
P51/INTP2	√	√
P52/TO00	–	√
P53/TI00	–	√

It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 - P53 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).

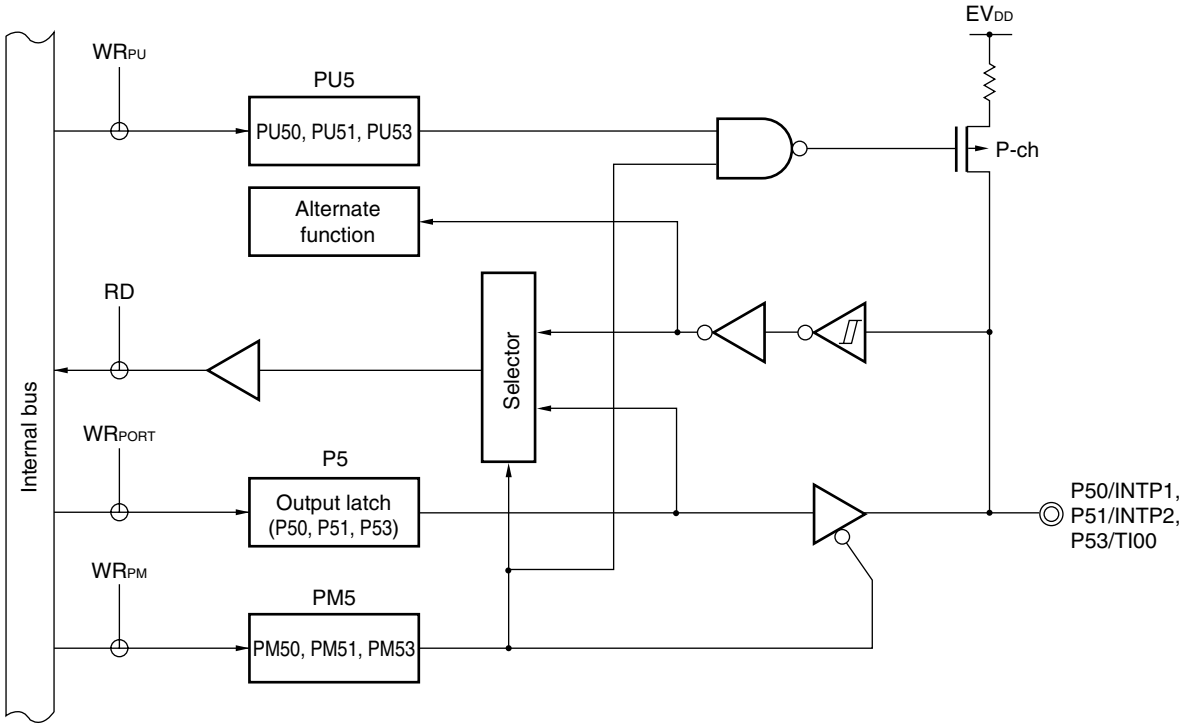
This port can also be used for external interrupt request input and timer I/O.

Reset signal generation sets port 5 to input mode.

Figures 4-16 - 4-17 show block diagrams of port 5.

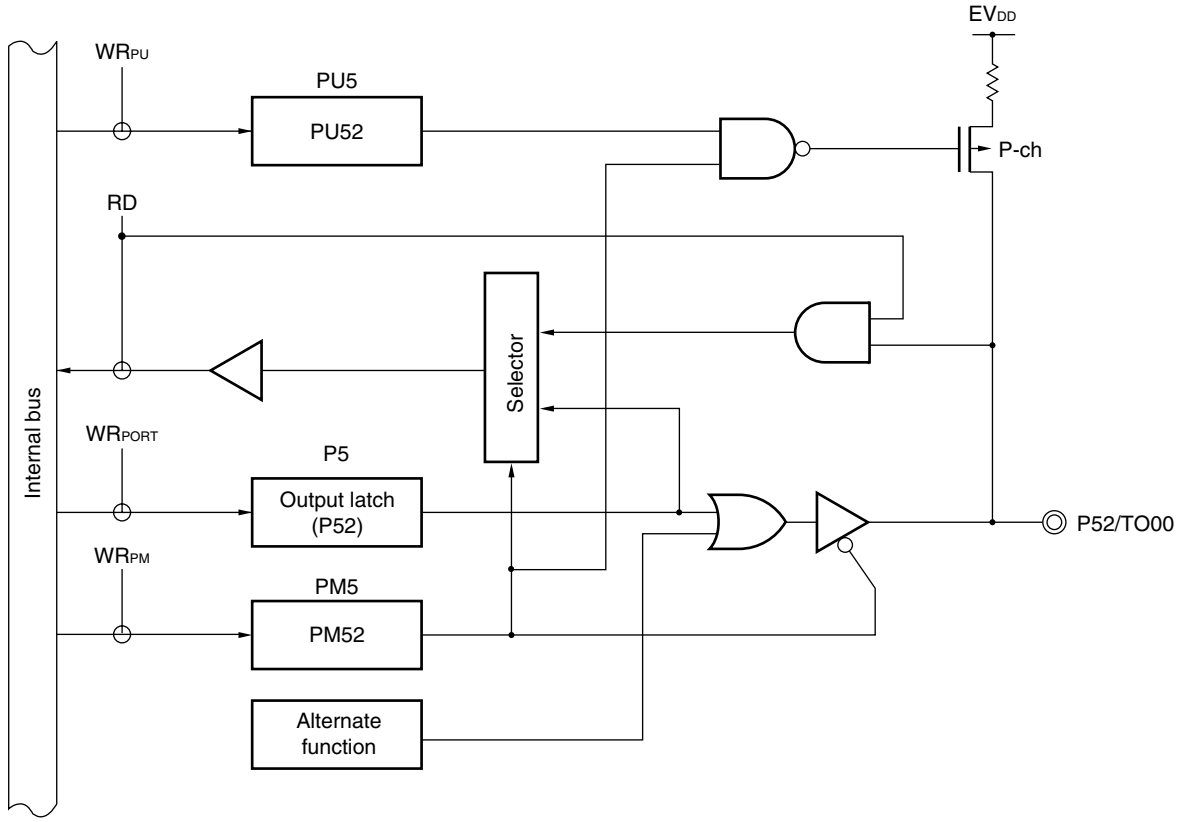
Cautions 1. To use P52/TO00 as a general-purpose port, set bits 0 (TO00) of timer output register 0 (TO0) and bits 0 (TOE00) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 4-16. Block Diagram of P50, P51, P53



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

Figure 4-17. Block Diagram of P52



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

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P60/SCL0	√	√
P61/SDA0	√	√
P62	√	√
P63	√	√

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

The output of the P60 - P63 pins is N-ch open-drain output (6 V tolerance).

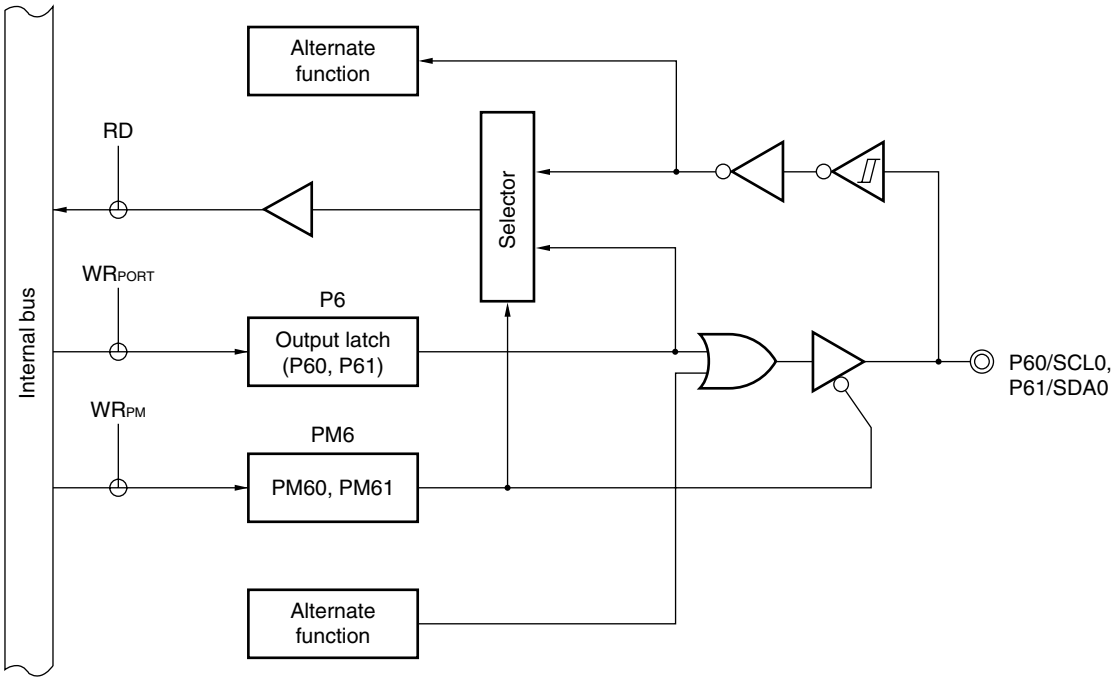
This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

Figures 4-18 - 4-19 show block diagrams of port 6.

Cautions Stop the operation of serial interface IICA when using P60/SCL0 and P61/SDA0 as general-purpose ports.

Figure 4-18. Block Diagram of P60, P61



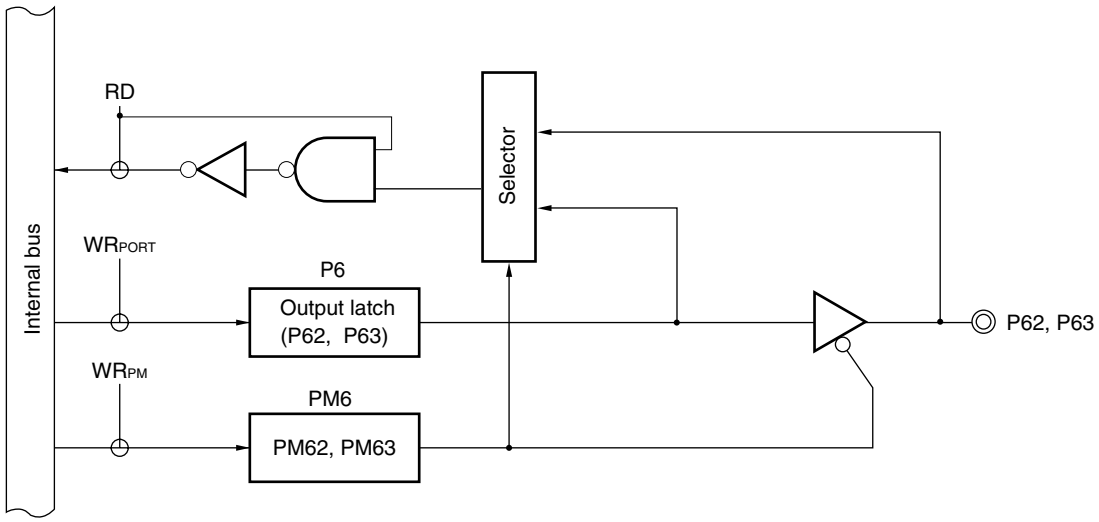
P6: Port register 6

PM6: Port mode register 6

RD: Read signal

WR××: Write signal

Figure 4-19. Block Diagram of P62, P63



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

4.2.8 Port 7

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P70/KR0	√	√
P71/KR1	√	√
P72/KR2	√	√
P73/KR3	√	√
P74/KR4/INTP8	–	√
P75/KR5/INTP9	–	√
P76/KR6/INTP10	–	√
P77/KR7/INTP11	–	√

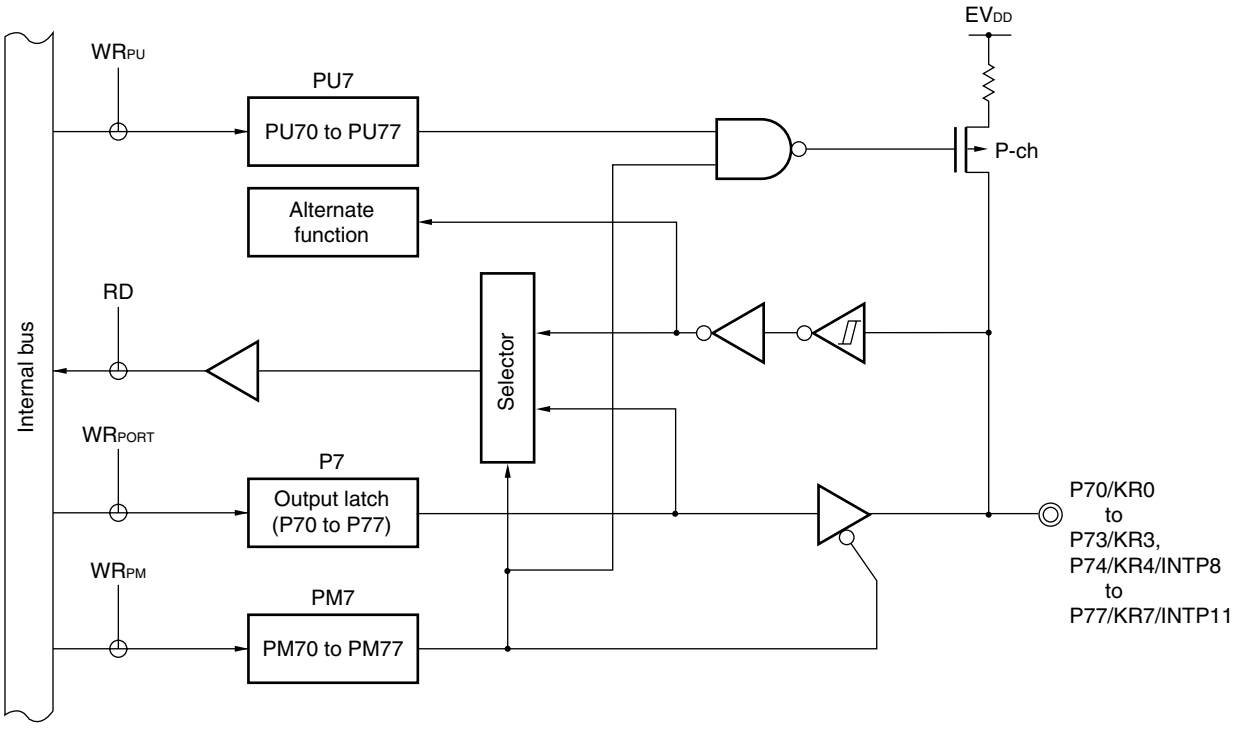
It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

This port can also be used for key return input and interrupt request input.

Reset signal generation sets port 7 to input mode.

Figure 4-19 shows a block diagram of port 7.

Figure 4-20. Block Diagram of P70 - P77



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

4.2.9 Port 11

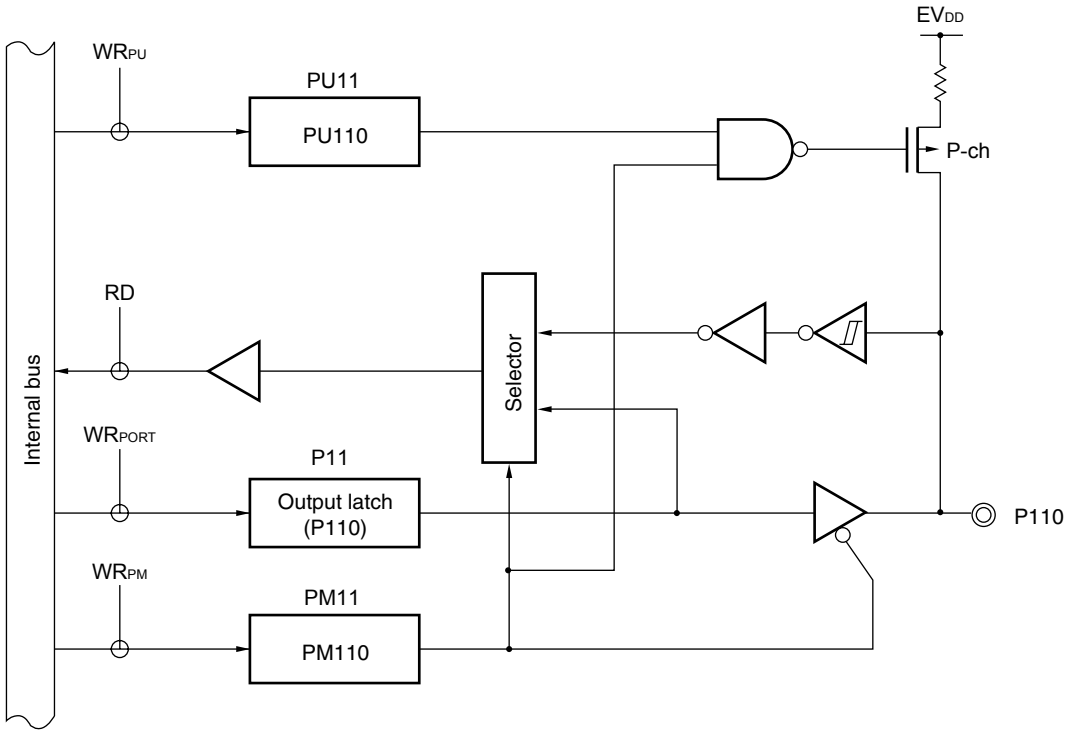
	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μ PD78F102y:y = 5, 6)
P110	–	√
P111	–	√

It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 11 (PM11). When 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 11 (PU11).

Reset signal generation sets to input mode.

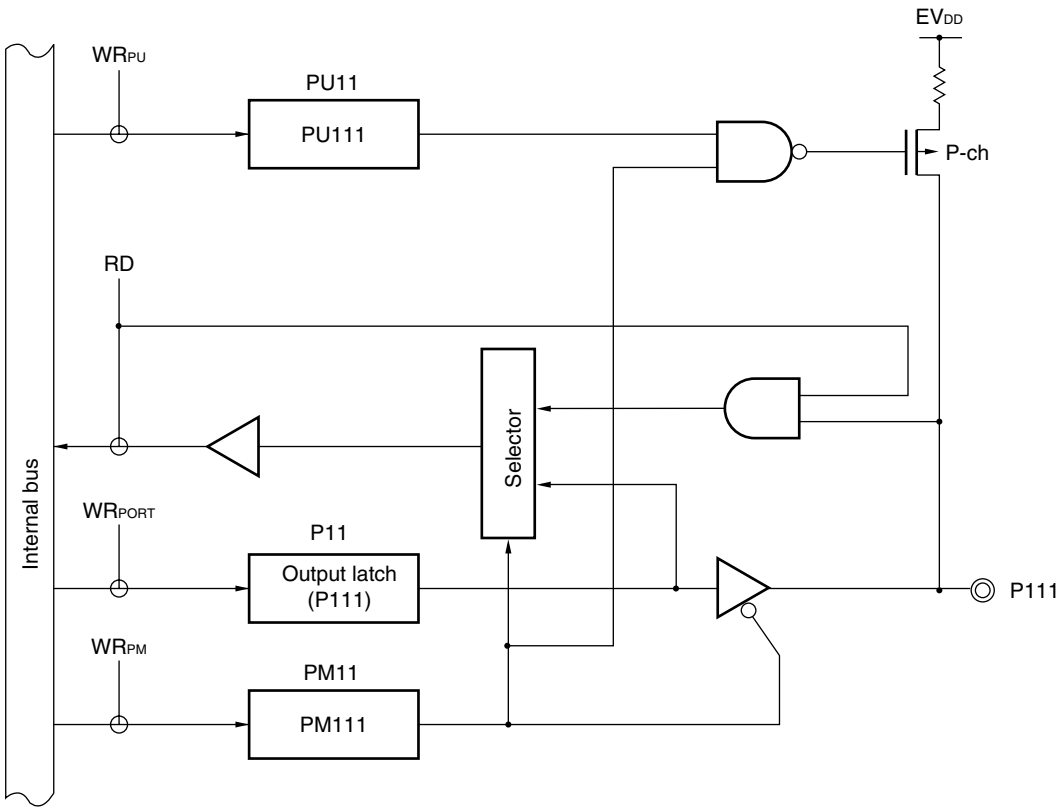
Figure 4-20, 4-21 shows a block diagram of port 11.

Figure 4-21. Block Diagram of P110



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
RD: Read signal
WR××: Write signal

Figure 4-22. Block Diagram of P111



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
RD: Read signal
WR××: Write signal

4.2.10 Port 12

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μ PD78F102y:y = 5, 6)
P120/INTP0/EXLVI	√	√
P121/X1	√	√
P122/X2/EXCLK	√	√
P123/XT1	√	√
P124/XT2	√	√

P120 is an I/O port with an output latch. It 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 - P124 are input ports.

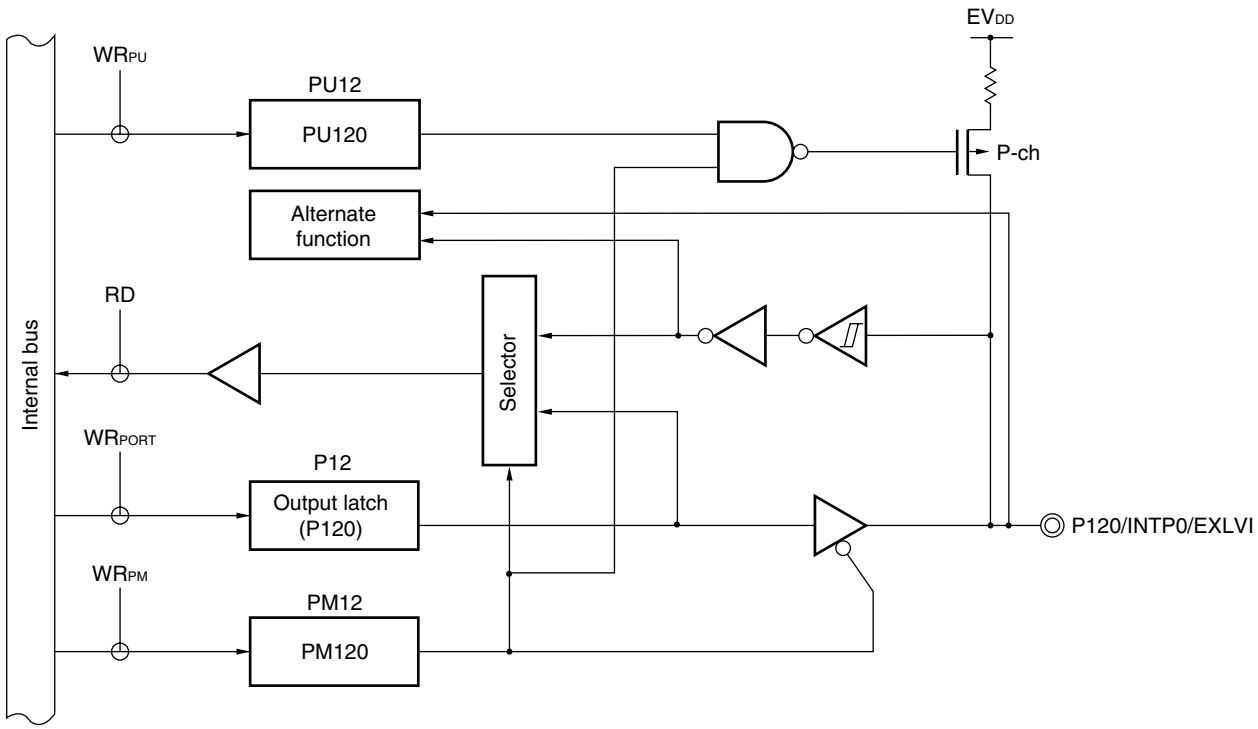
This port can also be used for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input for main system clock.

Reset signal generation sets port 12 to input mode.

Figures 4-22 to 4-24 show block diagrams of port 12.

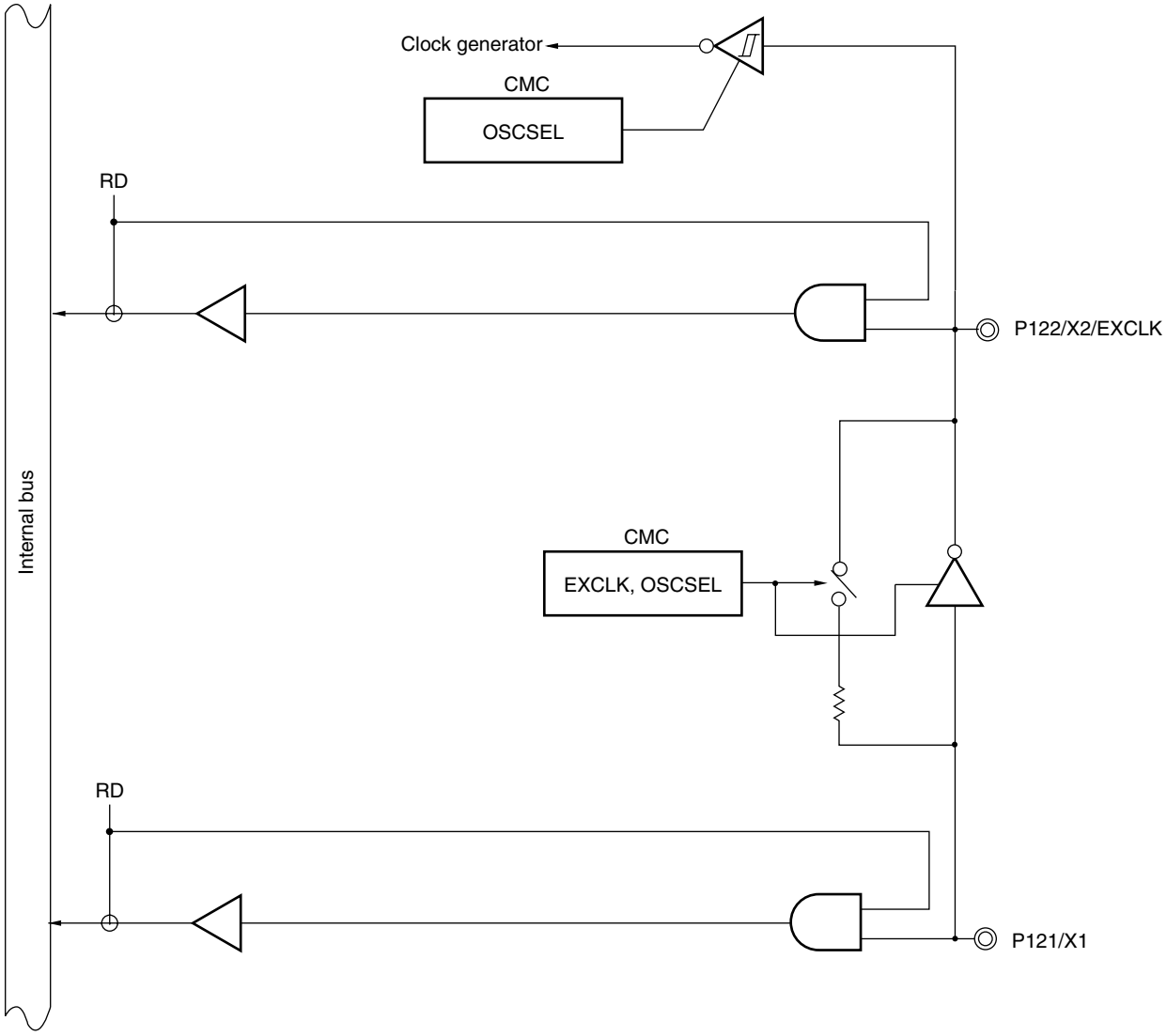
Caution The function setting on P121 - P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

Figure 4-23. Block Diagram of P120



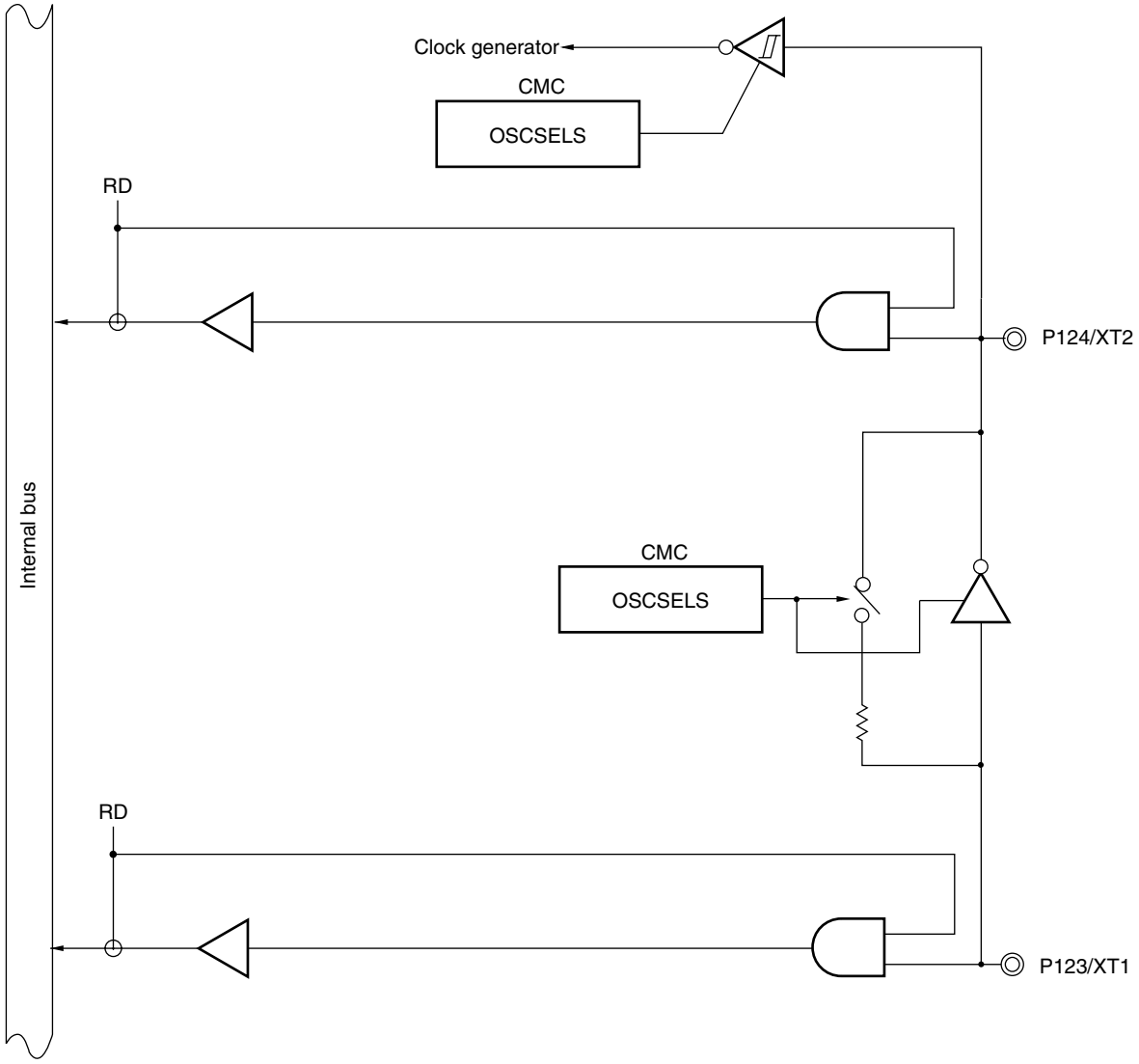
P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

Figure 4-24. Block Diagram of P121, P122



CMC: Clock operation mode control register
RD: Read signal

Figure 4-25. Block Diagram of P123, P124



CMC: Clock operation mode control register
RD: Read signal

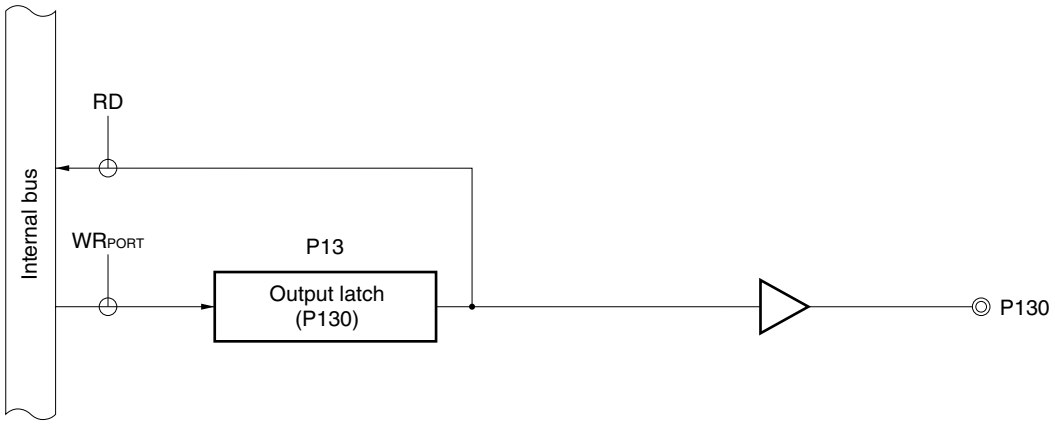
4.2.11 Port 13

	78K0R/KC3-L(48pin) (μPD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P130	–	√

P130 is a 1-bit output-only port with an output latch.

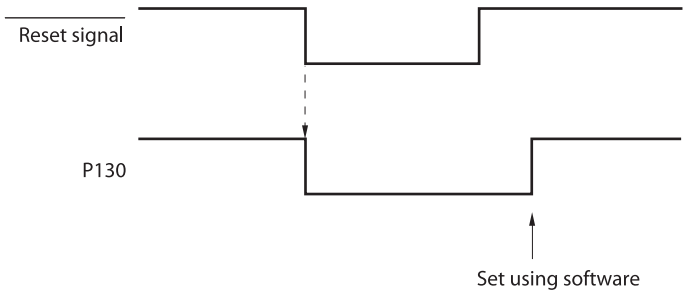
Figure 4-25 shows block diagrams of port 13.

Figure 4-26. Block Diagram of P130



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

Remark When reset is affected, 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.



4.2.12 Port 14

	78K0R/KC3-L(48pin) (μ PD78F102y:y = 2-4)	78K0R/KE3-L(64pin) (μPD78F102y:y = 5, 6)
P140/PCLBUZ0/INTP6	√	√
P142/$\overline{\text{SCK20}}$/SCL20	–	√
P143/SI20/RxD2/SDA20	–	√
P144/SO20/TxD2	–	√

It is an I/O port with an output latch. It can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140, P142 - P144 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).

Input to the P142, P143 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 14 (PIM14).

Output from the P142 - P144 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 14 (POM14).

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

Reset signal generation sets port 14 to input mode.

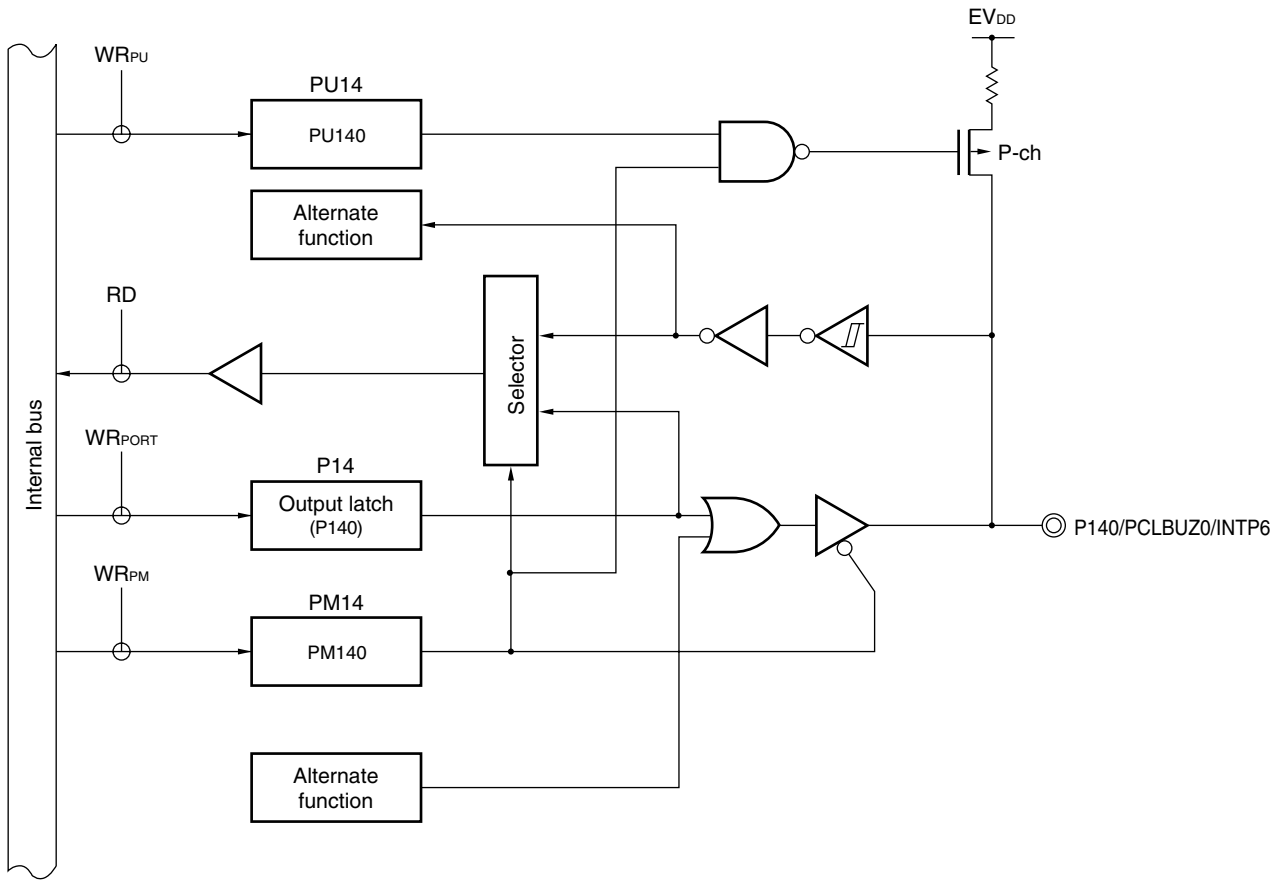
Figures 4-25 - 4-28 show block diagrams of port 14.

Cautions 1. To use P142/$\overline{\text{SCK20}}$/SCL20, P143/SI20/RxD2/SDA20, or P144/SO20/TxD2 as a general-purpose port, note the serial array unit 1 setting. For details, refer to the following tables.

- **Table 11-9 Relationship between Register Settings and Pins (Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)**
- **Table 11-10 Relationship between Register Settings and Pins (Channel 1 of Unit 1: UART2 Reception)**

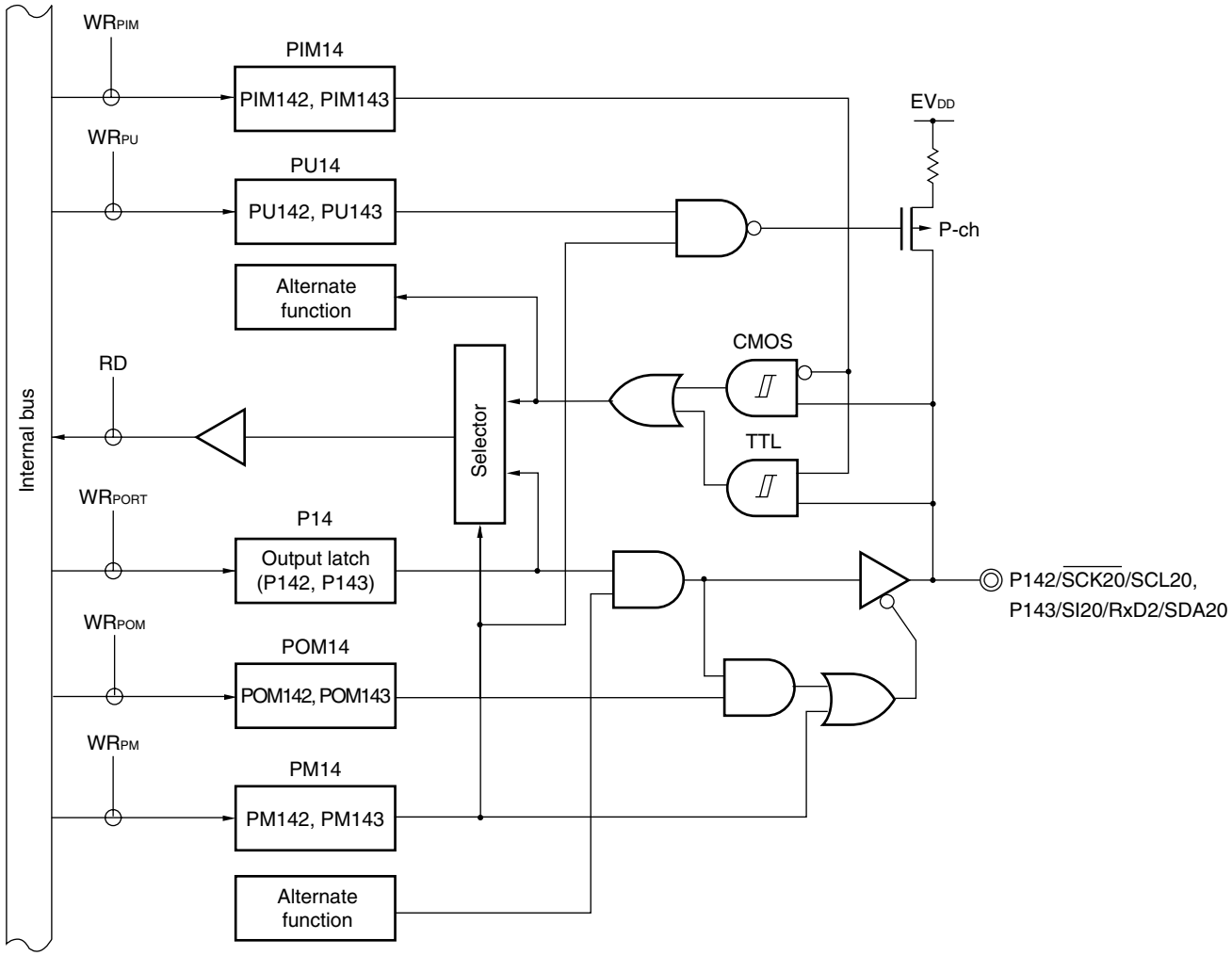
2. To use P140/PCLBUZ0/INTP6 as a general-purpose port, set bit 7 of clock output select register 0 (CKS0) to “0”, which is the same as their default status settings.

Figure 4-27. Block Diagram of P140



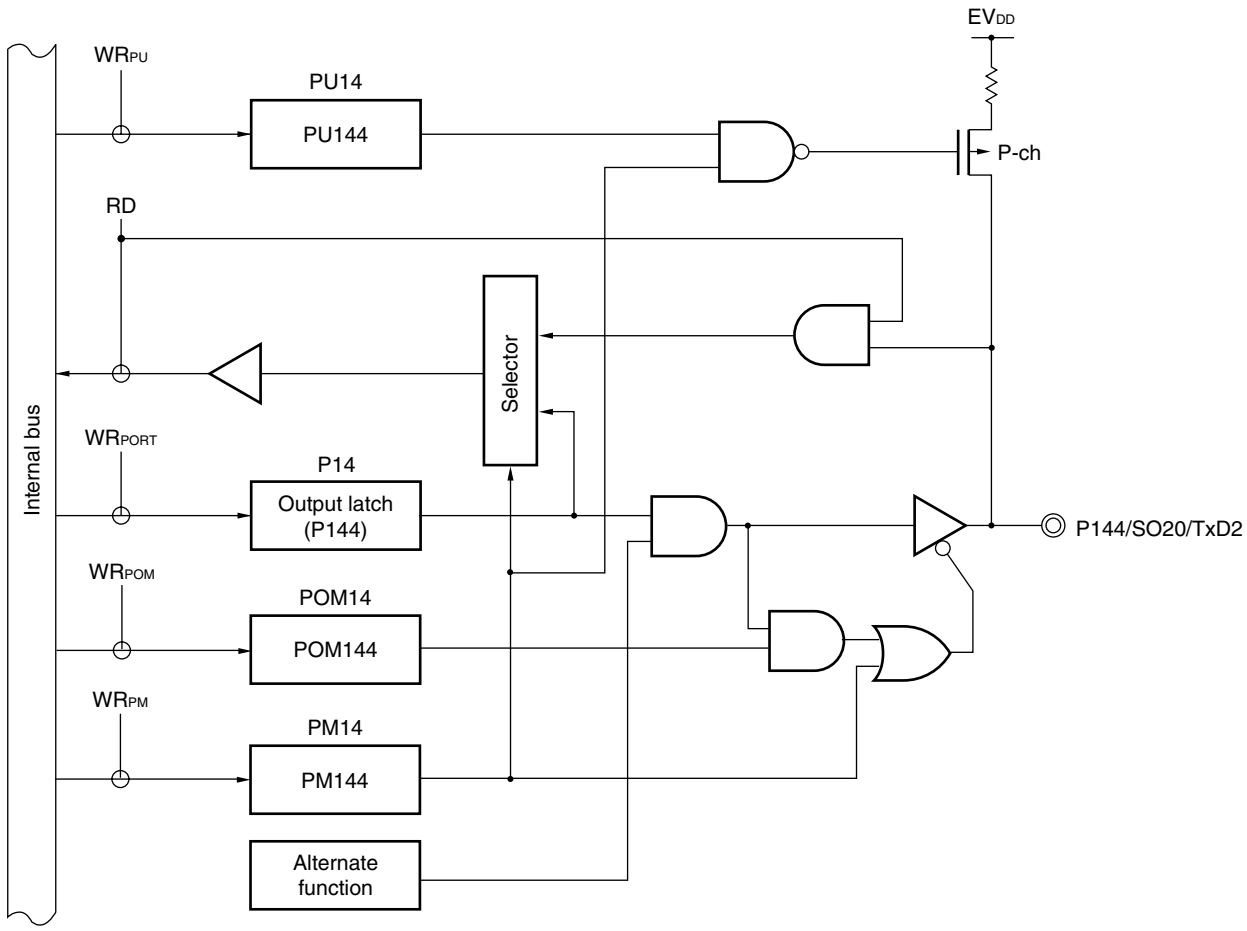
P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
RD: Read signal
WR××: Write signal

Figure 4-28. Block Diagram of P142, P143



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
PIM14: Port input mode register 14
POM14: Port output mode register 14
RD: Read signal
WR××: Write signal

Figure 4-29. Block Diagram of P144



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
POM14: Port output mode register 14
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

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

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- A/D port configuration register (ADPC)

(1) Port mode registers (PMxx)

These registers specify input or output mode for the port in 1-bit units.

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

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Settings of Port Mode Register, and Output Latch When Using Alternate Function**.

Figure 4-30. Format of Port Mode Register (78K0R/KC3-L)

	Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
	PM0	1	0	0	PM04	PM03	PM02	1	1	FFF20H	FFH	R/W
	PM1	PM17	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	1	1	1	PM31	0	FFF23H	FFH	R/W
	PM4	0	0	1	0	0	0	PM41	PM40	FFF24H	FFH	R/W
	PM5	1	1	0	0	0	1	PM51	PM50	FFF25H	FFH	R/W
	PM6	0	0	0	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
	PM7	0	0	0	0	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
<R>	PM11	1	1	1	1	1	1	1	0	FFF2BH	FFH	R/W
	PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
	PM14	1	1	1	1	0	0	1	PM140	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0-7, 12, 14 ; n = 0-7)
0	Output Mode(Output Buffer on)
1	Input mode(Output Buffer off)

Note 1. Be sure to set bits 5, 6 of PM0, bits 0 of PM3, bits 2-4,6,7 of PM4, bits 3, 5 of PM5, bits 5,7 of PM6, bits 4, 7 of
<R> PM7, bits 0 of PM11, bits 2,3 of PM14 to “0”. Please set it to 0 for the second time, when it returns to initial value with reset.

<R> 2. Be sure to set bits 0,1,7 of PM0, bits 2-7 of PM3, bit 5 of PM4, bits 2,6,7 of PM5, bit 4 of PM6, bits 1-7 of PM11, bits 1-7 of PM12, bits 1, 4-7 of PM14 to “1”.

Figure 4-31. Format of Port Mode Register (78K0R/KE3-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	0	0	PM04	PM03	PM02	1	1	FFF20H	FFH	R/W
PM1	PM17	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	1	1	1	PM31	0	FFF23H	FFH	R/W
PM4	0	0	1	0	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	0	0	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	0	0	0	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	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	1	1	1	PM144	PM143	PM142	1	PM140	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0-7, 11, 12, 14 ; n = 0 - 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 1. Be sure to set bits 5, 6 of PM0, bits 0 of PM3, bits 4, 6, 7 of PM4, bits 4, 5 of PM5, bits 5, 7 of PM6. Please set it to 0 for the second time, when it returns to initial value with reset.

2. Be sure to set bits 0,1,7 of PM0, bits 2-7 of PM3, bit 5 of PM4, bits 6,7 of PM5, bit 4 of PM6, bits 2-7 of PM11, bits 1-7 of PM12, 1,5-7 of PM14 to “1”.

(2) Port registers (Pxx)

These registers set the output latch value of a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read[Note].

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

Reset signal generation clears these registers to 00H.

Note It is always 0 and never a pin level that is read out if a port is read during the input mode when P2 are set to function as an analog input for a A/D converter.

Figure 4-32. Format of Port Register (78K0R/KC3-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	P04	P03	P02	0	0	FFF00H	00H (output latch)	R/W
P1	P17	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	0	0	0	P31	0	FFF03H	00H(output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H(output latch)	R/W
P5	0	0	0	0	0	0	P51	P50	FFF05H	00H(output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H(output latch)	R/W
P7	0	0	0	0	P73	P72	P71	P70	FFF07H	00H(output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W[Note]
P14	0	0	0	0	0	0	0	P140	FFF0EH	00H(output latch)	R/W

Pmn	m = 0-7, 12, 14 ; n = 0-7	
	Output data control(After output mode)	Input data reading(After input mode)
0	Output 0	Low level input
1	Output 1	High level Input

Note P121-P124 is Read-only.

Figure 4-33. Format of Port Register (78K0R/KC3-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	P04	P03	P02	0	0	FFF00H	00H(output latch)	R/W
P1	P17	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	0	0	0	P31	0	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	0	0	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	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	P144	P143	P142	0	P140	FFF0EH	00H (output latch)	R/W

Pmn	m = 0 - 7, 11 - 14; n = 0 - 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 - 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 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. 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.

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

Reset signal generation clears these registers to 00H.

Figure 4-34. Format of Pull-up Resistor Option Register (78K0R/KC3-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	PU04	PU03	PU02	0	0	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	0	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	00H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	F0035H	00H	R/W
PU7	0	0	0	0	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	0	0	0	0	PU140	F003EH	00H	R/W

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

Figure 4-35. Format of Pull-up Resistor Option Register (78K0R/KC3-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	PU04	PU03	PU02	0	0	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	0	F0033H	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5	0	0	0	0	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU11	0	0	0	0	0	0	PU111	PU110	F0039H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	PU144	PU143	PU142	0	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 - 7, 11, 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 registers (PIM0, PIM1, PIM14[Note])

PIM0, PIM1, PIM14 are registers that set the input buffer of P03, P04, P10, P11, P142, P143 in 1-bit units.

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

PIM0, PIM1, PIM14 are registers that can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 4-36. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	PIM04	PIM03	0	0	0	F0040H	00H	R/W
PIM1	0	0	0	0	0	0	PIM11	PIM10	F0041H	00H	R/W
PIM14[Note]	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 1, 14; n = 0 - 4)
0	Normal input buffer
1	TTL input buffer

Note : Only 78K0R/KE3-L.

(5) Port output mode registers (POM0, POM1, POM14[Note])

These registers set the output mode of P02 - P04, P10, P12, P142 - P144 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.

POM0, POM1, POM14 are registers that can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 4-37. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	POM04	POM03	POM02	0	0	F0050H	00H	R/W
POM1	0	0	0	0	0	POM12	0	POM10	F0051H	00H	R/W
POM14[Note]	0	0	0	POM144	POM143	POM142	0	0	F005EH	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 14; n = 0, 2 - 4)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Note : Only 78K0R/KE3-L.

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

This register switches the P20/ANI0 - P27/ANI7 pins to digital I/O of port or analog input of A/D converter.

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

Reset signal generation sets this register to 10H.

Figure 4-38. 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 2							
					ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	D	D	D	D	D	D
0	0	1	1	1	A	D	D	D	D	D	D	D
0	1	0	0	0	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D
Other than above					Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode registers 2 (PM2).

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

3. P20/ANI0 - P27/ANI7 are set as analog inputs in the order of P27/ANI7, …, P20/ANI0 by the A/D port configuration register (ADPC). When using analog inputs, start designing from P27/ANI17.

<R> **4. Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.**

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change.
The data of the output latch is cleared when a reset signal is generated.

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

When parts of ports 0, 1, 14[Note] operate with V_{DD} = 1.8 – 3.6 V, I/O connections with an external device that operates on 2.5 V, 3 V power supply voltage are possible.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by port input mode registers (PIM0, PIM1, PIM14[Note]).

Moreover, regarding outputs, different 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 (POM0, POM1, POM14[Note]).

(1) Setting procedure when using I/O pins of UART0 - UART2, CSI00, CSI10, CSI20 functions

(a) Use as 2.5 V, 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 UART0 :P11
In case of UART1 :P03
In case of UART2[Note] :P143
In case of CSI00 :P10, P11
In case of CSI10 :P03, P04
In case of CSI20[Note] :P142, P143

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

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

Note : 78K0R/KE3-L Only

Remark : 78K0R/KC3-L:n= 0, 1
78K0R/KE3-L:n= 0, 1, 14

(b) Use as 2.5 V, 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 : P12
In case of UART1 : P02
In case of UART2 Note: P144
In case of CSI00 : P10, P12
In case of CSI10 : P02, P04
In case of CSI20 Note : P142, P144

<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> Communication is started by setting the serial array unit.

Note : 78K0R/KE3-L Only
Remark : 78K0R/KC3-L:n= 0, 1
78K0R/KE3-L:n= 0, 1, 14

(2) Setting procedure when using I/O pins of simplified IIC10 and 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 : P03, P04
In case of simplified IIC20 Note : P142, P143

<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 corresponding bit of the PMn 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 IIC mode.

Note : 78K0R/KE3-L Only
Remark : 78K0R/KC3-L:n= 0, 1
78K0R/KE3-L:n= 0, 1, 14

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 4-6.

Table 4-6. Settings of Port Mode Register, and Output Latch When Using Alternate Function

Pin Name	Alternate Function		PM××	P××	Pin Name	Alternate Function		PM××	P××
	Function Name	I/O				Function Name	I/O		
P02	SO10	Output	0	1	P40	TOOL0	I/O	×	×
	TxD1	Output	0	1	P41	TOOL1	Output	×	×
P03	SI10	Input	1	×	P42	TI04	Input	1	×
	RxD1	Input	1	×		TO04	Output	0	0
	SDA10	I/O	0	1	P50, P51	INTP1, INTP2	Input	1	×
P04	$\overline{SCK10}$	Input	1	×	P52 note2	TO00	Output	0	0
		Output	0	1	P53 note2	TI00	Input	1	×
	SCL10	I/O	0	1	P60	SCL0	I/O	0	0
P10	$\overline{SCK00}$	Input	1	×	P61	SDA0	I/O	0	0
		Output	0	1	P70 - P73	KR0 - KR3	Input	1	×
P11	SI00	Input	1	×	P74 - P77 note2	INTP8 - INTP11	Input	1	×
	RxD0	Input	1	×		KR4 - KR7	Input	1	×
P12	SO00	Output	0	1	P120	INTP0	Input	1	×
	TxD0	Output	0	1		EXLVI	Input	1	×
P13	TxD3	Output	0	1	P140	PCLBUZ0	Output	0	0
P14	RxD3	Input	1	×		INTP6	Input	1	×
P15	RTCDIV	Output	0	0	P142 note 2	$\overline{SCK20}$	Input	1	×
	RTCCL	Output	0	0			Output	0	1
P16	TI01	Input	1	×		SCL20	I/O	0	1
	TO01	Output	0	0	P143 note 2	SI20	Input	1	×
P17	INTP5	Input	1	×		RxD2	Input	1	×
	TI02	Input	1	×		SDA20	I/O	0	1
	TO02	Output	0	0	P144 note 2	SO20	Output	0	1
P20 - P27 Note1	ANI0 - ANI7 Note1	Input	1	×		TxD2	Output	0	1
P31	TI03	Input	1	×					
	INTP4	Input	1	×					

Note 1. The functions of the ANI0/P20 - ANI7/P27 pins can be selected by using the A/D port configuration register (ADPC), the analog input channel specification register (ADS).
2. 78K0R/KE3-L Only.

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

Table 4-7. Setting Functions of ANI0/P20 - ANI7/P27 Pins

ADPC	PM2	ADS	ANI0/P20 - ANI7/P27, 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	Selects ANI.	Setting prohibited
		Does not select ANI.	

Remark ×: don't care
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 - 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.

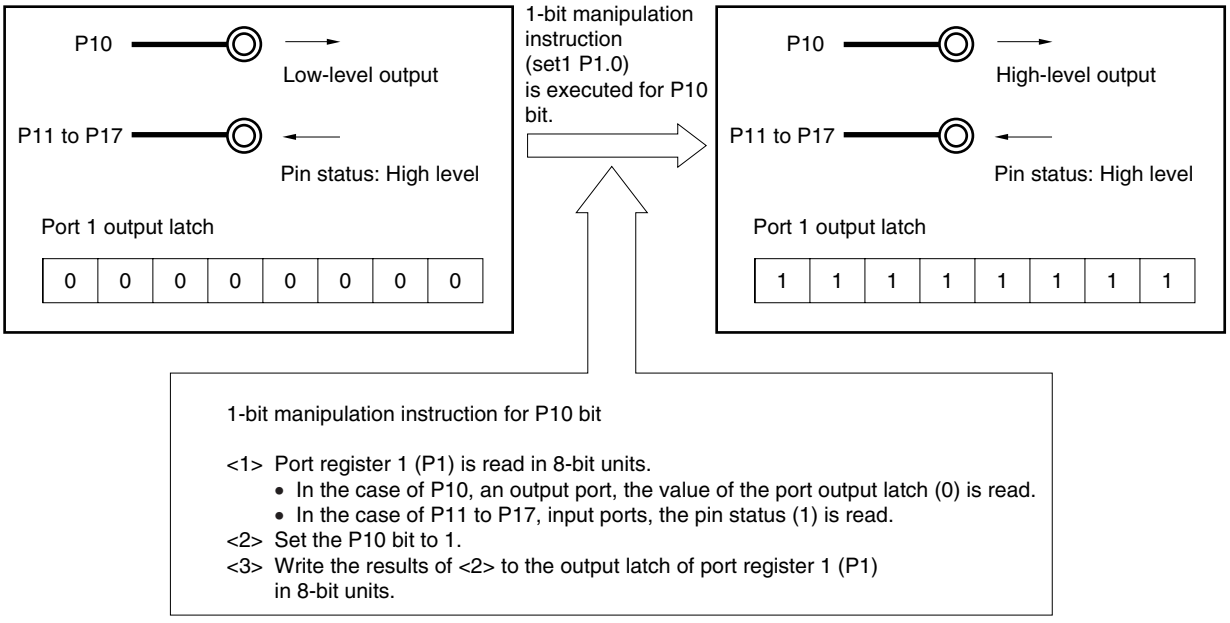
Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the 78K0R/KC3-L, KE3-L.

<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.

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

Figure 4-39. 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, 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 1, 8 or 20 MHz internal high-speed oscillation clock, use the option byte to set the frequency in advance (for details, see **CHAPTER 22 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 dedicated to 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.

(4) USB Clock

- **PLL**

Divide or multiply the 12MHz, 16 MHz, 20 MHz of (fx) clock generated in X1 oscillator or external input clock (fex), and generate 48 MHz. Select PLLM bit division and multiplication ratio of PLL control register (PLLC), and operate or stop PLL based on PLL STOP bit.

Remarks 1. f_{SUB} : Subsystem clock frequency
f_{IL} : Internal low-speed oscillation clock frequency
f_{USB} : USB clock oscillation 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

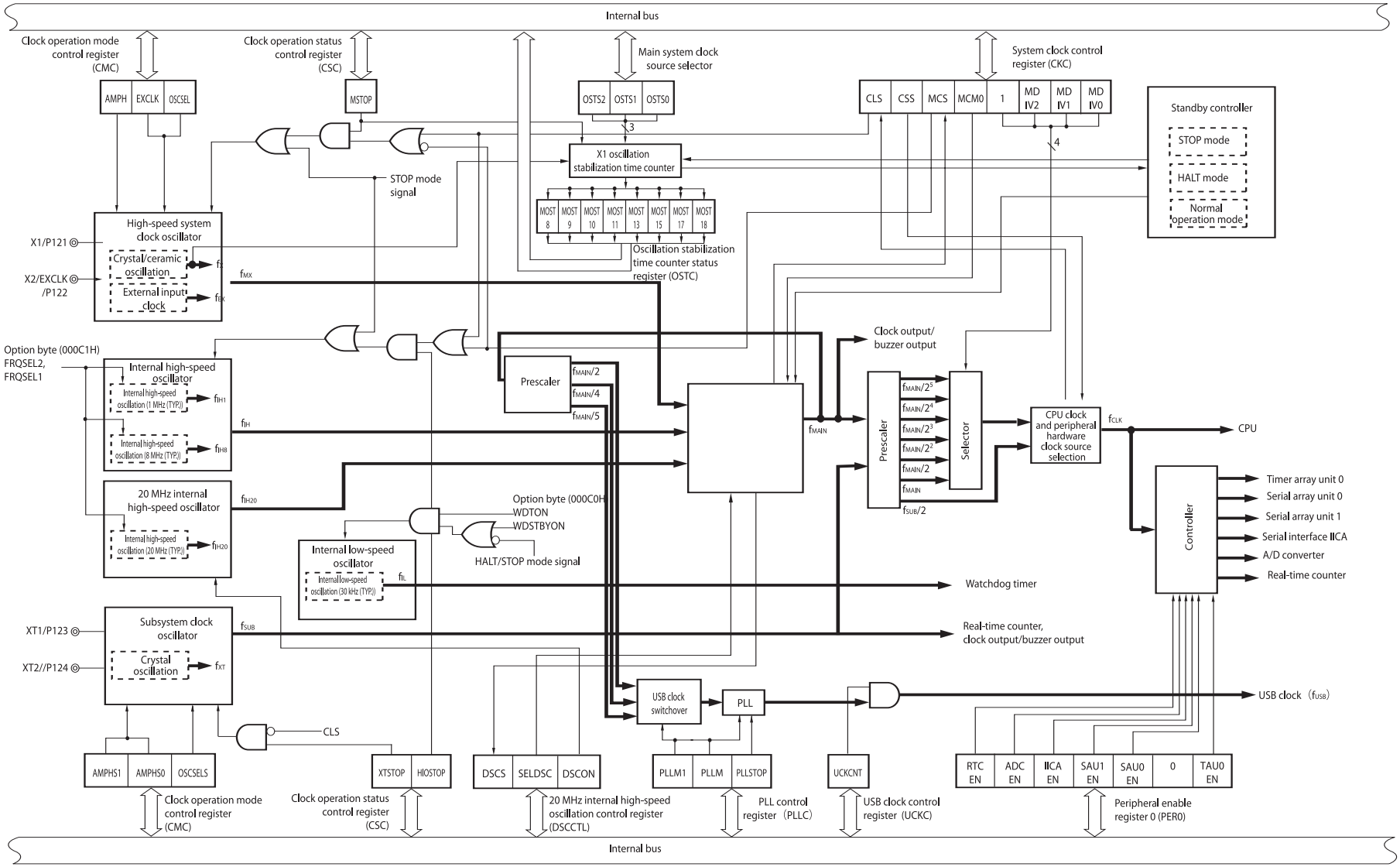
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1 Configuration of Clock Generator

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

Figure 5-1. Block Diagram of Clock Generator



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

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
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Internal low-speed oscillation clock frequency
f_{USB}: USB clock oscillation frequency

5.3 Registers Controlling Clock Generator

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

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

(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 power consumption oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power consumption oscillation
1	1	

AMPH	Control of X1 clock oscillation frequency
0	2 MHz ≤ fx ≤ 10 MHz
1	10 MHz < fx ≤ 20 MHz

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. When CMC is used at the default value (00H), be sure to set 00H to this register after reset release in order to prevent malfunctioning during a program loop.

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

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1 = 1) as the mode of the XT1 oscillator, use the recommended resonators described in CHAPTER 28 ELECTRICAL SPECIFICATIONS.**

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

Remark fx: X1 clock frequency

(2) 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 5-3. 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	1	MDIV2	MDIV1	MDIV0

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

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

MCM0	Main system clock (f_{MAIN}) operation control
0	Selects the internal high-speed oscillation clock (f_{IH}) or 20 MHz internal high-speed oscillation clock (f_{IH20}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

CSS	MDIV2	MDIV1	MDIV0	Selection of CPU/peripheral hardware clock (f_{CLK})
0	0	0	0	f_{MAIN}
	0	0	1	$f_{MAIN}/2$ (This is the default setting if MCM0 = 0.)
	0	1	0	$f_{MAIN}/2^2$
	0	1	1	$f_{MAIN}/2^3$ Note 2
	1	0	0	$f_{MAIN}/2^4$ Note 2
	1	0	1	$f_{MAIN}/2^5$ Notes 2, 3
1 Note 4	×	×	×	$f_{SUB}/2$
Other than above				Setting prohibited

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

<R> **2.** Setting is prohibited if the 1 MHz Internal high-speed oscillation clock frequency (f_{IH1}) is selected as the main system clock (f_{MAIN}).

3. Setting is prohibited if the high-speed system clock (f_{MX}) is selected as the main system clock (f_{MAIN}) and if $f_{MX} < 4$ MHz.

<R> **4.** Changing the value of the MCM0 bit is prohibited while the CSS bit is set to 1.

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** are listed on the next page.)

Cautions 1. Be sure to set bit 3 to 1.

2. The clock set by CSS, MCM0, 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, clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

3. 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 28 ELECTRICAL SPECIFICATIONS.

4. Do not switchover to subsystem clock at the time of USB operation. In case switchover is necessary, set UCKC=0 beforehand, and stop supply to USB.

The fastest instruction can be executed in 1 clock of the CPU clock in the **78K0R/KC3-L, 78K0R/KE3-L**. Therefore, the relationship between the CPU clock (f_{CLK}) and the minimum instruction execution time is as shown in Table 5-2.

Table 5-2 Relationship between CPU Clock and Minimum Instruction Execution Time

CPU Clock (Value set by the 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 12 MHz Operation	At 16 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 20 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{MAIN}	0.0833 μs	0.0625 μs	0.05 μs	0.125 μs (TYP.)	0.05 μs	–
$f_{MAIN}/2$	0.167 μs	0.125 μs	0.1 μs	0.25 μs (TYP.) (default)	0.1 μs	–
$f_{MAIN}/2^2$	0.333 μs	0.25 μs	0.2 μs	0.5 μs (TYP.)	0.2 μs	–
$f_{MAIN}/2^3$	0.666 μs	0.5 μs	0.4 μs	1.0 μs (TYP.)	0.4 μs	–
$f_{MAIN}/2^4$	1.33 μs	1 μs	0.8 μs	2.0 μs (TYP.)	0.8 μs	–
$f_{MAIN}/2^5$	2.67 μs	2 μs	1.6 μs	4.0 μs (TYP.)	1.6 μs	–
$f_{SUB}/2$	–			–		61 μs

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

(3) 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 5-4. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W								
Symbol	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP

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

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

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

<R> **Note** The 1 MHz or 8 MHz (TYP.) internal high-speed oscillation clock stops. Stopping the internal high-speed oscillator (HIOSTOP = 1) is prohibited while the 20 MHz internal high-speed oscillation clock is operating (DSCON = 1). Stop the 20 MHz internal high-speed oscillation clock by using the 20 MHz internal high-speed oscillation control register (DSCCTL) and not the HIOSTOP bit.

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

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 CSC 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 5-3 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 and peripheral hardware clocks operate with a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
Subsystem clock	CPU and peripheral hardware clocks operate 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

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

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

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

- If the X1 clock starts oscillation, while the 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.

Reset signal generation, the STOP instruction, or 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 5-5. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R									
Symbol	7	6	5	4	3	2	1	0	
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	

MOST8	MOST9	MOST10	MOST11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status			
									fx = 12 MHz	fx = 16 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	21.33 μs max.	16 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	21.33 μs min	16 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	42.67 μs min.	32 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	85.33 μs min.	64 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	170.67 μs min.	128 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	682.67 μs min.	512 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	2.73 ms min.	2.05 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	10.92 ms min.	8.19 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	21.85 ms min.	16.38 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.

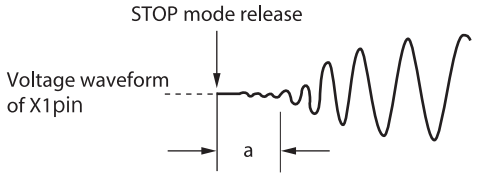
2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.

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 fx: X1 clock oscillation frequency

(5) 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. Using OSTC, the oscillation stabilization time up to the time set to OSTS can be checked.

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

Reset signal generation sets OSTS to 07H.

Figure 5-6 Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W								
Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time status			
				f_X = 12 MHz	f_X = 16 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	21.33 μs	Setting prohibited	Setting prohibited
0	0	1	$2^9/f_X$	42.67μs	32 μs	25.6 μs
0	1	0	$2^{10}/f_X$	85.33 μs	64 μs	51.2 μs
0	1	1	$2^{11}/f_X$	170.67 μs	128 μs	102.4 μs
1	0	0	$2^{13}/f_X$	682.67μs	512 μs	409.6 μs
1	0	1	$2^{15}/f_X$	2.73ms	2.05 ms	1.64 ms
1	1	0	$2^{17}/f_X$	10.92 ms.	8.19 ms	6.55 ms
1	1	1	$2^{18}/f_X$	21.85 ms	16.38 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.

<R> **3. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.**

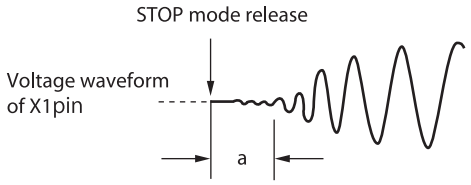
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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

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

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

This register can be used to control oscillation of the 20 MHz internal high-speed oscillation clock (f_{IH20}) and select the 20 MHz internal high-speed oscillation clock (f_{IH20}) as the CPU/peripheral hardware clock.

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 (The CPU/peripheral hardware clock (f_{CLK}) operates on the 20 MHz internal high-speed oscillation clock.)

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	Operating or stopping 20 MHz internal high-speed oscillation clock (f_{IH20})
0	Stopped
1	Operated

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.

<R> **4. If 1 MHz internal oscillation is selected by using the option byte, 20 MHz internal high-speed oscillation cannot be used. Do not set (1) the DSCON bit.**

(7) Peripheral Enable register 0 (PER0)

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

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

- Real-time counter
- A/D converter
- Serial interface IICA
- Serial array unit 0
- Serial array unit 1
- Timer array unit

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register 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	0	ADCEN	IICAEN	SAU1EN	SAU0EN	0	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock supply 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	Enables input clock supply. • SFR used by the real-time counter (RTC) can be read and written.

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

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

Note The input clock that can be controlled by RTCEN is used when the register that is used by the real-time counter (RTC) is accessed from the CPU. RTCEN cannot control supply of the operating clock (f_{SUB}) to RTC.

Caution Be sure to clear bit 1 and 6 to 0.

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

Address: F00F0H After reset: 00H R/W								
Symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	0	TAU0EN

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

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

TAU0EN	Control of timer array unit 0 input clock supply
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	Enables input clock supply. • SFR used by timer array unit 0 can be read and written.

Caution Be sure to clear bit 1 and 6 to 0.

<R> **(8) Operation speed mode control register (OSMC)**

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

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

If the microcontroller operates on a system clock of 10 MHz or more, set this register to 01B.

If the microcontroller operates at low speed on a system clock of 10 MHz or less, power consumption can be reduced, because the voltage booster can be stopped by setting this register to its initial value, 00B. Furthermore, when CPU operates with the system clock of 1 MHz, the power consumption can be further reduced by setting the FLPC bit to 1.

If the RTCLPC bit is set to 1 and real-time counter is operating, current consumption can be reduced, because the circuit that synchronizes the clock to the peripheral functions, except the real-time counter, is stopped in STOP mode and in HALT mode while subsystem clock is selected as CPU clock.

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

Reset signal generation clears this register to 00H.

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

Address: F00F3H After reset: 00H R/W

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

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

RTCLPC	Setting in STOP mode and in HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See Table 17-1 for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time counter

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

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 the FSEL bit to "1", then change f_{CLK} after three or more clocks have elapsed.

4. To set the FSEL bit to 0, set f_{CLK} to 10 MHz or less in advance.

5. Set FSEL = 0 to shift to STOP mode while $V_{DD} \leq 2.7$ V.

6. The HALT mode current when STOP mode and when the subsystem clock is used can be reduced by setting the RTCLPC bit to 1. However, no clock can be supplied to the peripheral functions other than the real-time counter during HALT mode while subsystem clock is selected as CPU clock. Set bit 7 (RTCEN) of peripheral enable registers 0 (PER0), to 1, and all of bits 0 to 6 of the PER0 register to 0 before setting subsystem clock HALT mode.

7. If the FLPC bit is set to a frequency of 1 MHz or less and then set (1), it cannot be cleared (0) or set to a frequency of more than 1 MHz.

(9) PLL **control register (**PLLC**)**

This register is used to set PLL operation mode.

Set PLLC using 1 bit or 8 bit memory manipulation instruction.

It turns to 1 according to the generated reset signal.

Figure 5-10. Format of PLL Control Register (PLLC)

Address:F059CH After reset:01H R/W								
Symbol	7	6	5	4	3	2	1	0
PLLC	0	0	0	0	0	PLLM1	PLLM	PLLSTOP

PLLM1	PLLM	Supply clock to PLL/PLL multiplier selection	
		Supply clock	Multiplier selection
0	0	f_{MX}/2	8multiplier[note1]
0	1	f_{MX}/4	12multiplier[note2]
1	0	f_{MX}/5	12multiplier[note3]
1	1	One set prohibition	

PLLSTOP	PLL operation control
0	Operate PLL
1	Stop PLL

Note1. When f_{MX} = 12 MHz, f_{USB} = 48 MHz.

2. When f_{MX} = 16 MHz, f_{USB} = 48 MHz.

3. When f_{MX} = 20 MHz, f_{USB} = 48 MHz.

Cautions 1. When using USB, set PLL supply clock without fail, at the time of initial settings after reset .

<Setting order>

<1> Stop PLL (PLLSTOP = 1)

<2> Select PLLM1, PLLM (PLLM1 = 0, PLLM = 0:When f_{MX} is 12 MHz, PLLM1 = 0, PLLM = 1: When f_{MX} is16 MHz, PLLM1 = 1, PLLM0 = 0 : When f_{MX} is20 MHz)

<3> Enable PLL operation (PLLSTOP = 0)

2. Carry out PLL oscillation start operation (PLLSTOP = 0) in USB clock (f_{USB}) supply stop position (UCKCNT = 0)

3. Change in settings of multiplication rate of PLL operation(PLLM, PLLM1 bit change) is prohibited

4. At the time of PLL stop (PLLSTOP = 1) clock supply to USB function controller (UCKCNT = 1) is prohibited.

5. At the time of clock supply to USB function controller, PLL cannot be stopped (PLLSTOP = 1).

6. Multiplication rate settings cannot be changed (PLLM, PLLM1bit change) at the time of clock supply to USB function controller.

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

(10) USB Clock control register (UCKC)

This is a register which controls the USB clock (f_{USB})supplying to the USB function controller.

Set **UCKC** using 1 bit or 8 bit memory manipulation instruction

It turns to 00H according to the generated reset signal.

Figure 5-11 Format of USB clock control register (UCKC)

Address:F059DH After Reset:00H R/W								
Symbol	7	6	5	4	3	2	1	0
UCKC	UCKCNT	0	0	0	0	0	0	0

UCKCNT	USB clock supply control to USB function controller
0	Stop USB clock supply
1	USB clock supply

Caution: When shifting to the STOP mode, stop clock supply to USB function controller.

At the time of STOP mode, release Count PLL oscillation stabilization wait time (800 μs) using the software and supply clock to USB function controller when oscillation stabilization wait time ends.

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

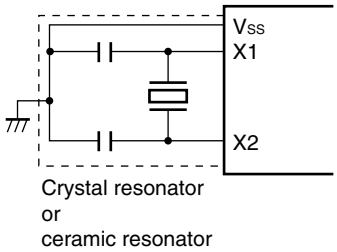
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

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

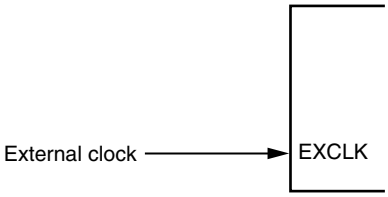
Figure 5-12 shows an example of the external circuit of the X1 oscillator.

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

(a) Crystal or ceramic oscillation



(b) External clock



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.

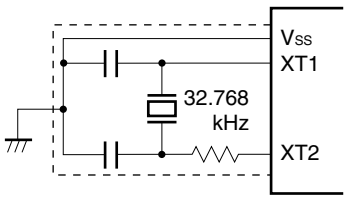
To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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

Figure 5-13. 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 5-12 and 5-13 to avoid an adverse effect from wiring capacitance.

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

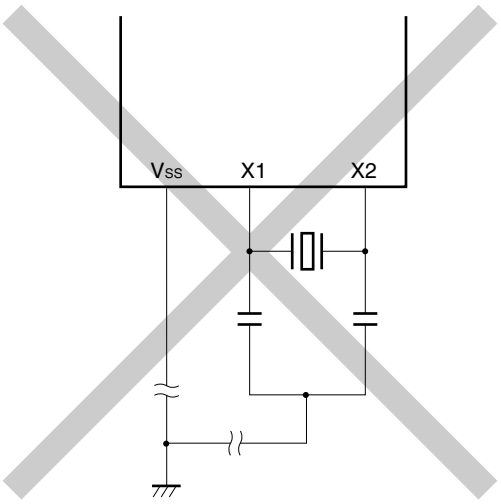
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1 = 1) as the mode of the XT1 oscillator, use the recommended resonators described in CHAPTER 28 ELECTRICAL SPECIFICATIONS.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1 = 1) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

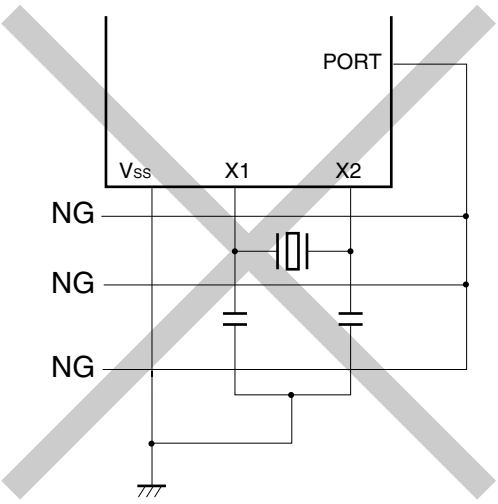
Figure 5-14 shows examples of incorrect resonator connection.

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

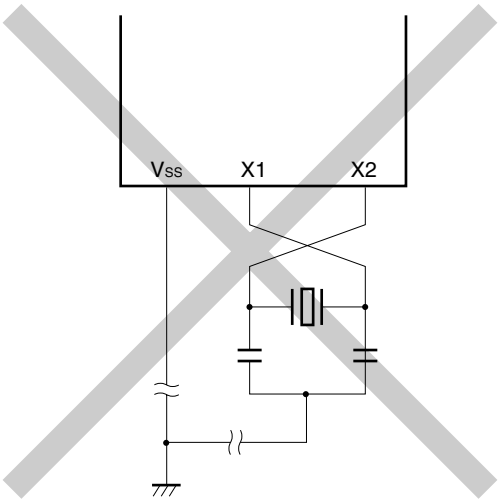
(a) Too long wiring



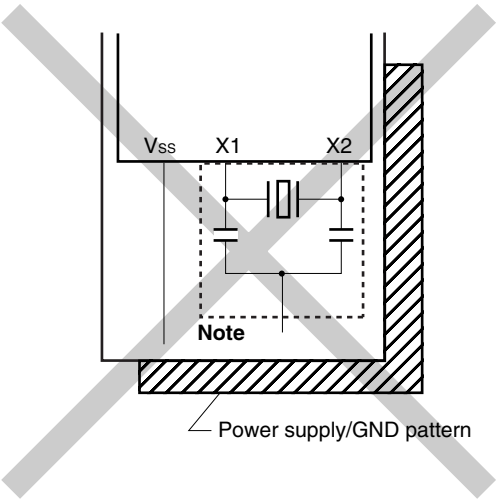
(b) Crossed signal line



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



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

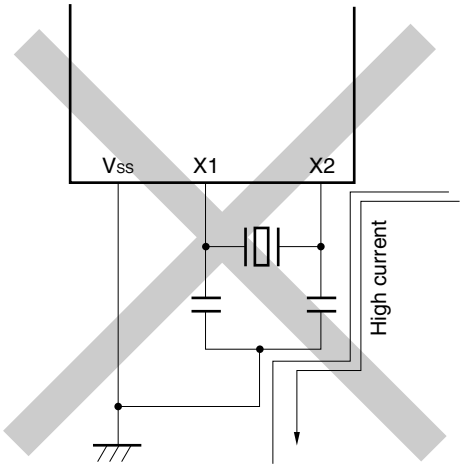


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

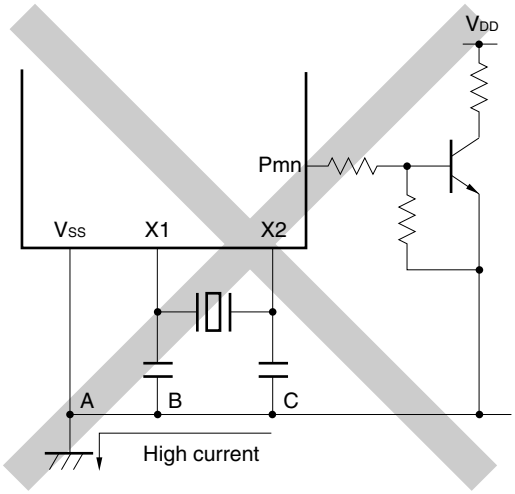
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

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

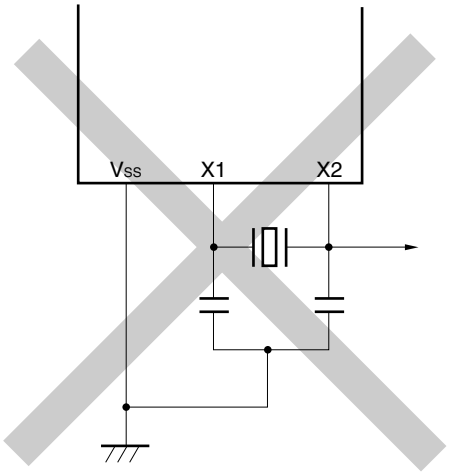
(e) Wiring near high alternating current



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



(g) Signals are fetched



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

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

5.4.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the **78K0R/KC3-L, 78K0R/KE3-L** (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 22 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 20 MHz internal high-speed oscillation control register (DSCCTL) 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/KC3-L, 78K0R/KE3-L.**

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.

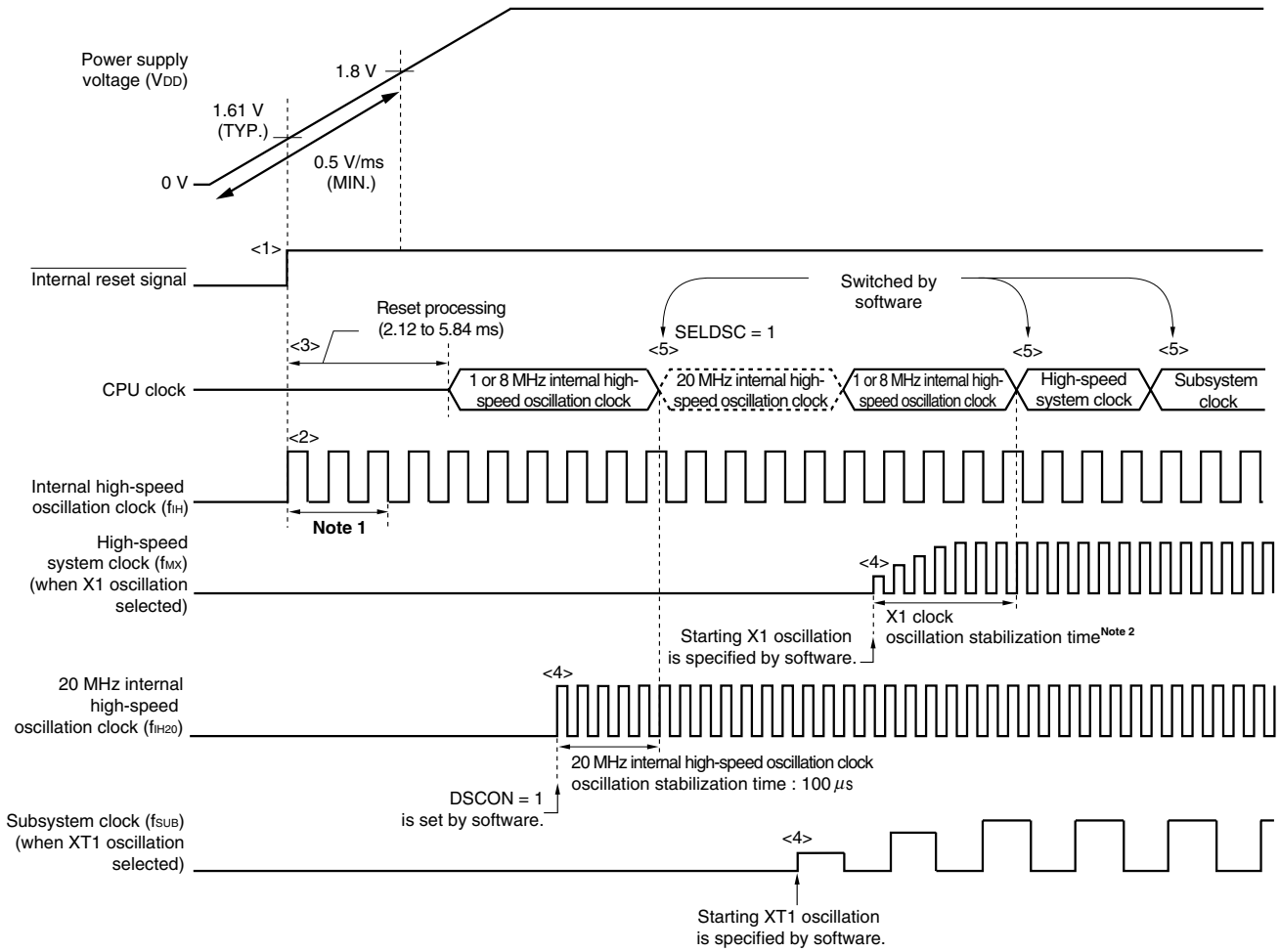
5.5 Clock Generator Operation

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}
  - 20 MHz internal high-speed oscillation clock f_{IH20}
- Subsystem clock f_{SUB}
- Internal low-speed oscillation clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}
- USB clock f_{USB}

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the**78K0R/KC3-L, 78K0R/KE3-L**.When the power supply voltage is turned on; the clock generator operation is shown in Figure 5-15 and Figure 5-16.

<R> **Figure 5-15. 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 Note 3 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**).

Switch to oscillation using the 20 MHz internal high-speed oscillation clock after confirming that the power supply voltage is at least 2.7 V and 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 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).

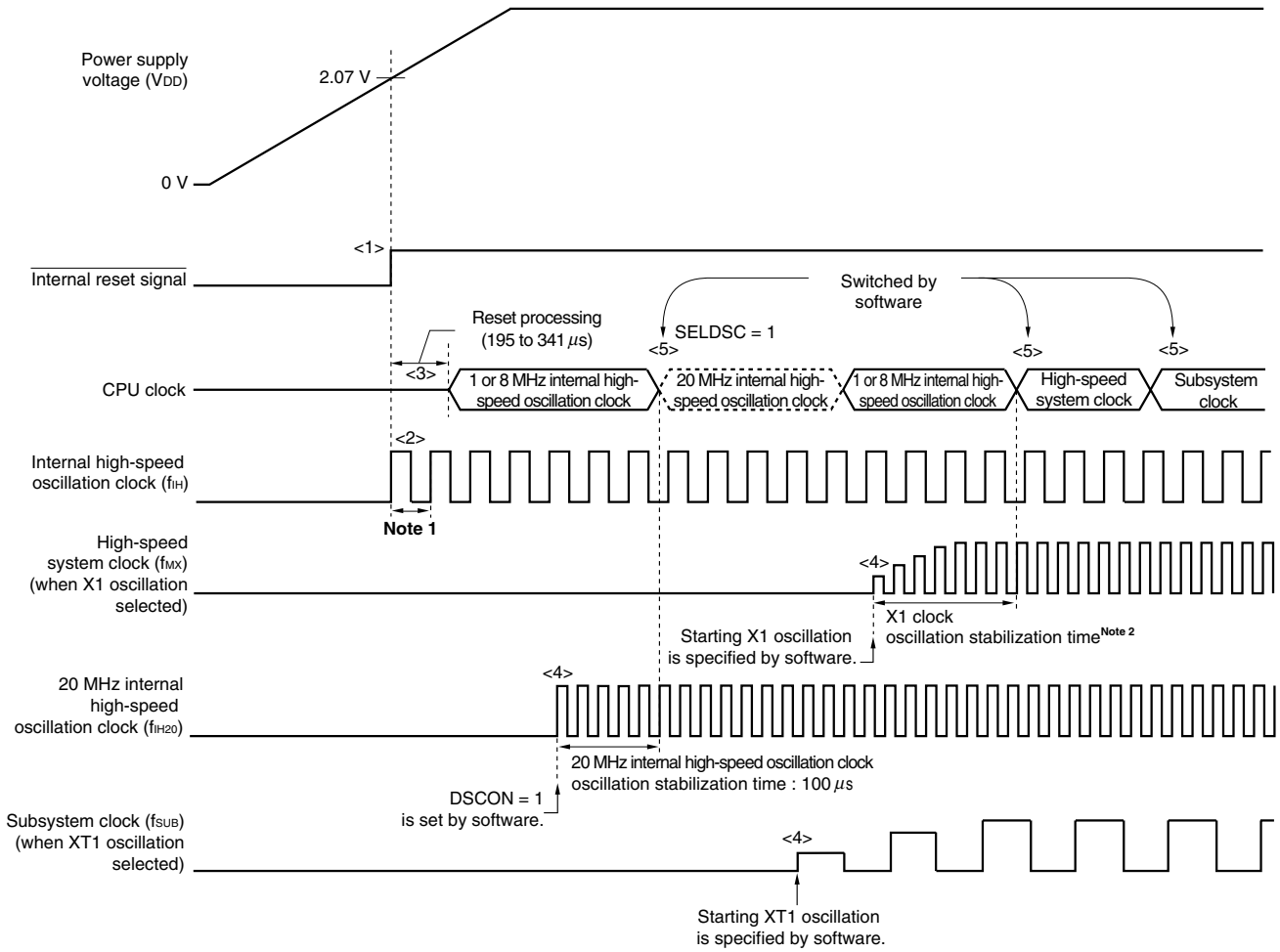
3. The microcontroller operates on the 8 MHz internal high-speed oscillation clock if 8 MHz or 20 MHz is selected for the internal high-speed oscillator by using the option byte or on the 1 MHz internal high-speed oscillation clock if 1 MHz is selected.

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 5-16). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-15 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 5-16. 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 28 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 28 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	0	ADCEN	IICAEN	SAU1EN	SAU0EN	0	TAU0EN

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

Caution Be sure to clear bit 6 to 0.

Remark RTCEN: Control of the real-time counter input clock
ADCEN: Control of the A/D converter input clock
IICAEN: Control of the serial interface IICA input clock
SAU1EN: Control of the serial array unit 1 input clock
SAU0EN: Control of the serial array unit 0 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 17 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	$f_{IH}/2^4$
	1	0	1	$f_{IH}/2^5$

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 17 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, 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 28 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 clock (CKC register)

CSS	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
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, 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 28 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).

5. 6. 5 USB Clock control

The USB function controller clock (f_{USB} = 48 MHz) uses the division clock of the main system clock frequency (f_{MAIN}) by multiplying it by PLL.

- When USB clock is provided from f_{MX} = 12/16/20 MHz (Setting method example)

<1> Set PLLSTOP to 1 (PLLC Register)

Set PLLSTOP to 1, and stop PLL operation.

<2> Set PLLM to 0/1 (PLLC Register)

When f_{MAIN} = 12 MHz, set PLLM to 0, and select 8 multiplier.

When f_{MAIN} = 16 MHz, set PLLM to 1, and select 12 multiplier.

When f_{MAIN} = 20 MHz, set PLLM1 to 1, and select 12 multiplier.

<3> Set PLLSTOP to 0 (PLLC Register)

PLL operation starts when PLLSTOP is set to 0.

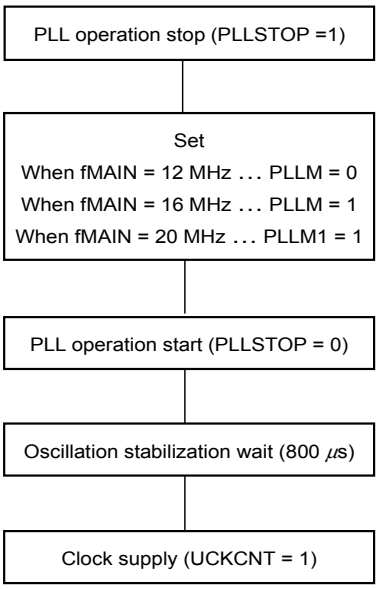
<4> PLL Oscillation stabilization wait

800 μs wait using software. Other software processes can be carried out during wait.

<5> Set UCKCNT to 1 (UCKC Register)

USB function controller clock supply starts when UCKCNT is set to 1

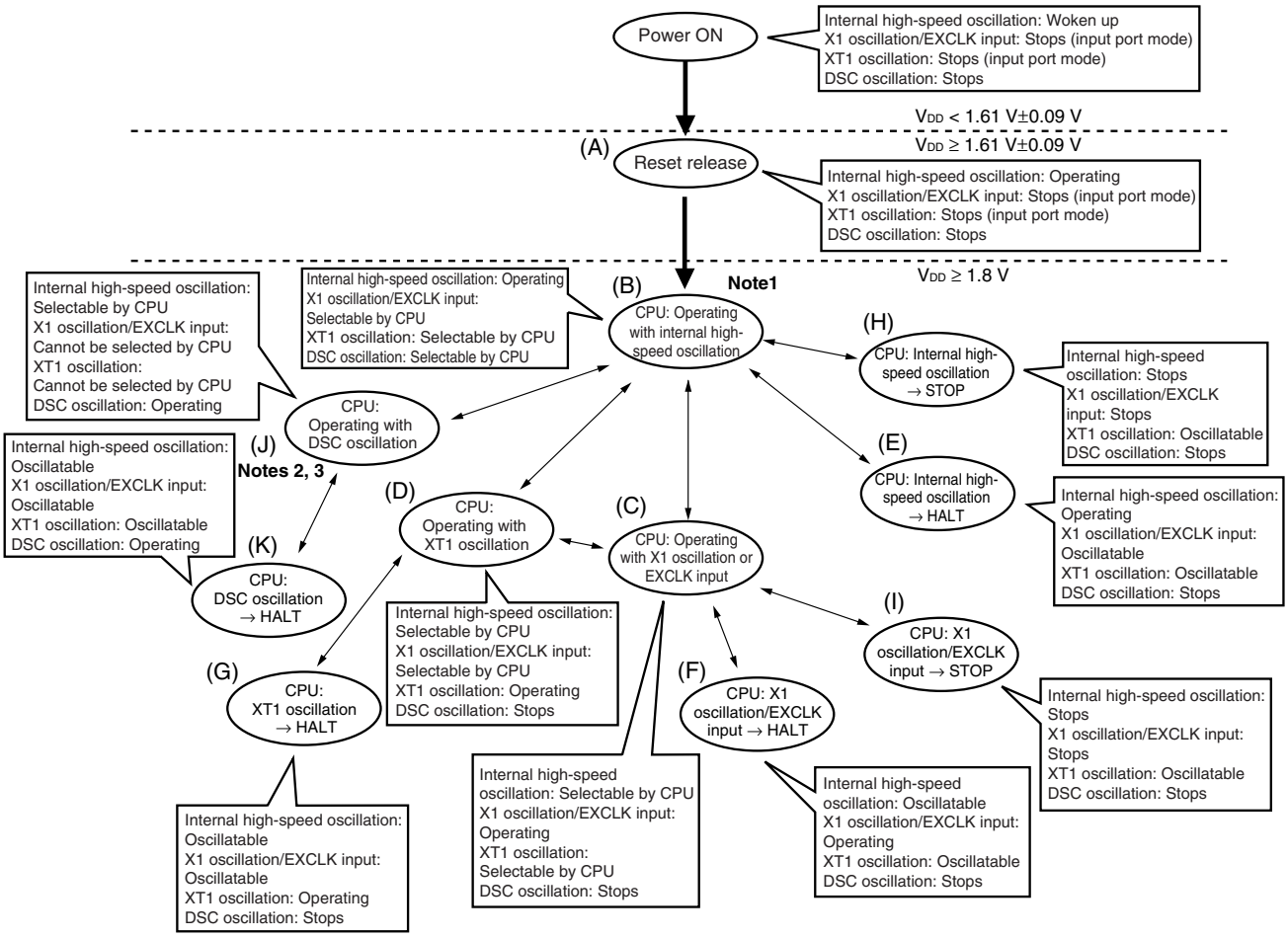
<Control flow>



5.6.6 CPU clock status transition diagram

Figure 5-17 shows the CPU clock status transition diagram of this product.

<R> **Figure 5-17. CPU Clock Status Transition Diagram**



Notes 1. After reset release, operation at 4 MHz (8 MHz/2) 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)

<R> **2.** Specify 20 MHz internal oscillation after checking that V_{DD} is at least 2.7 V.

<R> **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
<R> (A) → (B) → (C) (external main clock)	1	1	×	0	0/1 Note 2	Must not be checked	1

Notes 1. The CMC register 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 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 28 ELECTRICAL SPECIFICATIONS).

Remark ×: don't care

(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	Must be checked	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 5-17.

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
<R> (B) → (C) (external main clock)	1	1	×	**Note 2**	0	0/1 Note 3	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 fx > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

Remarks 1. ×: don't care

2. (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

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 5-17.

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 5-17.

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
<R> (D) → (C) (external main clock)	**Note 1**	0	0/1 Note 2	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 28 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 5-17.

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 stop		Sets the OSTS register	
	In X1 oscillation		–	

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

5.6.7 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 XT1 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.	–

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

By setting bits 0 to 2, 4, and 6 (MDIV0 to MDIV2, 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 Table 5-6 to Table 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.

Table 5-6. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{MAIN}	←→ (changing the division ratio)	f_{MAIN}	See Table 5-7
f_{IH}	←→	f_{MX}	See Table 5-8
f_{MAIN}	←→	f_{SUB}	See Table 5-9

Table 5-7. Maximum Number of Clocks Required for f_{MAIN} ↔ f_{MAIN} (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 for 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 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		1 + f_{IH}/f_{MX} clock
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	1 + f_{MX}/f_{IH} clock	

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

<R>

Table 5-9. Maximum Number of Clocks Required for f_{MAIN} ↔ f_{SUB}

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

Remarks 1. The number of clocks listed in Table 5-7 to Table 5-9 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-7 to Table 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

5.6.9 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	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
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

Each timer array unit has 8, 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.

Single-operation Function	Combination-operation Function
• Interval timer • Square wave output • External event counter • Divider function (channel 0 of 78K0R/KE3-L only) • Input pulse interval measurement • Measurement of high-/low-level width of input signal	• PWM output • One-shot pulse output • Multiple PWM output

Channel 7 can be used to realize LIN-bus reception processing in combination with UART3 of serial array unit 1.

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

Single-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 (INTTM0n) at fixed intervals.

(2) Square wave output

A toggle operation is performed each time INTTM0n is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0n).

(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 (TI0n) has reached a specific value.

Remark n: Channel number (n = 0 to 7)
However, in case of timer input pin (TI0n, TO0n) n changes as below.
KE3-L: n = 0-4
KC3-L: n = 1-3

(4) Divider function (channel 0 of 78K0R/KE3-L only)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).

(5) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0n). 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 (TI0n), 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 KE3-L:n = 0-4
KC3-L:n = 1-3

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 function (channel 7 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

ITE0	Configuration
Timer/counter	Timer counter register 0n (TCR0n)
Register	Timer data register 0n (TDR0n)
Timer input	At the time of 78K0R/KC3-L : TI01 – TI03 pins, RxD3 pin (for LIN-bus) At the time of 78K0R/KE3-L : TI00 – TI04 pins, RxD3 pin (for LIN-bus)
Timer output	At the time of 78K0R/KC3-L : TO01 – TO03 pins, RxD3 pin (for LIN-bus) At the time of 78K0R/KE3-L : TO00 – TO04 pins, RxD3 pin (for LIN-bus)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0) • Timer channel start register 0 (TS0) • Timer channel stop register 0 (TT0) • Timer input select register 0 (TIS0) • 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 0n (TMR0n) • Timer status register 0n (TSR0n) • Input switch control register (ISC) • Noise filter enable registers 1 (NFEN1) • Port mode registers 1, 3, 4, 5[note](PM1, PM3, PM4, PM5[note]) • Port registers 1, 3, 4, 5[note](P1, P3, P4, P5[note])

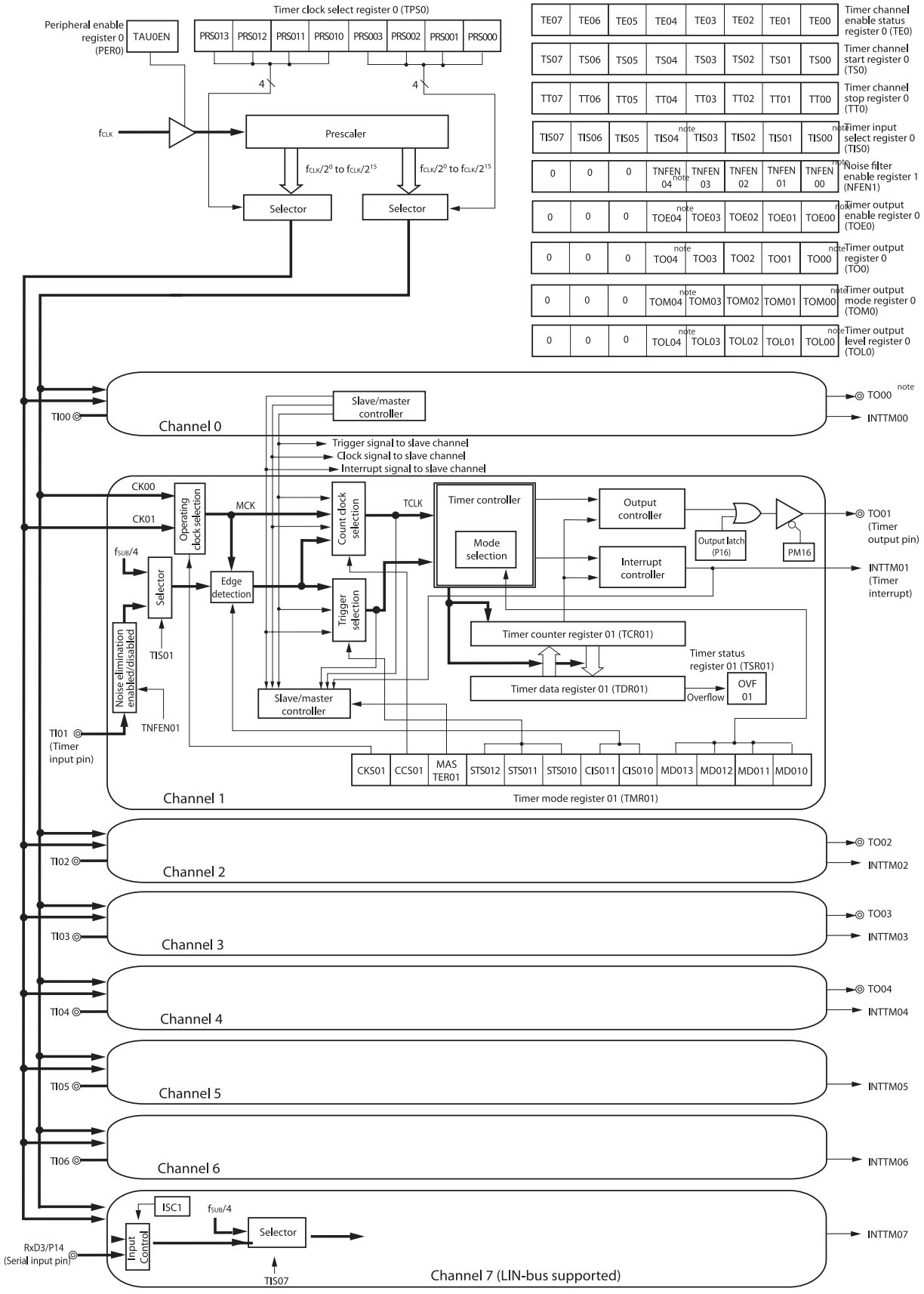
Remark n: Channel number (n = 0 to 7)

Validity of timer I/O pin of each channel of timer array unit, changes according to product.

Timer array unit channel	I/O pin of each product	
	KC3-L	KE3-L
Channel0	–	P00/TI00, P01/TO00
Channel1	P16/TI01/TO01	
Channel2	P17/TI02/TO02	
Channel3	P31/TI03/TO03	
Channel4	–	P42/TI04/TO04
Channel5	–	
Channel6	–	
Channel7	–	

Figure 6-1 shows the block diagram.

Figure 6-1. Block Diagram of Timer Array Unit



Note 78K0R/KE3-L only

(1) Timer counter register 0n (TCR0n)

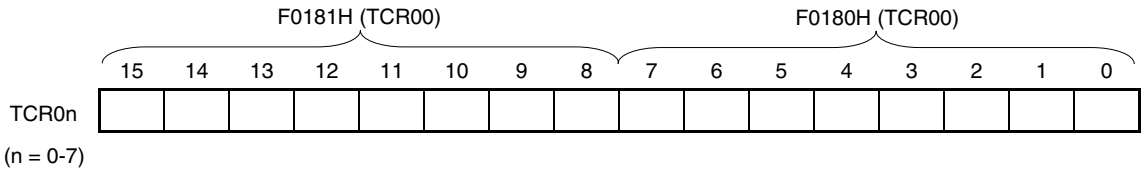
TCR0n is a 16-bit read-only register and is used to count clocks.

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 MD0n3 - MD0n0 bits of TMR0n.

Figure 6-2. Format of Timer Counter Register 0n (TCR0n)

Address: F0180H, F0181H (TCR00) - F018EH, F018FH (TCR07) After reset: FFFFH R



The count value can be read by reading TCR0n.

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

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

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

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

Caution The count value is not captured to TDR0n even when TCR0n is read.
Read out value of TCR0n register, operation mode change or current status of operation changes as below.

The TCR0n register read value differs as follows according to operation mode changes and the operating status.

Table 6-2 TCR0n Register Read Value in Various Operation Modes

Operation Mode	Count Mode	TCR0n Register Read Value[Note]			
		Operation mode change after reset	Operation mode change after count operation paused (TT0n = 1)	Operation restart after count operation paused (TT0n = 1)	During start trigger wait status after one count
Interval timer mode	Countdown	FFFFH	Undefined	Stop value	–
Capture mode	Count up	0000H	Undefined	Stop value	–
Event counter mode	Countdown	FFFFH	Undefined	Stop value	–
One-count mode	Countdown	FFFFH	Undefined	Stop value	FFFFH
Capture & one-count mode	Count up	0000H	Undefined	Stop value	Capture value of TDR0n register + 1

Note The read values of the TCR0n register when TS0n has been set to "1" while TE0n = 0 are shown. The read value is held in the TCR0n register until the count operation starts.

Remark n = 0-7

(2) Timer data register 0n (TDR0n)

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 MD0n3 to MD0n0 bits of TMR0n.

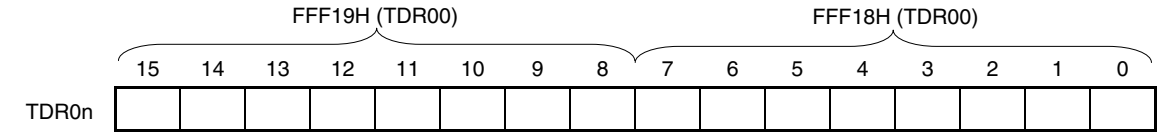
The value of TDR0n can be changed at any time.

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

Reset signal generation clears this register to 0000H.

Figure 6-3 Format of Timer Data Register 0n (TDR0n)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), FFF64H, FFF65H (TDR02) - FFF6EH, FFF6FH (TDR07) After reset: 0000H R/W



(i) When TDR0n is used as compare register

Counting down is started from the value set to TDR0n. When the count value reaches 0000H, an interrupt signal (INTTM0n) is generated. TDR0n holds its value until it is rewritten.

Caution TDR0n does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When TDR0n is used as capture register

The count value of TCR0n is captured to TDR0n when the capture trigger is input.

A valid edge of the TI0n pin can be selected as the capture trigger. This selection is made by TMR0n.

Remark n: Channel number (n = 0 - 7),

However, In case of timer input pin (TI0n), n changes as below.

KE3-L: n = 0-4

KC3-L: n = 1-3

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 0 (TPS0)
- Timer mode register 0n (TMR0n)
- Timer status register 0n (TSR0n)
- Timer channel enable status register 0 (TE0)
- Timer channel start register 0 (TS0)
- Timer channel stop register 0 (TT0)
- Timer input select register 0 (TIS0)
- 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 registers 1 (NFEN1)
- Port mode registers 1, 3, 4, 5[note] (PM1, PM3, PM4, PM5[note])
- Port registers 1, 3, 4, 5[note] (P1, P3, P4, P5[note])

Note 78K0R/KE3-L only

Remark n = 0-7

(1) Peripheral enable register 0 (PER0)

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

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

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

Reset signal generation clears this register to 00H.

Figure 6-4 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

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

Cautions 1. When setting the timer array unit, be sure to set TAU0EN = 1 first. If TAU0EN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values (Timer input selection (TIS0), input exchange control register (ISC), except for noise filter enable registers 1 (NFEN1), port mode registers 1, 3, 4, 5 note (PM1, PM3, PM4, PM5 note), and port registers 1, 3, 4, 5 note (P1, P3, P4, P5 note).

2. Be sure to clear bit 1 and 6 of PERO register to 0.

Note 78K0R/KE3-L Only

(2) Timer clock select register m (TPS0)

TPS0 is a 16-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel. CK01 is selected by bits 7 - 4 of TPS0, CK00 is selected by bits 3 - 0.

Rewriting of TPS0 during timer operation is possible only in the following cases.

Rewriting of PRS000 - PRS003 bits: Possible only when all the channels set to CKS0n = 0 are in the operation stopped state (TE0n = 0)

Rewriting of PRS010 - PRS013 bits: Possible only when all the channels set to CKS0n = 1 are in the operation stopped state (TE0n = 0)

TPS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TPS0 can be set with an 8-bit memory manipulation instruction with TPS0L.

Reset signal generation TPS0 clears this register to 0000H.

Figure 6-5 Format of Timer Clock Select Register 0 (TPS0)

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
TPS0	0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0m3	PRS 0m2	PRS 0m1	PRS 0m0	Selection of operation clock (CK0m) Note				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	10 MHz	12 MHz	16 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	5 MHz	6 MHz	8 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	2.5 MHz	3 MHz	4 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	1.25 MHz	1.5 MHz	2 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	625 kHz	750 kHz	1 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	312.5 kHz	375 kHz	500 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	156.2 kHz	187.5 kHz	250 kHz	312.5 kHz
0	1	1	1	$f_{CLK}/2^7$	78.1 kHz	93.7 kHz	125 kHz	156.2 kHz
1	0	0	0	$f_{CLK}/2^8$	39.1 kHz	46.8 kHz	62.5 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	19.5 kHz	23.4 kHz	31.2 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	9.76 kHz	11.7 kHz	15.6 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	4.88 kHz	5.85 kHz	7.81 kHz	9.76 kHz
1	1	0	0	$f_{CLK}/2^{12}$	2.44 kHz	2.92 kHz	3.91 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	1.22 kHz	1.46 kHz	1.95 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	610 Hz	732 Hz	976 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	305 Hz	366 Hz	488 Hz	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the sysTE0 clock control register (CKC) value), stop the timer array unit (TT0 = 00FFH).

Caution Be sure to clear bits 15- 8 to "0".

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

2. m = 0, 1 n = 0-7

(3) Timer mode register 0n (TMR0n)

TMR0n sets an operation mode of channel n. 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 TMR0n is prohibited when the register is in operation (when TE0 = 1). However, bits 7 and 6 (CIS0n1, CIS0n0) can be rewritten even while the register is operating with some functions (when TE0 = 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**).

TMR0n can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-6 Format of Timer Mode Register 0n (TMR0n) (1/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

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

CKS 0n	Selection of operation clock (MCK) of channel n
0	Operation clock CK00 set by TPS0 register
1	Operation clock CK01 set by TPS0 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 CCS0n bit.	

CCS 0n	Selection of count clock (TCLK) of channel n
0	Operation clock MCK specified by CKS0n bit
1	Valid edge of input signal input from TI0n pin/subsysTE0 clock divided by four (f_{SUB}/4)
Count clock TCLK is used for the timer/counter, output controller, and interrupt controller.	

MASTER 0n	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 in combination-operation function.
Only the even channel can be set as a master channel (MASTER0n = 1). Be sure to use the odd channel as a slave channel (MASTER0n = 0). Clear MASTER0n = 0 for a channel that is used with the single-operation function.	

Caution Be sure to clear bits 14, 13, 5, 4 to "0".

Remark n: Channel number (n = 0 - 7)

However, In case of timer input pin (TI0n), n changes as below.

KE3-L: n = 0-4

KC3-L: n = 1-3

Figure 6-6. Format of Timer Mode Register 0n (TMR0n) (2/3)

Address: F0190H, F0191H (TMR00) - F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) - 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

STS 0n2	STS 0n1	STS 0n0	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 TI0n pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TI0n pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination-operation function).
Other than above			Setting prohibited

CIS 0n1	CIS 0n0	Selection of TI0n pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS0n2 to STS0n0 bits is other than 010B, set the CIS0n1 to CIS0n0 bits to 10B.		

Caution Be sure to clear bits 14, 13, 5, 4 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
0n = 00 to 07, 10 to 13

Figure 6-6 Format of Timer Mode Register 0n (TMR0n) (3/3)

Address: F0190H, F0191H (TMR00) - F019EH, F019FH (TMR07) After reset: 0000H R/W

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

MD 0n3	MD 0n2	MD 0n1	MD 0n0	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 MD0n0 bit varies depending on each operation mode (see table below).						

Operation mode (Value set by the MD0n3 - MD0n1 bits (see table above))	MD 0n0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode note 1 (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 2. 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

Notes 1. In one-count mode, interrupt output (INTTM0n) when starting a count operation and TO0n output are not controlled.

2. If the start trigger (TS0n = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark n = 0-7

(4) Timer status register 0n (TSR0n)

TSR0n indicates the overflow status of the counter of channel n.

TSR0n is valid only in the capture mode (MD0n3 to MD0n1 = 010B) and capture & one-count mode (MD0n3 - MD0n1 = 110B). It will not be set in any other mode. See Table 6-3 for the operation of the OVF bit in each operation mode and set/clear conditions.

TSR0n can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TSR0n can be set with an 8-bit memory manipulation instruction with TSR0nL.

Reset signal generation clears this register to 0000H.

Figure 6-7. Format of Timer Status Register 0n (TSR0n)

Address: F01A0H, F01A1H (TSR00) - F01AEH, F01AFH (TSR07) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSR0n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Table 6-3. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF	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	–
	set	(Use prohibited, not set and not cleared)

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

(5) Timer channel enable status register m (TE0)

TE0 is used to enable or stop the timer operation of each channel.

When a bit of timer channel start register m (TS0) is set to 1, the corresponding bit of this register is set to 1. When a bit of timer channel stop register m (TT0) is set to 1, the corresponding bit of this register is cleared to 0.

TE0 can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TE0 can be set with a 1-bit or 8-bit memory manipulation instruction with TE0L.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Channel Enable Status Register 0 (TE0)

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

TE0n	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Remark n = 0-7

(6) Timer channel start register 0 (TS0)

TS0 is a trigger register that is used to clear a timer counter (TCR0n) and start the counting operation of each channel.

When a bit (TS0n) of this register is set to 1, the corresponding bit (TE0n) of timer channel enable status register m (TE0) is set to 1. TS0n is a trigger bit and cleared immediately when TE0n = 1.

TS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TS0 can be set with a 1-bit or 8-bit memory manipulation instruction with TS0L.

Reset signal generation clears this register to 0000H.

Figure 6-9. Format of Timer Channel Start Register 0 (TS0)

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

TS0n	Operation enable (start) trigger of channel n
0	No trigger operation
1	TE0n is set to 1 and the count operation becomes enabled. The TCR0n count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-4).

Caution Be sure to clear bit 15 - 8 "0"

Remarks 1. When the TS0 register is read, 0 is always read.

2. n = 0-7

Table 6-4 Operations from Count Operation Enabled State to TCR0n Count Start (1/2)

Timer operation mode	Operation when TS0n = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TS0n=1) until count clock generation. The first count clock loads the value of TDR0n to TCR0n and the subsequent count clock performs countdown operation (see **6.3 (6) (a) Start timing in interval timer mode**).
• Event counter mode	Writing 1 - TS0n bit loads the value of TDR0n to TCR0n. The subsequent count clock performs countdown operation. The external trigger detection selected by STS0n2 - STS0n0 bits in the TMR0n 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 - TCR0n and the subsequent count clock performs count up operation (see **6.3 (6) (c) Start timing in capture mode**).

Table 6-4 Operations from Count Operation Enabled State to TCR0n Count Start (2/2)

Timer operation mode	Operation when TS0n = 1 is set
• One-count mode	When TS0n = 0, writing 1 - TS0n 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 TDR0n to TCR0n and the subsequent count clock performs countdown operation (see **6.3 (6) (d) Start timing in one-count mode**).
• Capture & one-count mode	When TS0n = 0, writing 1 - TS0n 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 TCR0n and the subsequent count clock performs count up operation (see **6.3 (6) (e) Start timing in capture & one-count mode**).

(a) Start timing in interval timer mode

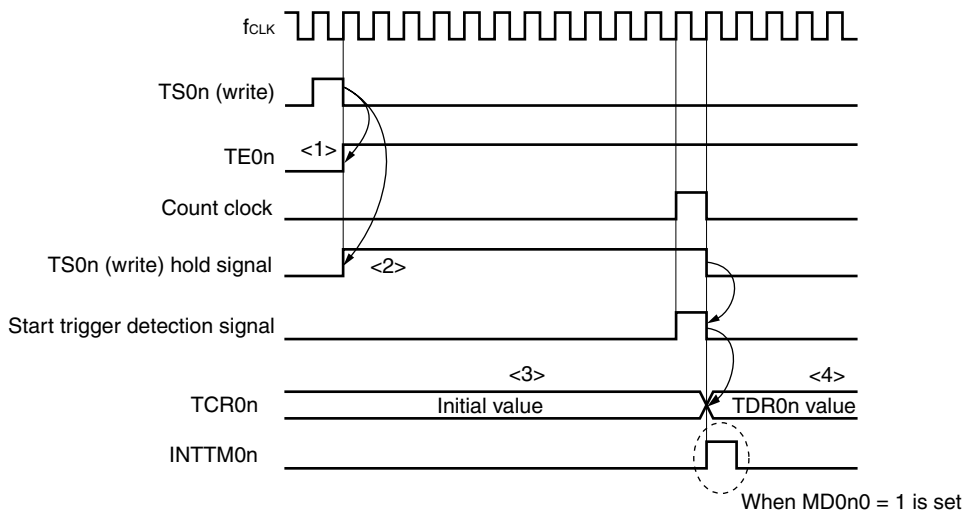
<1> Writing 1 - TS0n sets TE0n = 1

<2> The write data to TS0n is held until count clock generation.

<3> TCR0n holds the initial value until count clock generation.

<4> On generation of count clock, the “TDR0n value” is loaded to TCR0n and count starts.

Figure 6-10. Start Timing (In Interval Timer Mode)

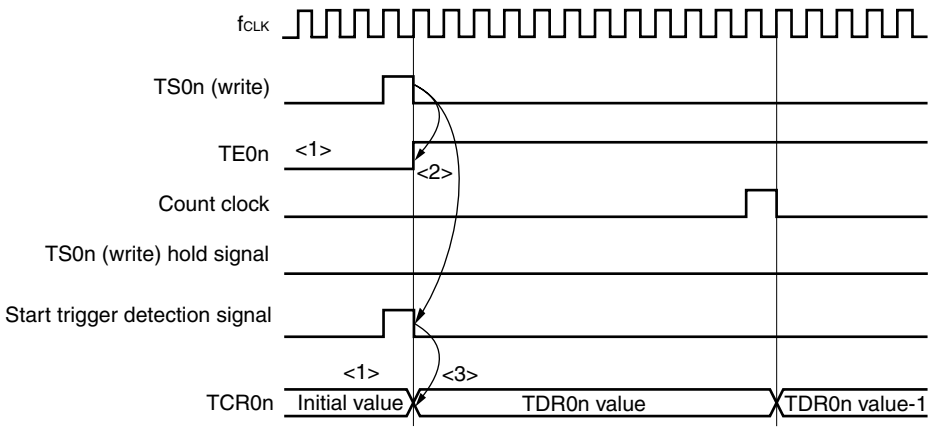


Caution In the first cycle operation of count clock after writing TS0n, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

(b) Start timing in event counter mode

<1> While TE0n is set to 0, TCR0n holds the initial value.
<2> Writing 1 - TS0n sets 1 - TE0n.
<3> As soon as 1 has been written to TS0n and 1 has been set to TE0n, the “TDR0n value” is loaded to TCR0n to start counting.
<4> After that, the TCR0n value is counted down according to the count clock.

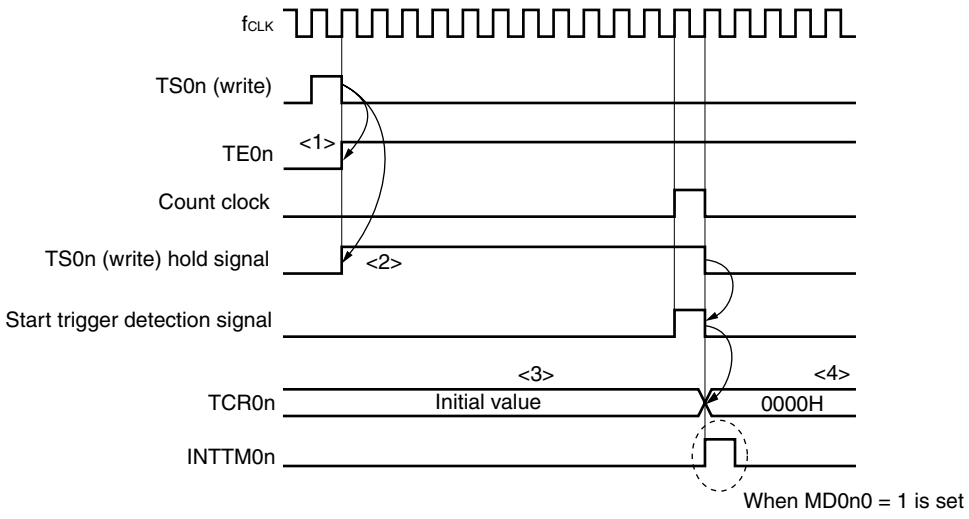
Figure 6-11. Start Timing (In Event Counter Mode)



(c) Start timing in capture mode

<1> Writing 1 - TS0n sets TE0n = 1
<2> The write data to TS0n is held until count clock generation.
<3> TCR0n holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to TCR0n and count starts.

Figure 6-12. Start Timing (In Capture Mode)



Caution In the first cycle operation of count clock after writing TS0n, 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 MD0n0 = 1.

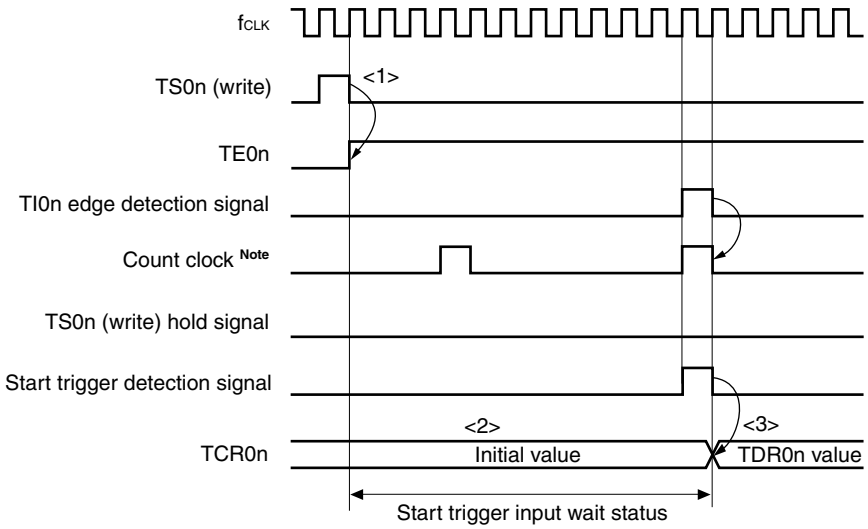
(d) Start timing in one-count mode

<1> Writing 1 - TS0n sets TE0n = 1

<2> Enters the start trigger input wait status, and TCR0n holds the initial value.

<3> On start trigger detection, the "TDR0n value" is loaded to TCR0n and count starts.

Figure 6-13. Start Timing (In One-count Mode)



Note When the one-count mode is set, the operation clock (MCK) is selected as count clock (CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TI0n is used).

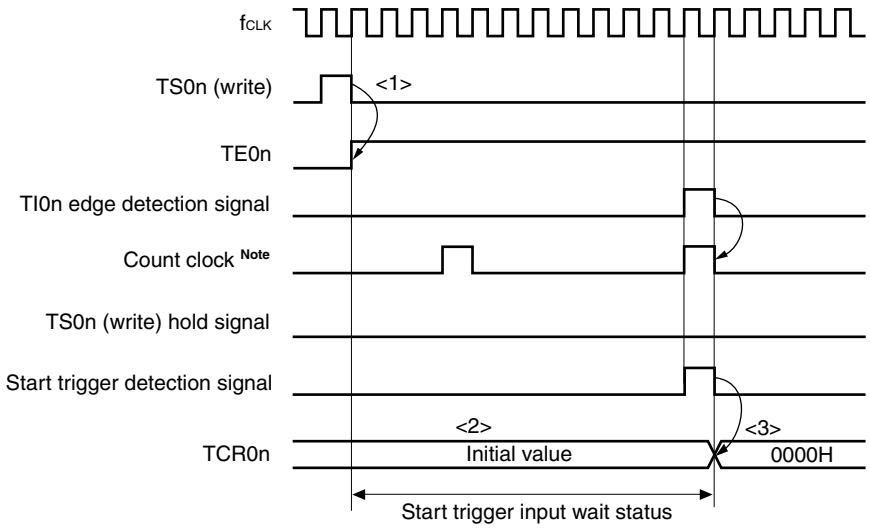
(e) Start timing in capture & one-count mode

<1> Writing 1 - TS0n sets TE0n = 1

<2> Enters the start trigger input wait status, and TCR0n holds the initial value.

<3> On start trigger detection, 0000H is loaded to TCR0n and count starts.

Figure 6-14. 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 (CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TI0n is used).

(7) Timer channel stop register m (TT0)

TT0 is a trigger register that is used to clear a timer counter (TCR0n) and start the counting operation of each channel.

When a bit (TT0n) of this register is set to 1, the corresponding bit (TE0n) of timer channel enable status register m (TE0) is cleared to 0. TT0n is a trigger bit and cleared to 0 immediately when TE0n = 0.

TT0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TT0 can be set with a 1-bit or 8-bit memory manipulation instruction with TT0L.

Reset signal generation clears this register to 0000H.

Figure 6-15. Format of Timer Channel Stop Register 0 (TT0)

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

TT0n	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 - 8 to "0".

Remarks 1. When the TT0 register is read, 0 is always read.

2. n = 0-7

(8) Timer input select register m (TIS0)

TIS0 is used to select whether a signal input to the timer input pin (TI0n) or the subsysTE0 clock divided by four (f_{SUB}/4) is valid for each channel.

TIS0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-16. Format of Timer Input Select Register 0 (TIS0)

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

TIS0n	Selection of timer input/subsysTE0 clock used with channel n
0	Input signal of timer input pin (TI0n) Channel 5,6 (In case of 78K0R/KC3-L, Channel 0, 4-6) there is no timer input pin, timer input Subsystem clock can only be divided by 4 (f_{SUB}/4). When timer input is not in use, TIS05, TIS06(In case of 78K0R/KC3-L, TIS0, TIS04-TIS06) please change it to 0 bit. In case of channel 7, it is possible to input number of Rxd3 pin (ISC1 = 1 of ISC Register).
1	Subsystem clock divided by 4 (f_{SUB}/4)

Remark **n : Chanel number (n = 0-4)**

However, In case of timer input pin (TI0n), n changes as below.

KE3-L:n = 0-4

KC3-L:n = 1-3

(9) Timer output enable register m (TOE0)

TOE0 is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TO0n 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 (TO0n).

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 6-17. 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	0	0	0	TOE04 note	TOE03 note	TOE02 note	TOE01 note	TOE00 note

TOE0n	Timer output enable/disable of channel n
0	The TO0n operation stopped by count operation (timer channel output bit). Writing to the TO0n bit is enabled. The TO0n pin functions as data output, and it outputs the level set to the TO0n bit. The output level of the TO0n pin can be manipulated by software.
1	The TO0n operation enabled by count operation (timer channel output bit). Writing to the TO0n bit is disabled (writing is ignored). The TO0n pin functions as timer output, and the TO0n bit is set or reset depending on the timer operation. The TO0n pin outputs the square-wave or PWM depending on the timer operation.

Note 78K0R/KE3-L Only

Caution Be sure to clear bits 15 - 5 bits 15 - 4 to "0".

Remark n : Channel number (n = 0-4)

However, In case of timer input pin (TO0n), n changes as below.

KE3-L:n = 0-4

KC3-L:n = 1-3

(10) Timer output register m (TO0)

TO0 is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TO0n) of each channel.

This register can be rewritten by software only when timer output is disabled (TOE0n = 0). When timer output is enabled (TOE0n = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P52/TO00[note], P16/TO01, P17/TO02, P31/TO03, P42/TO04[note], P05/TO05, P06/TO06, P54/TO07, P64/TO10, P65/TO11, P66/TO12, or P67/TO13 pin as a port function pin, set the corresponding TO0n 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 6-18. 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	0	0	0	TO04[note]	TO03	TO02	TO01	TO00[note]

TO0n	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Note 78K0R/KE3-L Only

Caution Be sure to clear bits 15 – 5 (In case of 78K0R/KC3-L) bits 15 - 4 to “0”.

Remark n : Chanel number (n = 0-4)

However, In case of timer input pin (TO0n), n changes as below.

KE3-L:n = 0-4

KC3-L:n = 1-3

(11) Timer output level register m (TOL0)

TOL0 is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOE0n = 1) in the combination-operation mode (TOM0n = 1). In the toggle mode (TOM0n = 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 6-19. 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

TOL0n	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Inverted output (active-low)

Note 78K0R/KE3-L Only

Caution Be sure to clear bits 15 - 5 (In case of 78K0R/KC3-L) bits 15 - 4 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. n = 0-4

(12) Timer output mode register m (TOM0)

TOM0 is used to control the timer output mode of each channel.

When a channel is used for the single-operation function, set the corresponding bit of the channel to be used to 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 master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOE0n = 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 6-20. 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
TO00	0	0	0	0	0	0	0	0	TO0 07	TO0 06	TO0 05	TO0 04	TO0 03	TO0 02	TO0 01	TO0 00

TOM 0n	Control of timer output mode of channel n
0	Toggle mode (to produce toggle output by timer interrupt request signal (INTTM0n)
1	Combination-operation mode (set by the timer interrupt request signal (INTTM0m) of the master channel, and reset by the timer interrupt request signal (INTTM0m) of the slave channel)

Note 78K0R/KE3-L Only

Caution Be sure to clear bits 15 - 5 (In case of 78K0R/KC3-L) bits 15 - 4 to "0".

Remark n: Chanel number, m: slave channel number
n = 0-4 (after master channel: n=0, 2)
n<m ≤ 4 (however m is a connected integer from n onwards)

(13) Input switch control register (ISC)

ISC is used to implement LIN-bus communication operation with channel 7 in association with serial array unit.

When bit 1 of ISC1 bit 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 6-21. 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	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input.
1	Input signal of RxD3 pin is used as timer input (to measure the pulse widths of the sync break field and sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD3 pin as an external interrupt (wakeup signal detection)

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

Remark When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 to 1.

(14) Noise filter enable registers 1 (NFEN1)

NFEN1 is used to set whether the noise filter can be used for the timer input signal of timer array unit 0 to 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 of the 2 clocks and synchronization are 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 6-22. 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[note]	TNFEN03	TNFEN02	TNFEN01	TNFEN00[note]

TNFEN04[note]	Enable/disable using noise filter of TI04/TO04/P42 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/INTP4/P31 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/TO02/P17 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01/TO01/INTP5/P16 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00[note]	Enable/disable using noise filter of TI00/P53 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note 78K0R/KE3-L Only

(15) Port mode registers 1 ,3 ,4 , 5[note] (PM1, PM3, PM4, PM5[note])

These registers set input/output of ports 1, 3, 4, 5[note] in 1-bit units.
When using the P52/TO00, P17/TO02/TI02 for timer output, set (PMxx) and the output latches of (Pxx) to 67 to 0.
For example:
In case of using P10/TO02/TI02 as timer output,
Set 1 of port mode register to 0.
Set P10 of port register 1 to 1.

When using the (P53/TI00, P10/TO02/TI02) pins for timer input, set (PMxx) to 1. At this time, the output latches of (Pxx) may be 0 or 1.
For example:
In case of using P10/TO02/TI02 as timer output,
Set 1 of port mode register to 0.
Set P10 of port register 1 to 0.

PM1, PM3, PM4, PM5[note]can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Note 78K0R/KE3-L Only

Figure 6-23. Format of Port Mode Registers 1, 3, 4, 5[note] (PM1, PM3, PM4, PM5[note])

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	0	0	1	0	PM43	PM42	PM41	PM40

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	0	0	PM53	PM52	PM51	PM50

PMmn	Pmn pin I/O mode selection (m = 1, 3, 4, 5 ; n = 0-7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

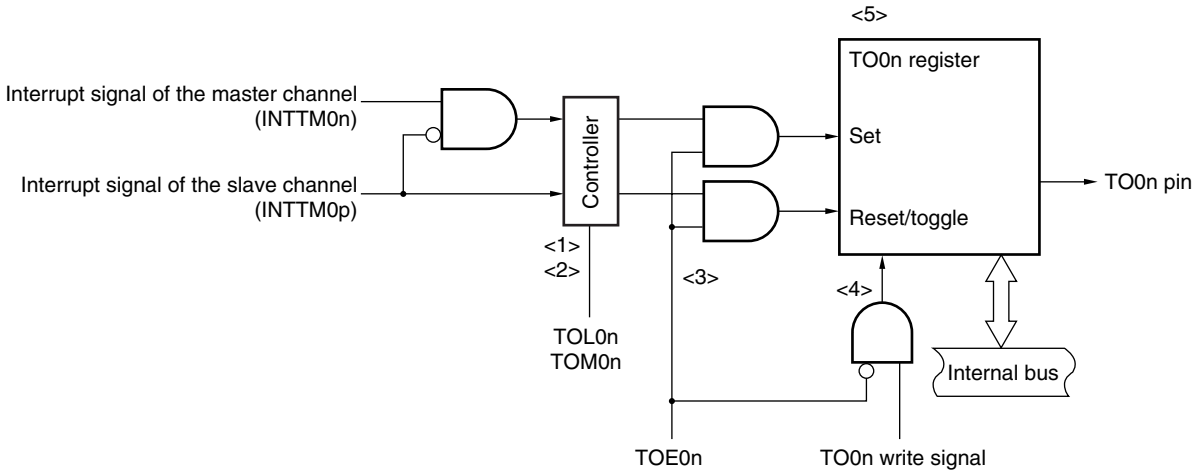
Note 1 Be sure to set 0 bit of PM3, 4, 6, 7 bit of PM4, 4,5 bit of PM5 to 0. Be sure to set, when it returns to initial value while resetting, to 0.
2 Be sure to set 2-7 bit of PM3, 5 bit of PM4, 6, 7 bit of PM5 to 0.

Remark The format of port mode register 1, 3, 4, 5 of 78K0R/KE3-L is given above. Please refer 4.3 (1) port mode register (PMxx) for other products.

6.4 Channel Output (TO0n pin) Control

6.4.1 TO0n pin output circuit configuration

Figure 6-24. Output Circuit Configuration



The following describes the TO0n pin output circuit.

<1> When TOM0n = 0 (toggle mode), the set value of the TOL0n bit is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to the TO0n bit.

<2> When TOM0n = 1 (combination-operation mode), both INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0n bit.
At this time, the TOL0n bit becomes valid and the signals are controlled as follows:

When TOL0n = 0: Forward operation (INTTM0n → set, INTTM0p → reset)
When TOL0n = 1: Reverse operation (INTTM0n → reset, INTTM0p → set)

When INTTM0n and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTM0n (set signal) is masked.

<3> When TOE0n = 1, INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0n bit. Writing to the TO0n bit (TO0n write signal) becomes invalid.
When TOE0n = 1, the TO0n pin output never changes with signals other than interrupt signals.
To initialize the TO0n pin output level, it is necessary to set TOE0n = 0 and to write a value to TO0n.

<4> When TOE0n = 0, writing to TO0n bit to the target channel (TO0n signal) becomes valid. When TOE0n = 0 neither INTTM0n (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to TO0n bit.

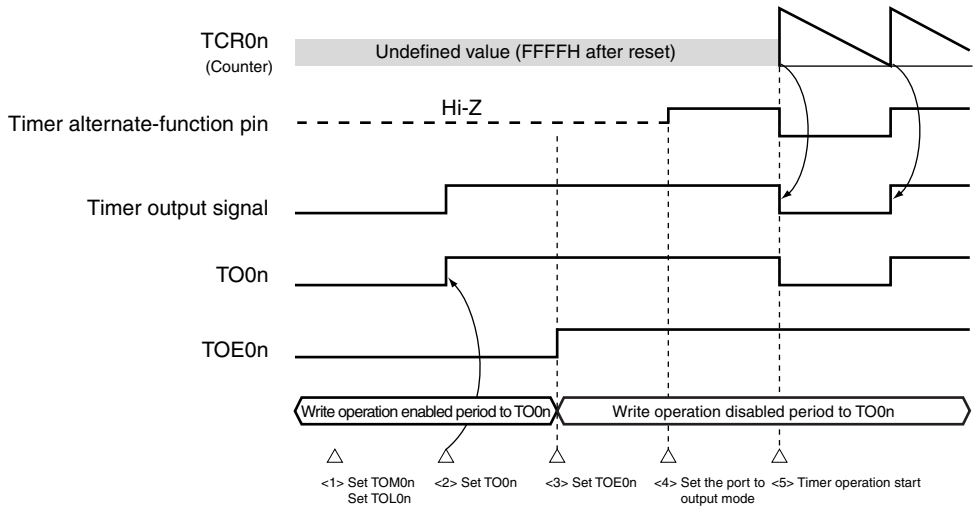
<5> The TO0n bit can always be read, and the TO0n pin output level can be checked.

Remark 1. n=0-4 (However in case of master channel n=0,2)
2. In case of timer output pin (TO0n), n changes as below.
KE3-L:n = 0-4, KC3-L:n = 1-3
3. p = n+1, n+2, n+3 … However p ≤ 4

6.4.2 TO0n Pin Output Setting

The following figure shows the procedure and status transition of TO0n output pin from initial setting to timer operation start.

Figure 6-25. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOM0n bit (0: Toggle mode, 1: Combination-operation mode)
- TOL0n bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting TO0n.
<3> The timer output operation is enabled by writing 1 - TOE0n (writing to TO0n is disabled).
<4> The port I/O setting is set to output (see **6.3 (15) Port mode registers 1, 3, 4, 5**)
<5> The timer operation is enabled (TS0n = 1).

Remark n = 0-4

However in case of (TO0n) n changes as below.

KE3-L:n = 0-4

KC3-L:n = 1-3

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 TCR0n, TDR0n) are independent of the TO0n output circuit and changing the values set in TO0, TOE0, TOL0,TOM0 does not affect the timer operation, the values can be changed during timer operation.

When the values set in TOE0, TOL0, TOM0 (except for TO0) are changed close to the timer interrupt (INTTM0n), the waveform output to the TO0n pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0n) signal generation timing.

Remark n = 0-7

However in case of (TO0n) n changes as below.

KE3-L:n = 0-4

KC3-L:n = 1-3

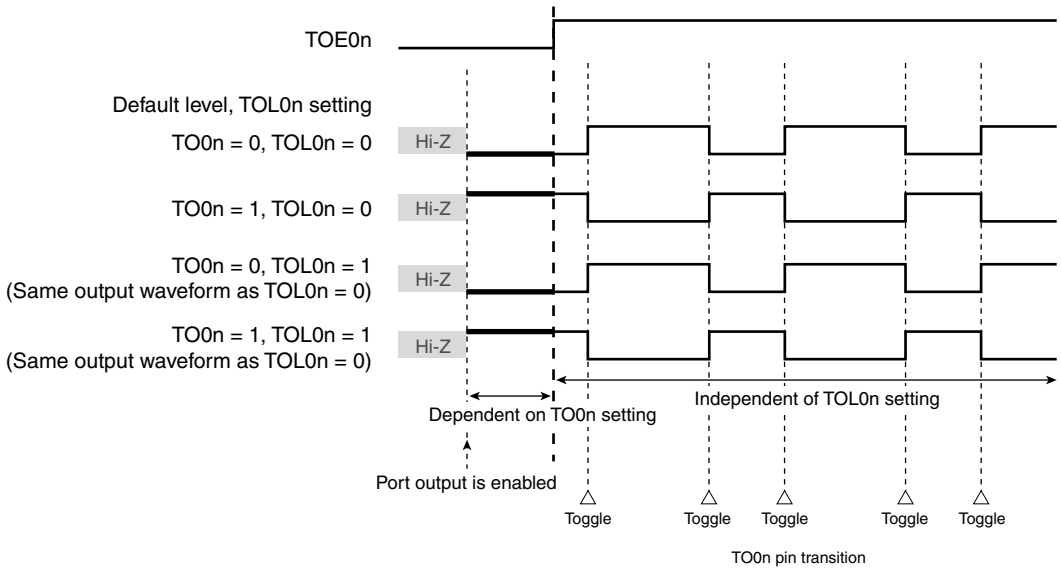
(2) Default level of TO0n pin and output level after timer operation start

The following figure shows the TO0n pin output level transition when writing has been done in the state of TOE0n = 0 before port output is enabled and TOE0n = 1 is set after changing the default level.

(a) When operation starts with TOM0n = 0 setting (toggle output)

The setting of TOL0n is invalid when TOM0n = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TO0n pin is reversed.

Figure 6-26. TO0n Pin Output Status at Toggle Output (TOM0n = 0)



Remarks 1. Toggle: Reverse TO0n pin output status

2. n = 0-4

However in case of (TO0n) n changes as below.

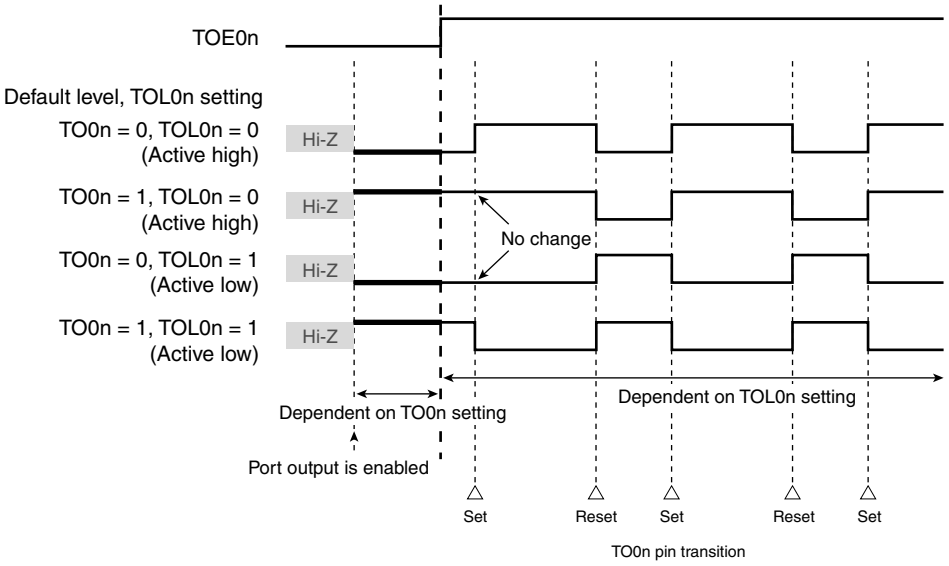
KE3-L:n = 0-4

KC3-L:n = 1-3

(b) When operation starts with TOM0n = 1 setting (Combination-operation mode (PWM output))

When TOM0n = 1, the active level is determined by TOL0n setting.

Figure 6-27. TO0n Pin Output Status at PWM Output (TOM0n = 1)



Remarks 1. Set: The output signal of TO0n pin changes from inactive level to active level.

Reset: The output signal of TO0n pin changes from active level to inactive level.

2. n = 0-4

However in case of (TO0n) nchanges as below.

KE3-L:n = 0-4

KC3-L:n = 1-3

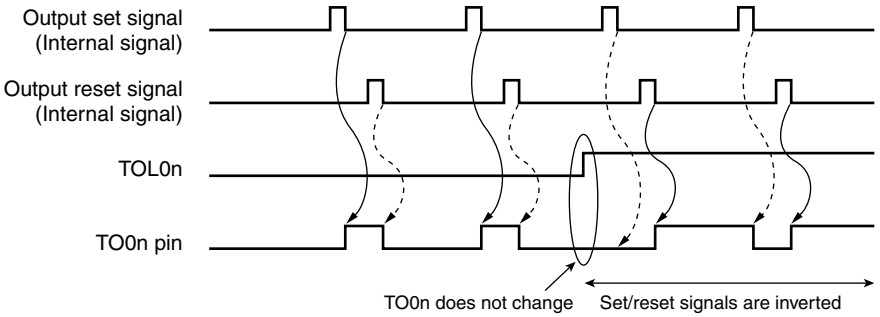
(3) Operation of TO0n pin in combination-operation mode (TOM0n = 1)

(a) When TOL0n setting has been changed during timer operation

When the TOL0n setting has been changed during timer operation, the setting becomes valid at the generation timing of TO0n change condition. Rewriting TOL0n does not change the output level of TO0n.

The following figure shows the operation when the value of TOL0n has been changed during timer operation (TOM0n = 1).

Figure 6-28 Operation when TOL0n Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of TO0n pin changes from inactive level to active level.

Reset: The output signal of TO0n pin changes from active level to inactive level.

2. n = 0-4

However in case of (TO0n) n changes as below.

KE3-L:n = 0-4

KC3-L:n = 1-3

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TO0n pin/TO0n set timing at master channel timer interrupt (INTTM0n) generation is delayed by 1 count clock by the slave channel.

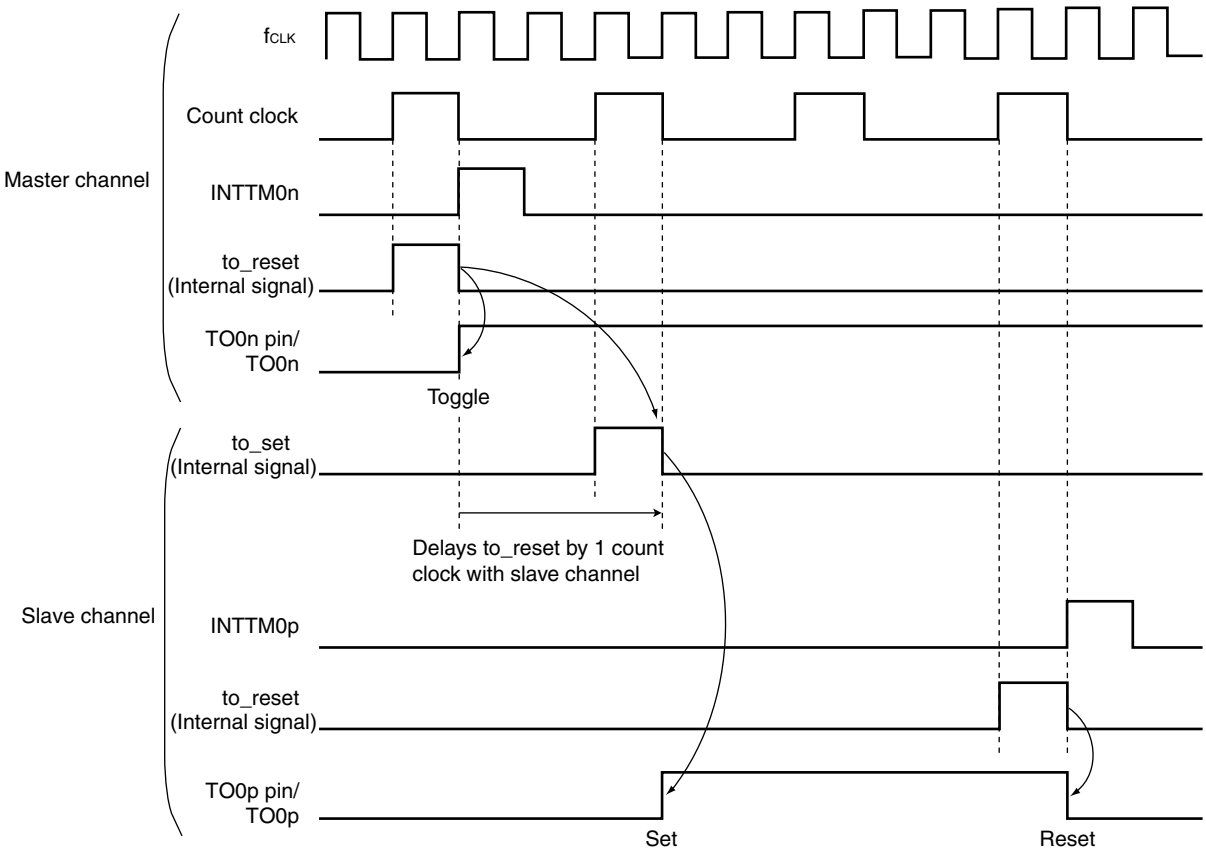
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-31 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel: TOE0n = 1, TOM0n = 0, TOL0n = 0

Slave channel: TOE0p = 1, TOM0p = 1, TOL0p = 0

Figure 6-29. Set/Reset Timing Operating Statuses



Remarks 1. to_reset: TO0n pin reset/toggle signal

to_set: TO0n pin set signal

2. n = 0-4 (However in case of master channel n = 0, 2)

3. In case of timer output pin (TO0n), n changes as below.

KE3-L:n = 0-4

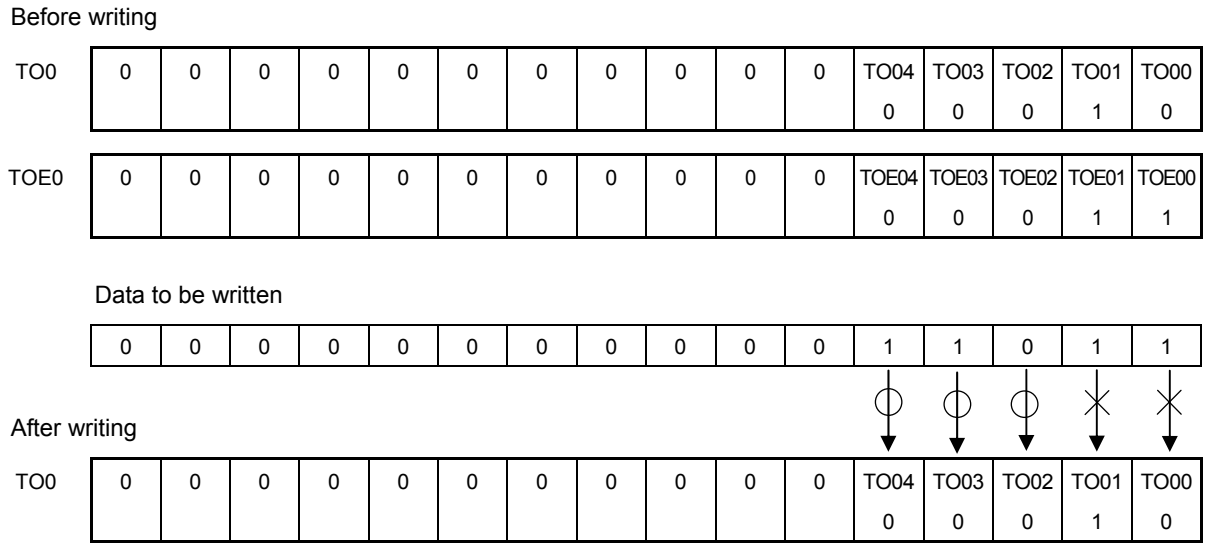
KC3-L:n = 1-3

4. p = n+1, n+2, n+3…
(However p ≤ 4)

6.4.4 Collective manipulation of TO0n 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, TO0n of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOE0n = 0 to a target TO0n (channel output).

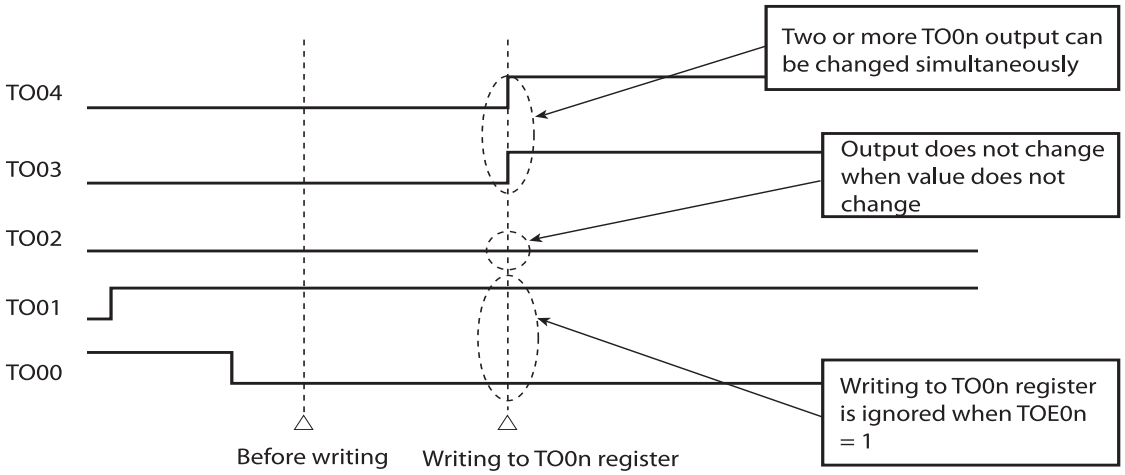
Figure 6-30. Example of TO0n Bits Collective Manipulation



Writing is done only to TO0n bits with TOE0n = 0, and writing to TO0n bits with TOE0n = 1 is ignored.

TO0n (channel output) to which TOE0n = 1 is set is not affected by the write operation. Even if the write operation is done to TO0n, it is ignored and the output change by timer operation is normally done.

Figure 6-31. TO0n Pin Statuses by Collective Manipulation of TO0n Bits



Caution When TOE0n = 1, even if the output by timer interrupt of each timer (INTTM0n) contends with writing to TO0n, output is normally done to TO0n pin.

Remark n = 0-7
However In case of timer output pin (TO0n), n changes as below.
KE3-L:n = 0-4
KC3-L:n = 1-3

6.4.5 Timer Interrupt and TO0n Pin Output at Operation Start

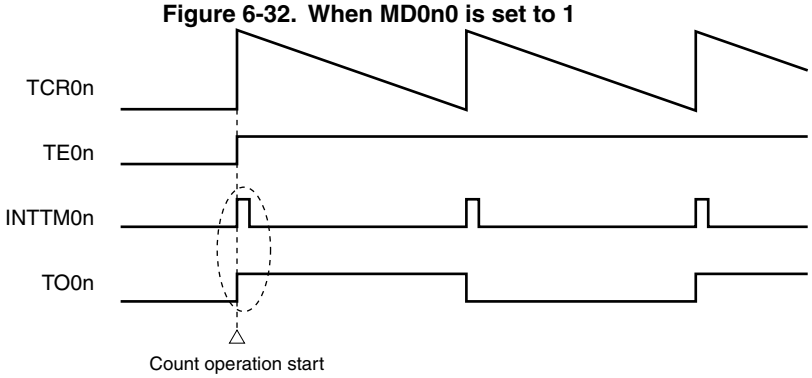
In the interval timer mode or capture mode, the MD0n0 bit in the TMR0n register sets whether or not to generate a timer interrupt at count start.

When MD0n0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTM0n) generation.

In the other modes, neither timer interrupt at count operation start nor TO0n output is controlled.

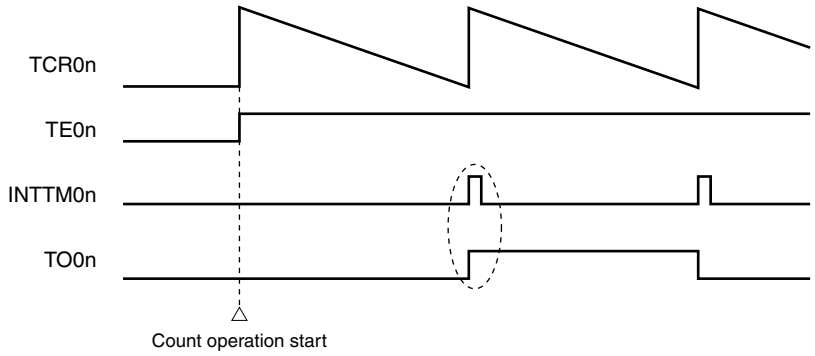
Below shows operation examples when the interval timer mode (TOE0n = 1, TOM0n = 0) is set.

Figure 6-32. When MD0n0 is set to 1



When MD0n0 is set to 1, a timer interrupt (INTTM0n) is output at count operation start, and TO0n performs a toggle operation.

Figure 6-33. When MD0n0 is set to 0



When MD0n0 is set to 0, a timer interrupt (INTTM0n) is not output at count operation start, and TO0n does not change either. After counting one cycle, INTTM0n is output and TO0n performs a toggle operation.

Remark **n = 0-7**
However In case of timer output pin (TO0n), n changes as below.
KE3-L:n = 0-4
KC3-L:n = 1-3

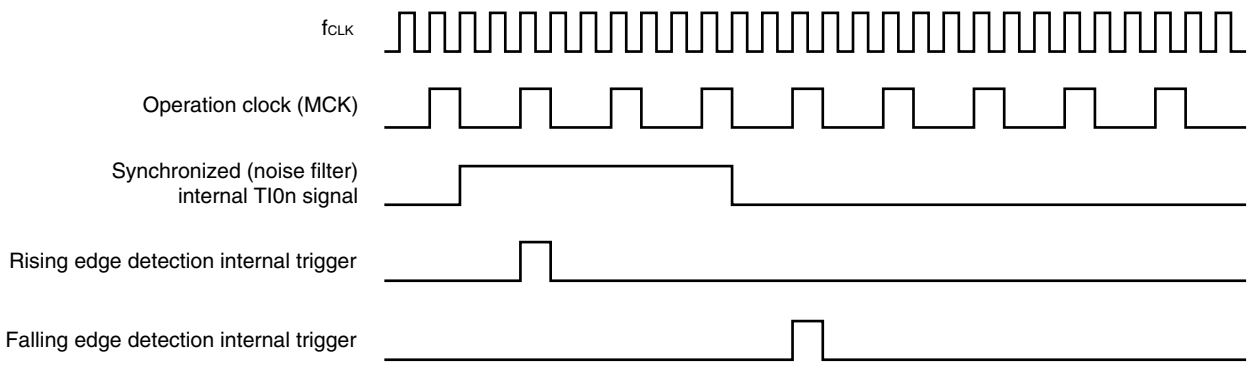
6.5 Channel Input (TI0n Pin) Control

6.5.1 TI0n edge detection circuit

(1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (MCK).

Figure 6-34. Edge Detection Basic Operation Timing



Remark **n = 0-4**
However In case of timer output pin (TI0n), n changes as below.
KE3-L:n = 0-4
KC3-L:n = 1-3

6.6 Basic Function of Timer Array Unit

6.6.1 Overview of single-operation function and combination-operation function

The timer array unit (Below known as TAU) 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.

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, channel 2, channel 4, etc.) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 2 is set as a master channel, channel 3 or those that follow (channel 3, channel 4, channel 5….) can be set as a slave channel.
If channel 0 is set as a master channel, channel 1, 2, or 3 (only up to channel 3 on TAU1.) can be set as a slave channel.

(4) Multiple 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, 4 of the TAU0 are set as master channels, channels 1 - 3 can be set as the slave channels of master channel 0. Channels 5 - 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 other 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.

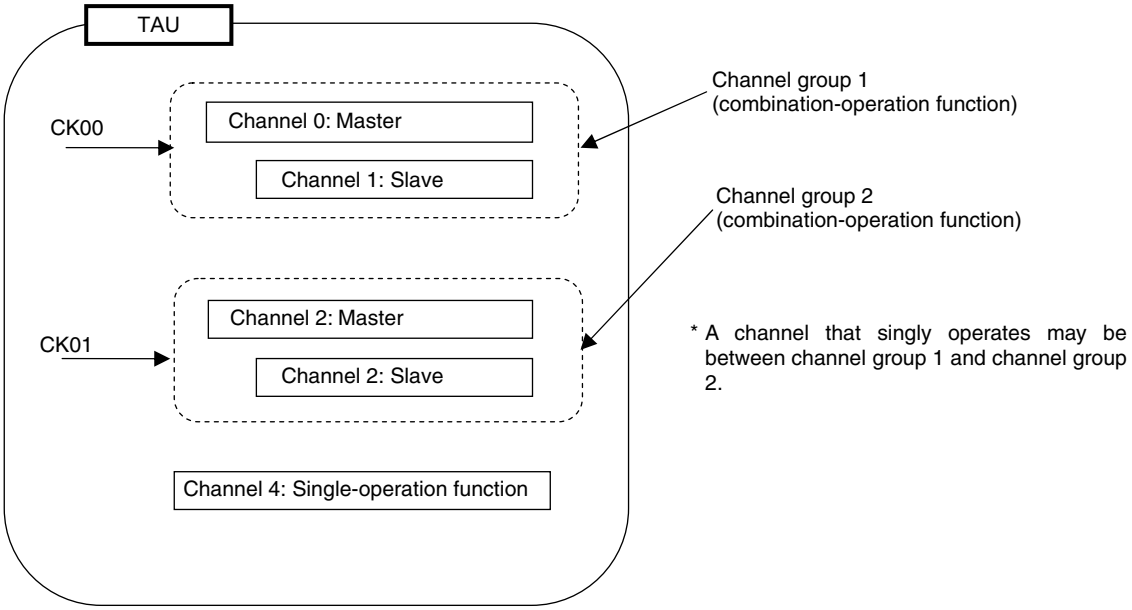
Remark n = 0-7

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 Operation 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 INTTM0n (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

A subsystem clock divided by four ($f_{SUB}/4$) can be selected as the count clock, in addition to CK00 and CK01. Consequently, the interval timer can be operated with the count clock fixed to $f_{SUB}/4$, 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)), however, stop the timer array unit (TAU) (TT0 = 00FFH) first.

(2) Operation as square wave output

TO0n performs a toggle operation as soon as INTTM0n has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0n can be calculated by the following expressions.

- Period of square wave output from TO0n = Period of count clock × (Set value of TDR0n + 1) × 2
- Frequency of square wave output from TO0n = Frequency of count clock/{(Set value of TDR0n + 1) × 2}

TCR0n operates as a down counter in the interval timer mode.

TCR0n loads the value of TDR0n at the first count clock after the channel start trigger bit (TS0n) is set to 1. If MD0n0 of TMR0n = 0 at this time, INTTM0n is not output and TO0n is not toggled. If MD0n0 of TMR0n = 1, INTTM0n is output and TO0n is toggled.

After that, TCR0n count down in synchronization with the count clock.

When TCR0n = 0000H, INTTM0n is output and TO0n is toggled at the next count clock. At the same time, TCR0n loads the value of TDR0n again. After that, the same operation is repeated.

TDR0n can be rewritten at any time. The new value of TDR0n becomes valid from the next period.

Remark 1. n = 0-7

However in case of square wave output operation or timer output pin(TO0n)n changes as below

KE3-L:n = 0-4

KC3-L:n = 1-3

2. f_{CLK} :CPU/Peripheral hardware clock frequency

f_{SUB} :Subsystem clock oscillation frequency

Figure 6-35. Block Diagram of Operation as Interval Timer/Square Wave Output

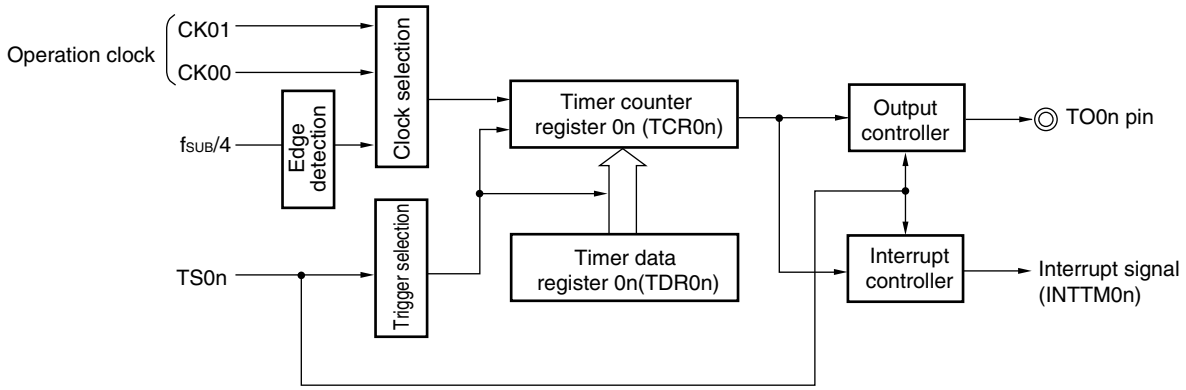
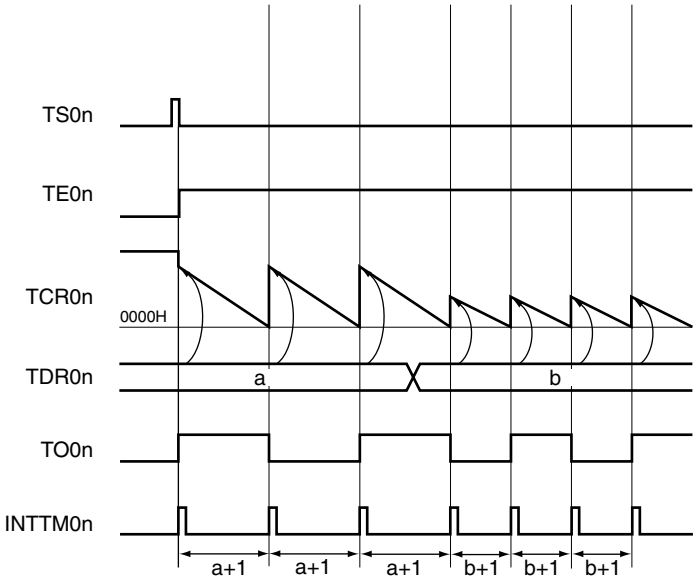


Figure 6-36. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)



Remark n = 0-7

However in case of square wave output operation or timer output pin(TO0n)n changes as below

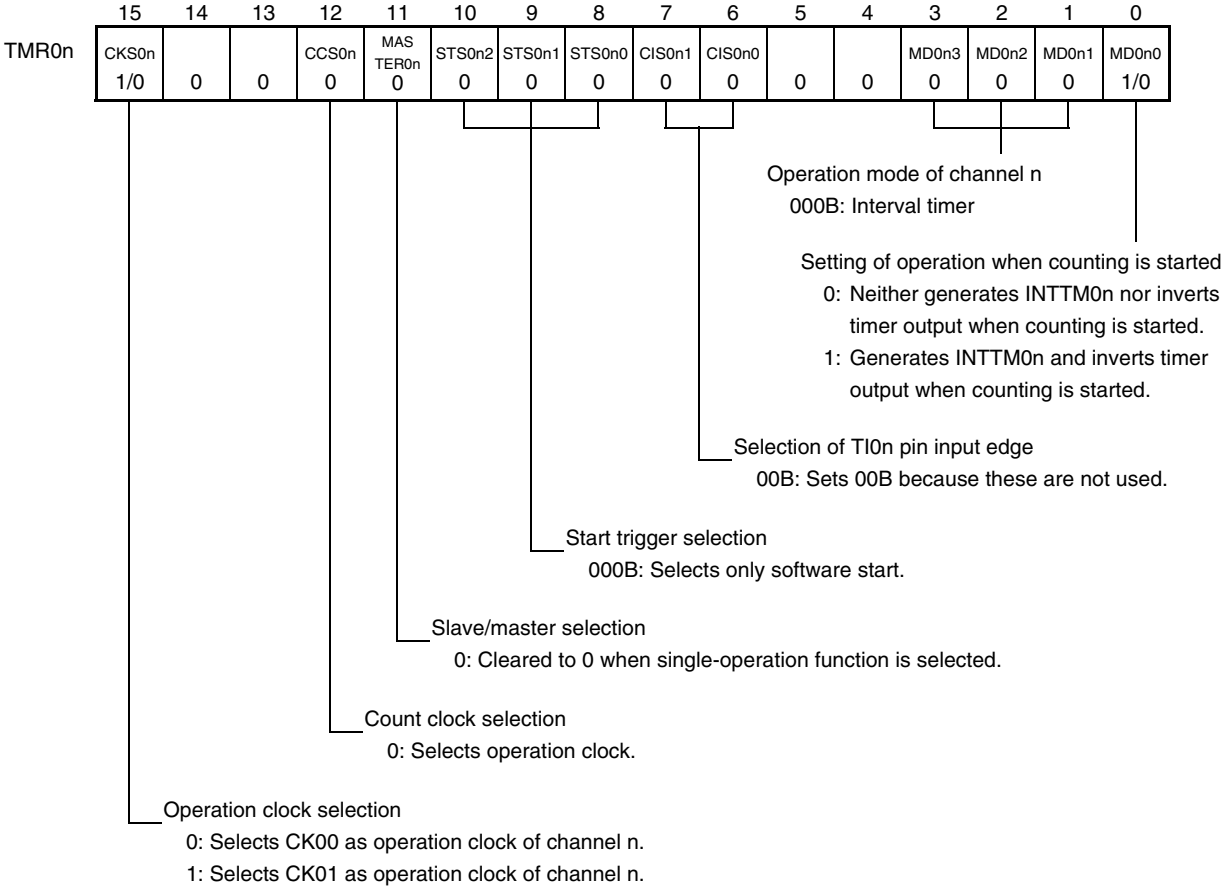
KE3-L:n = 0-4

KC3-L:n = 1-3

Figure 6-37. Example of Set Contents of Registers during Operation as Interval Timer/Square Wave Output (1/3)

(1) When CK00, CK01 is selected as count clock

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 1/0	0: Outputs 0 from TO0n. 1: Outputs 1 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 1/0	0: Stops the TO0n output operation by counting operation. 1: Enables the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (toggle mode)

(e) Timer output mode register 0 (TOM0)

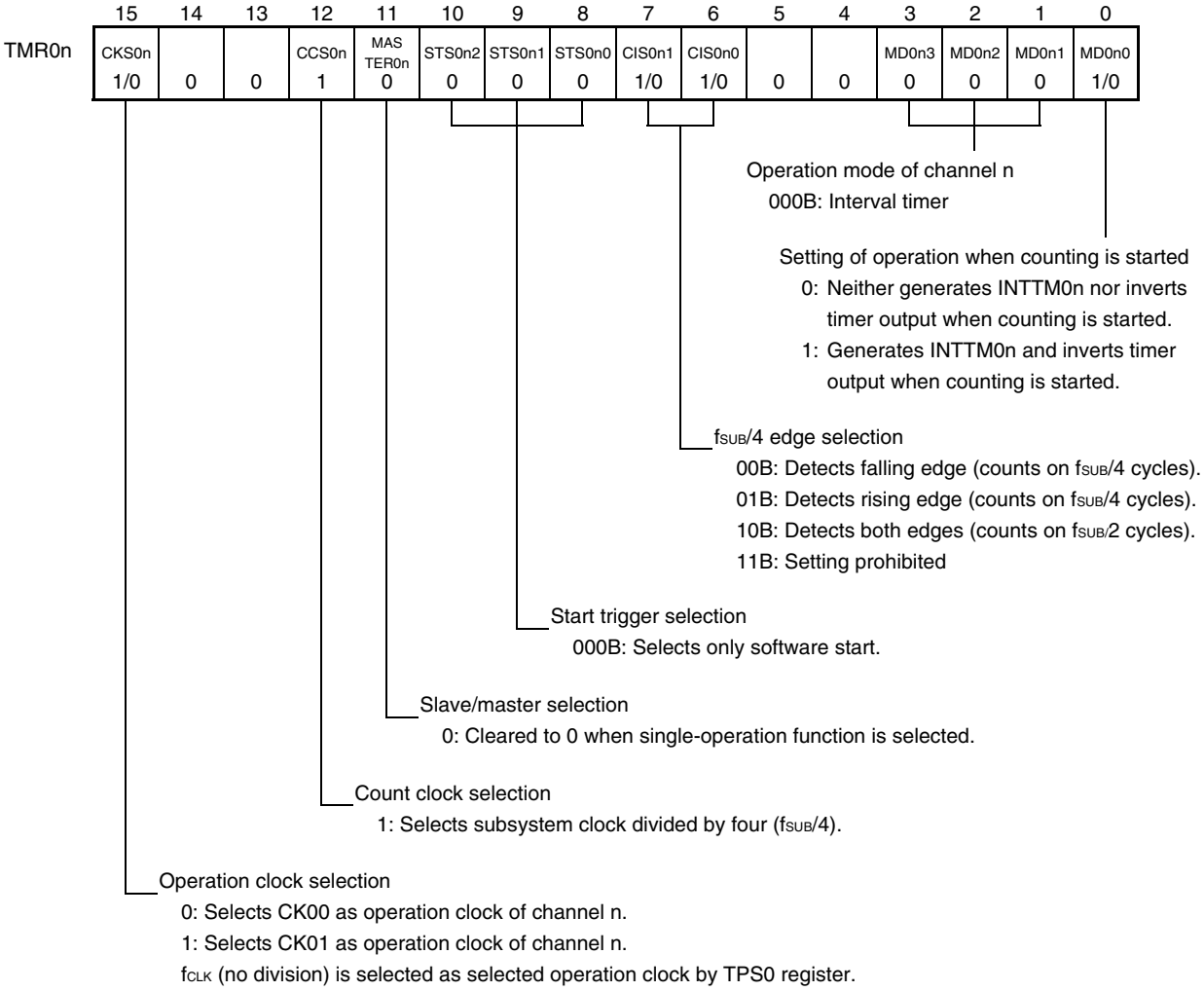
	Bit n	
TOM0	TOM0n 0	0: Sets toggle mode.

Remark n = 0-7,
However in case of square wave output operation or timer output pin(TO0n)n changes as below
KE3-L:n = 0-4
KC3-L:n = 1-3

Figure 6-37. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/3)

(2) When f_{SUB}/4 is selected as count clock

(a) Timer mode register 0n (TMR0n)



(b) Timer clock select register m0 (TPS0)

Bits 7 - 4, 3 – 0

TPS0	PRS0k3 to PRS0k0
	0000

0000B: Selects f_{CLK} (no division) as operation clock selected by CKS0n of TMR0n register.
k = 0 (bits 0 to 3) when CK00 is selected and k = 1 (bits 4 to 7) when CK01 is selected

(c) Timer input select register m (TIS0)

Bit n

TIS0	TIS0n
	1

1: Selects subsystem clock divided by four (f_{SUB}/4).

(d) Timer output register m (TO0)

Bit n

TO0	TO0n
	1/0

0: Outputs 0 from TO0n.
1: Outputs 1 from TO0n.

Remark 1. n = 0-7, k = 0, 1

However in case of square wave output operation or timer output pin(TO0n)n changes as below

KE3-L:n = 0-4

KC3-L:n = 1-3

2. f_{SUB}:Subsystem clock oscillation frequency

Figure 6-37. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (3/3)

(2) When f_{SUB}/4 is selected as count clock (continued)

(e) Timer output enable register m (TOE0)

	Bit n	
TOE0	TOE0n 1/0	0: Stops the TO0n output operation by counting operation. 1: Enables the TO0n output operation by counting operation.

(f) Timer output level register m (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (toggle mode)

(g) Timer output mode register m (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets toggle mode.

Remark n = 0-7

However in case of square wave output operation or timer output pin(TO0n)n changes as below

KE3-L:n = 0-4

KC3-L:n = 1-3

Figure 6-38. Operation Procedure of Interval Timer/Square Wave Output Function

	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 register (determines operation mode of channel). Sets the TIS0n bit to 1 (f_{SUB}/4) when f_{SUB}/4 is selected as the count clock. Sets interval (period) value to the TDR0n register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TO0n output Clears the TOM0n bit of the TOM0 register to 0 (toggle mode). Clears the TOL0n bit to 0. Sets the TO0n bit and determines default level of the TO0n output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOE0n to 1 and enables operation of TO0n. →	TO0n does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0n pin outputs the TO0n set level.
Operation start	Sets TOE0n to 1 (only when operation is resumed). Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Value of TDR0n is loaded to TCR0n at the count clock input. INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0n register is 1.
During operation	Set values of TMR0n, TOM0, and TOL0 registers cannot be changed. Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TO0 and TOE0 registers can be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of TDR0n is loaded to TCR0n again and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops. The TO0n output is not initialized but holds current status.
	TOE0n is cleared to 0 and value is set to TO0n bit. →	The TO0n pin outputs the TO0n set level.
TAU stop	To hold the TO0n pin output level Clears TO0n bit to 0 after the value to be held is set to the port register. → When holding the TO0n pin output level is not necessary Switches the port mode register to input mode. →	The TO0n pin output level is held by port function. The TO0n pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0n bit is cleared to 0 and the TO0n pin is set to port mode.)

Operation is resumed.

Remark n = 0-7

However in case of square wave output operation or timer output pin(TO0n)n changes as below

KE3-L:n = 0-4

KC3-L:n = 1-3

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 TI0n 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 TDR0n + 1

TCR0n operates as a down counter in the event counter mode.

When the channel start trigger bit (TS0n) is set to 1, TCR0n loads the value of TDR0n.

TCR0n counts down each time the valid input edge of the TI0n pin has been detected. When TCR0n = 0000H, TCR0n loads the value of TDR0n again, and outputs INTTM0n.

After that, the above operation is repeated.

TO0n must not be used because its waveform depends on the external event and irregular.

TDR0n can be rewritten at any time. The new value of TDR0n becomes valid during the next count period.

Figure 6-39. Block Diagram of Operation as External Event Counter

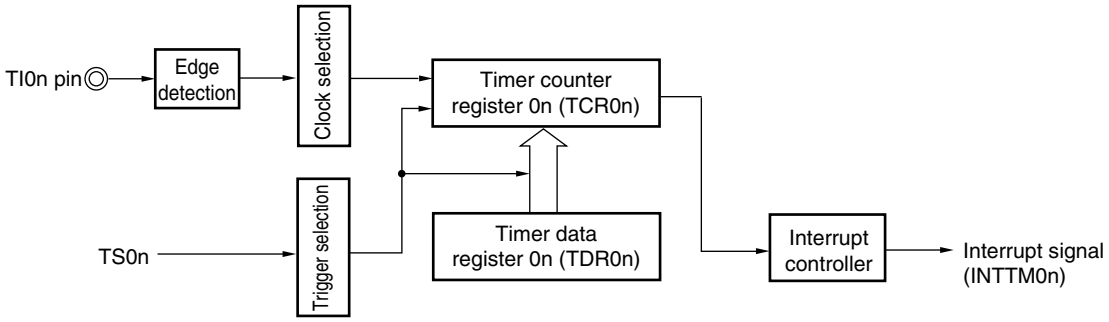
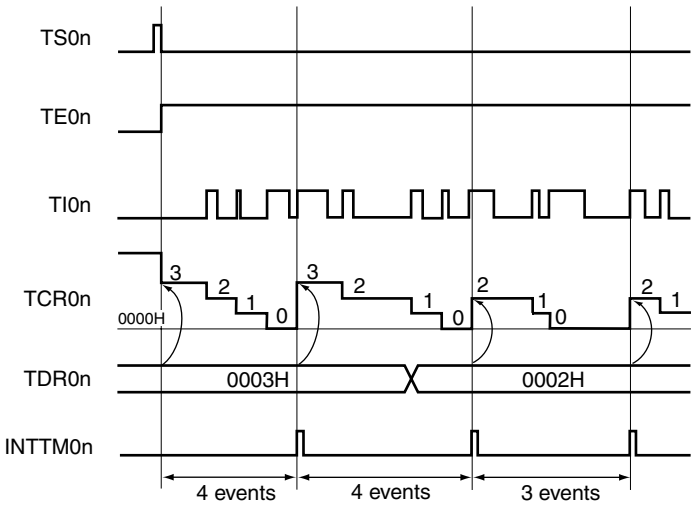


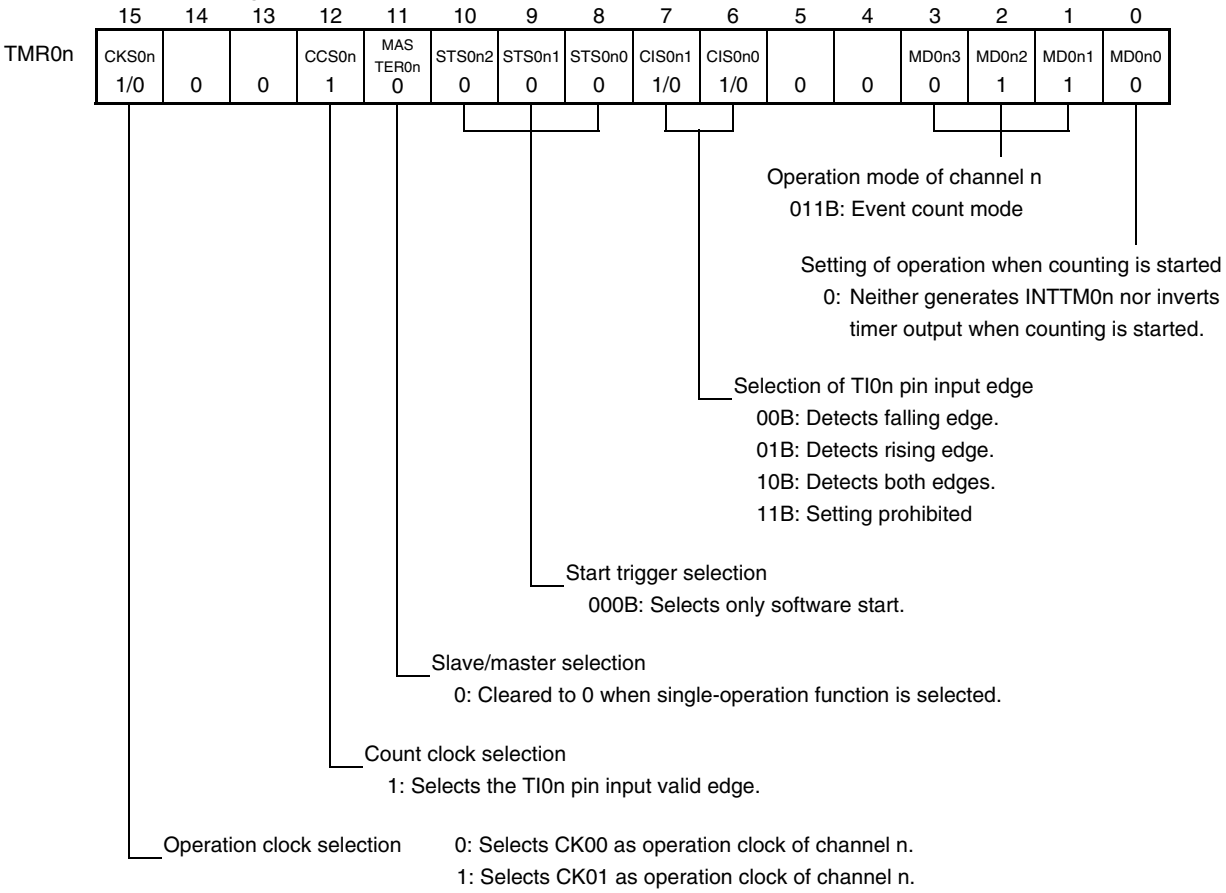
Figure 6-40. Example of Basic Timing of Operation as External Event Counter



Remark n = 0-4

Figure 6-41. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register 0n (TMR0n)



(b) Timer output register m (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register m (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register m (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when TOM0n = 0 (toggle mode).
	0	

(e) Timer output mode register m (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets toggle mode.
	0	

Remark n=0-4

Figure 6-42. 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 TMR0n register (determines operation mode of channel). Sets number of counts to the TDR0n register. Clears the TOE0n bit of the TOE0 register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Value of TDR0n is loaded to TCR0n and detection of the TI0n pin input edge is awaited.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of TMR0n, TOM0n, TOL0n, TO0n, and TOE0n registers cannot be changed.	Counter (TCR0n) counts down each time input edge of the TI0n pin has been detected. When count value reaches 0000H, the value of TDR0n is loaded to TCR0n again, and the count operation is continued. By detecting TCR0n = 0000H, the INTTM0n output is generated. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 0-4

6.7.3 Operation as frequency divider (only channel 0 of 78K0R/KE3-L)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from TO00.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR00 + 1)

TCR00 operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) is set to 1, TCR00 loads the value of TDR00 when the TI0n valid edge is detected. If MD0n0 of TMR0n = 0 at this time, INTTM0n is not output and TO0n is not toggled. If MD000 of TMR00 = 1, INTTM00 is output and TO00 is toggled.

After that, TCR00 counts down at the valid edge of TI00. When TCR00 = 0000H, it toggles TO00. At the same time, TCR0n loads the value of TDR00 again, and continues counting.

If detection of both the edges of TI00 is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

TDR00 can be rewritten at any time. The new value of TDR00 becomes valid during the next count period.

Figure 6-43. Block Diagram of Operation as Frequency Divider

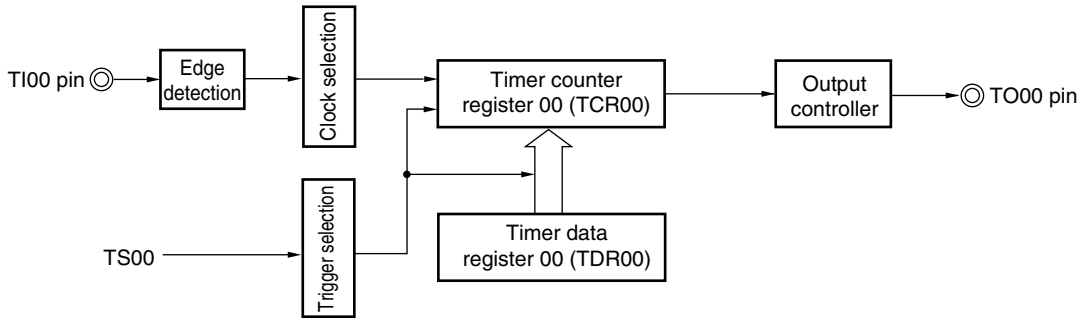


Figure 6-44. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)

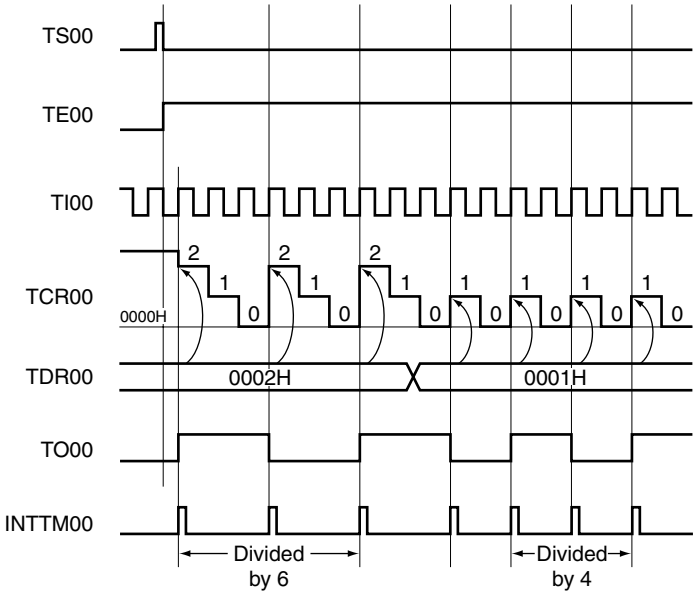
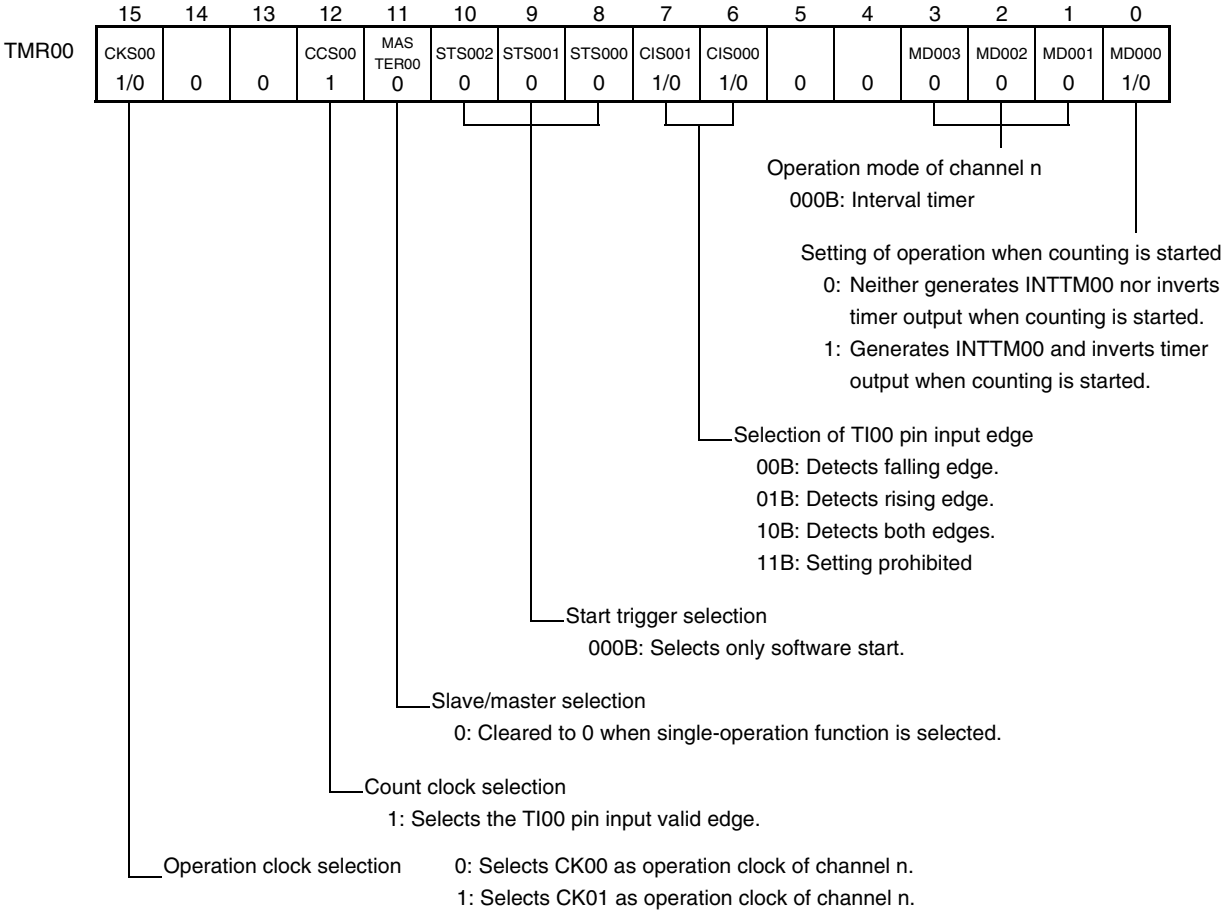


Figure 6-45. Example of Set Contents of Registers When Frequency Divider Is Used

(a) Timer mode register 00 (TMR00)



(b) Timer output register m (TO0)

	Bit 0	
TO0	TO00 1/0	0: Outputs 0 from TO00. 1: Outputs 1 from TO00.

(c) Timer output enable register m (TOE0)

	Bit 0	
TOE0	TOE00 1/0	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.

(d) Timer output level register m (TOL0)

	Bit 0	
TOL0	TOL00 0	0: Cleared to 0 when TOM0n = 0 (toggle mode)

(e) Timer output mode register m (TOM0)

	Bit 0	
TOM0	TOM00 0	0: Sets toggle mode.

Figure 6-46. 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 TMR00 register (determines operation mode of channel). Sets interval (period) value to the TDR00register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of the TOM0 register to 0 (toggle mode). Clears the TOL00 bit to 0. Sets the TO00 bit and determines default level of the TO00 output. →	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE00 to 1 and enables operation of TO00. →	TO00 does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 to 1 (only when operation is resumed). Sets the TS00 bit to 1. → The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of TDR00 is loaded to TCR00 at the count clock input. INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
During operation	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of TO0 and TOE0 registers can be changed. Set values of TMR00, TOM00, and TOL00 registers cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of TDR00 is loaded to TCR00 again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT00 bit is set to 1. → The TT00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 0, and count operation stops. TCR00 holds count value and stops. The TO00 output is not initialized but holds current status.
	TOE00 is cleared to 0 and value is set to the TO00 register. →	The TO00 pin outputs the TO0n set level.
TAU stop	To hold the TO00 pin output level Clears TO00 bit to 0 after the value to be held is set to the port register. →	The TO00 pin output level is held by port function.
	When holding the TO00 pin output level is not necessary Switches the port mode register to input mode. →	The TO00 pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed. → Operation start

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TI0n valid edge and the interval of the pulse input to TI0n can be measured.

The pulse interval can be calculated by the following expression.

TI0n input pulse interval = Period of count clock × ((**10000H** × **TSR0n: OVF**) + (Capture value of **TDR0n** + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of the TMR0n register, so an error equal to the number of operating clocks occurs.

TCR0n operates as an up counter in the capture mode.

When the channel start trigger (TS0n) is set to 1, TCR0n counts up from 0000H in synchronization with the count clock.

When the TI0n pin input valid edge is detected, the count value is transferred (captured) to TDR0n and, at the same time, the counter (TCR0n) is cleared to 0000H, and the INTTM0n is output. If the counter overflows at this time, the OVF bit of the TSR0n register is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, the OVF bit is configured as a cumulative flag; the correct interval value cannot be measured if an overflow occurs more than once.

Set STS0n2 -STS0n0 of the TMR0n register to 001B to use the valid edges of TI0n as a start trigger and a capture trigger.

When TE0n = 1, instead of the TI0n pin input, a software operation (TS0n = 1) can be used as a capture trigger.

Figure 6-47. Block Diagram of Operation as Input Pulse Interval Measurement

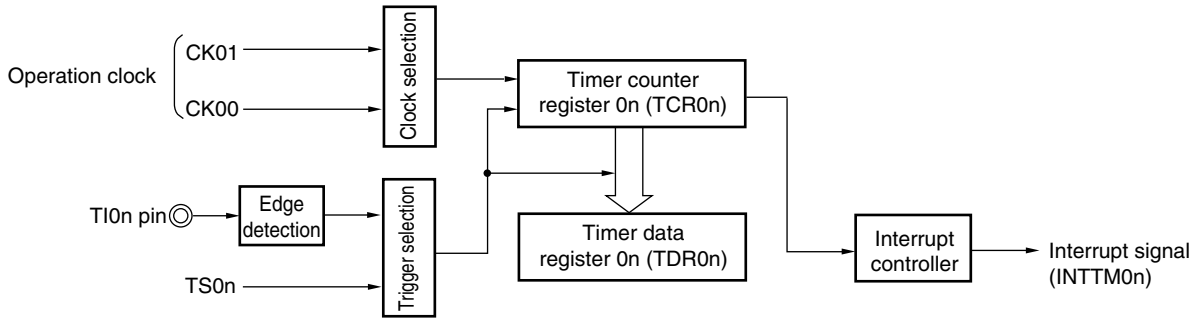
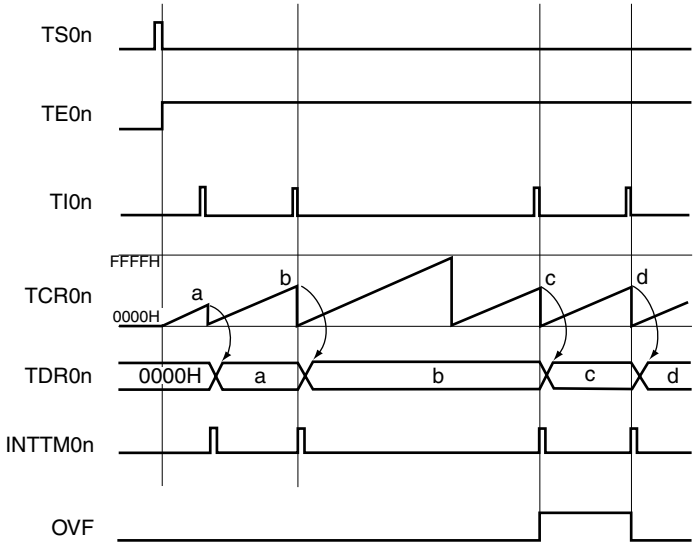


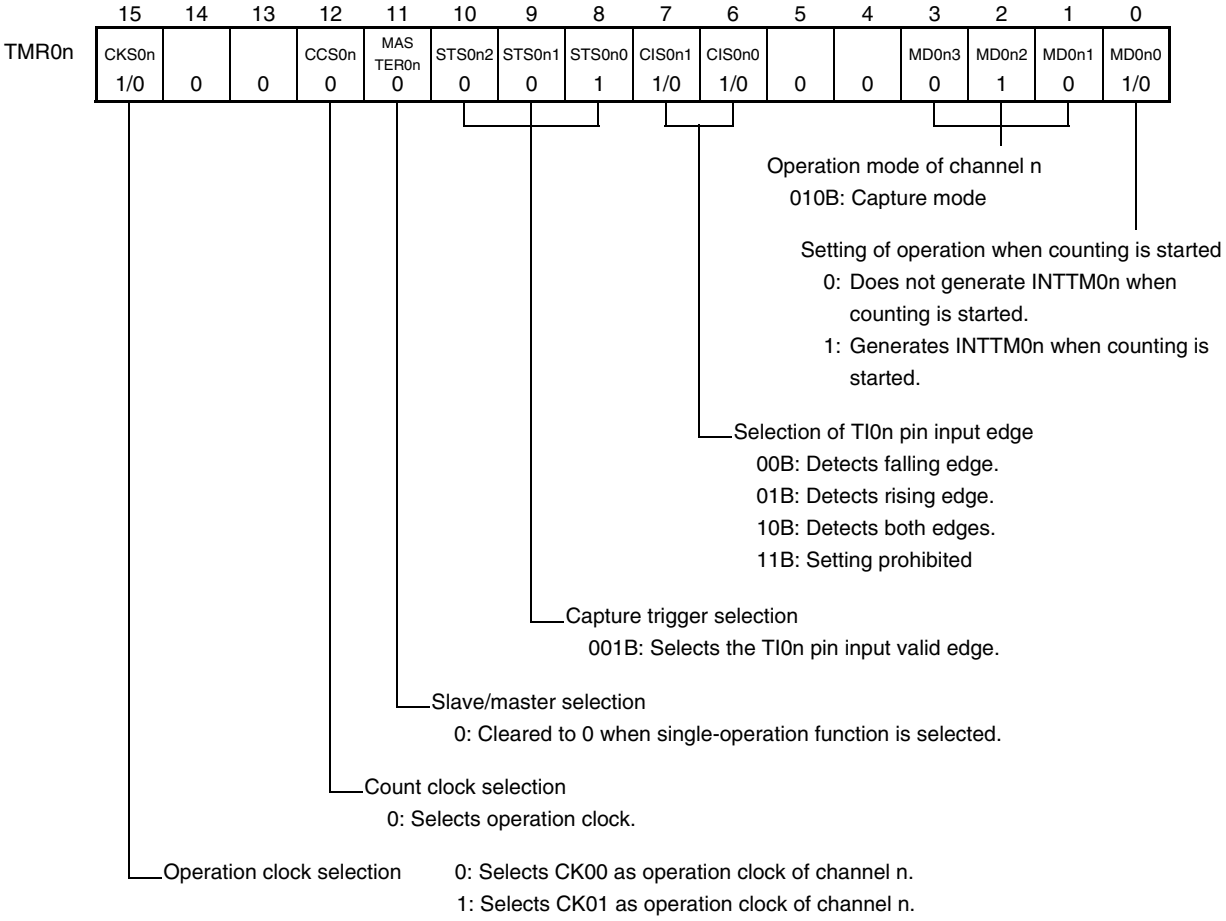
Figure 6-48. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)



Remark n = 0 - 4

Figure 6-49. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register 0n (TMR0n)



(b) Timer output register m (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register m (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops TO0n output operation by counting operation.
	0	

(d) Timer output level register m (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when TOM0n = 0 (toggle mode).
	0	

(e) Timer output mode register m (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets toggle mode.
	0	

Remark n = 0 - 4

Figure 6-50. 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 TMR0n register (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. TCR0n is cleared to 0000H at the count clock input. When the MD0n0 bit of the TMR0n register is 1, INTTM0n is generated.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of TOM0n, TOL0n, TO0n, and TOE0n registers cannot be changed.	Counter (TCRn) counts up from 0000H. When the TI0n pin input valid edge is detected, the count value is transferred (captured) to TDR0n. At the same time, TCR0n is cleared to 0000H, and the INTTM0n signal is generated. If an overflow occurs at this time, the OVF bit of the TSR0n register is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 0 - 4

6.7.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of TI0n and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0n can be measured. The signal width of TI0n can be calculated by the following expression.

Signal width of **TI0n** input = Period of count clock × ((**10000H** × **TSRn: OVF**) + (Capture value of **TDR0n** + **1**))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of the TMR0n register, so an error equal equivalent to one operation clock occurs.

TCR0n operates as an up counter in the capture & one-count mode.

When the channel start trigger (TS0n) is set to 1, TE0n is set to 1 and the TI0n pin start edge detection wait status is set.

When the TI0n start valid edge (rising edge of TI0n 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 TI0n when the high-level width is to be measured) is detected later, the count value is transferred to TDR0n and, at the same time, INTTM0n is output. If the counter overflows at this time, the OVF bit of the TSR0n register is set to 1. If the counter does not overflow, the OVF bit is cleared. TCR0n stops at the value “value transferred to TDR0n + 1”, and the TI0n 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 TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, the OVF 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 TI0n pin is to be measured can be selected by using the CIS0n1 and CIS0n0 bits of the TMR0n register.

Because this function is used to measure the signal width of the TI0n pin input, TS0n cannot be set to 1 while TE0n is 1.

CIS0n1, **CIS0n0** of **TMR0n** = **10B**: Low-level width is measured.
CIS0n1, CIS0n0 of **TMR0n** = **11B**: High-level width is measured.

Figure 6-51. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement

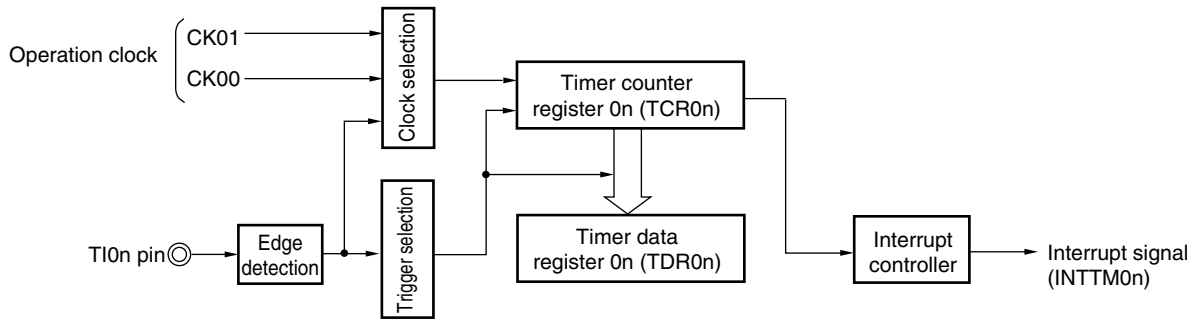
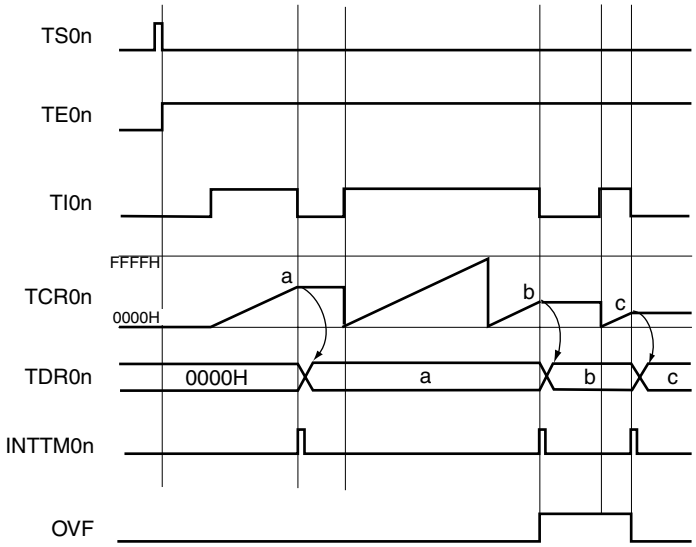


Figure 6-52. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark n = 0 - 4

Figure 6-53. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register 0n (TMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS0n			CCS0n	MASTER0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTM0n when counting is started.

Selection of TI0n pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TI0n pin input valid edge.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register m (TO0)

	Bit n
TO0	TO0n
	0

0: Outputs 0 from TO0n.

(c) Timer output enable register m (TOE0)

	Bit n
TOE0	TOE0n
	0

0: Stops the TO0n output operation by counting operation.

(d) Timer output level register m (TOL0)

	Bit n
TOL0	TOL0n
	0

0: Cleared to 0 when TOM0n = 0 (toggle mode).

(e) Timer output mode register m (TOM0)

	Bit n
TOM0	TOM0n
	0

0: Sets toggle mode.

Remark n = 0 - 4

Figure 6-54. 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 TMR0n register (determines operation mode of channel). Clears TOE0n to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and the TI0n pin start edge detection wait status is set.
	Detects TI0n pin input count start valid edge. →	Clears TCR0n to 0000H and starts counting up.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of TMR0n, TOM0n, TOL0n, TO0n, and TOE0n registers cannot be changed.	When the TI0n pin start edge is detected, the counter (TCR0n) counts up from 0000H. If a capture edge of the TI0n pin is detected, the count value is transferred to TDR0n and INTTM0n is generated. If an overflow occurs at this time, the OVF bit of the TSR0n register is set; if an overflow does not occur, the OVF bit is cleared. TCR0n stops the count operation until the next TI0n pin start edge is detected.
Operation stop	The TT0n bit is set to 1. → TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 0 - 4

6.8 Operation 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 **TDR0m** (slave)}/{Set value of **TDR0n** (master) + **1**} × **100**
0% output: Set value of **TDR0m** (slave) = **0000H**
100% output: Set value of **TDR0m** (slave) ≥ {Set value of **TDR0n** (master) + **1**}

Remark The duty factor exceeds 100% if the set value of TDR0m (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.

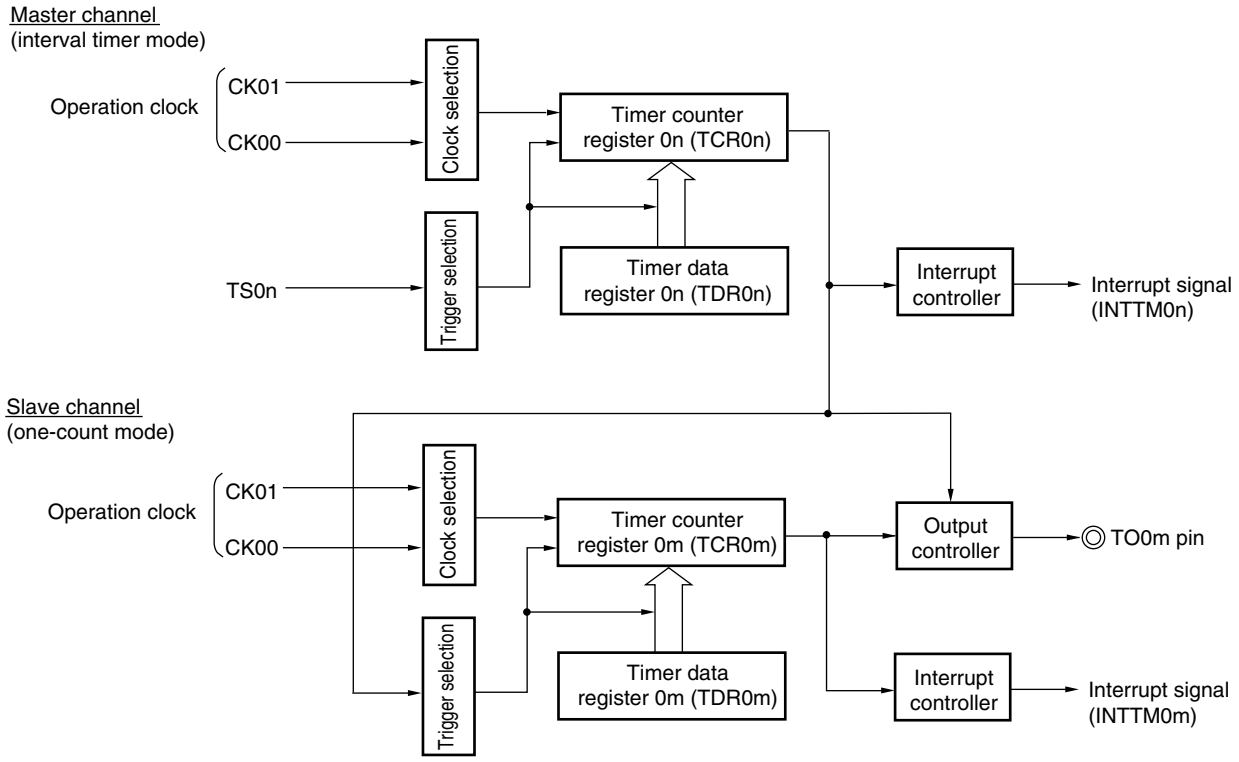
TCR0m of a slave channel operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0m pin. TCR0m of the slave channel loads the value of TDR0m, using INTTM0n of the master channel as a start trigger, and TCR0m counts down from the value of TDR0m. When TCR0m = 0000H, TCR0m stops counting until the next start trigger (INTTM0n of the master channel) is input.

The output level of TO0m becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0m = 0000H.

Caution To rewrite both TDR0n of the master channel and TDR0m of the slave channel, a write access is necessary two times. The timing at which the values of TDR0n and TDR0m are loaded to TCR0n and TCR0m 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 TO0m pin cannot output the expected waveform. To rewrite both TDR0n of the master and TDR0m of the slave, therefore, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel.

Remark n = 0, 2
m = n+1

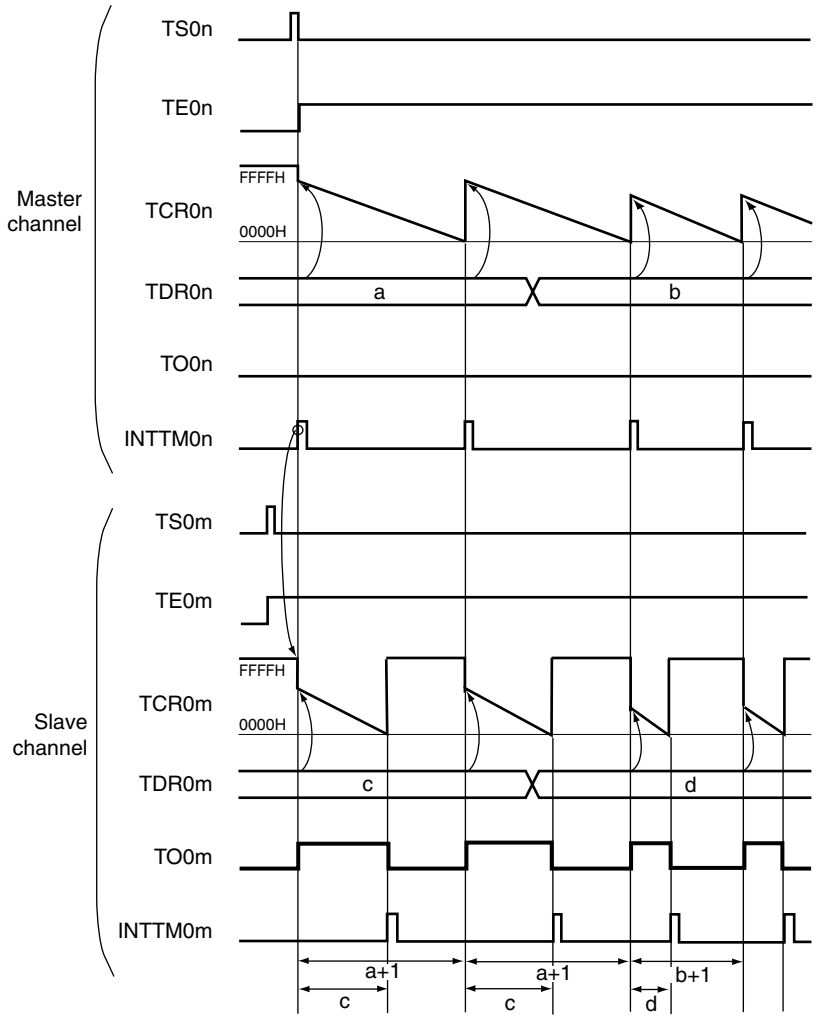
Figure 6-55. Block Diagram of Operation as PWM Function



Remark n = 0, 2

m= n+1

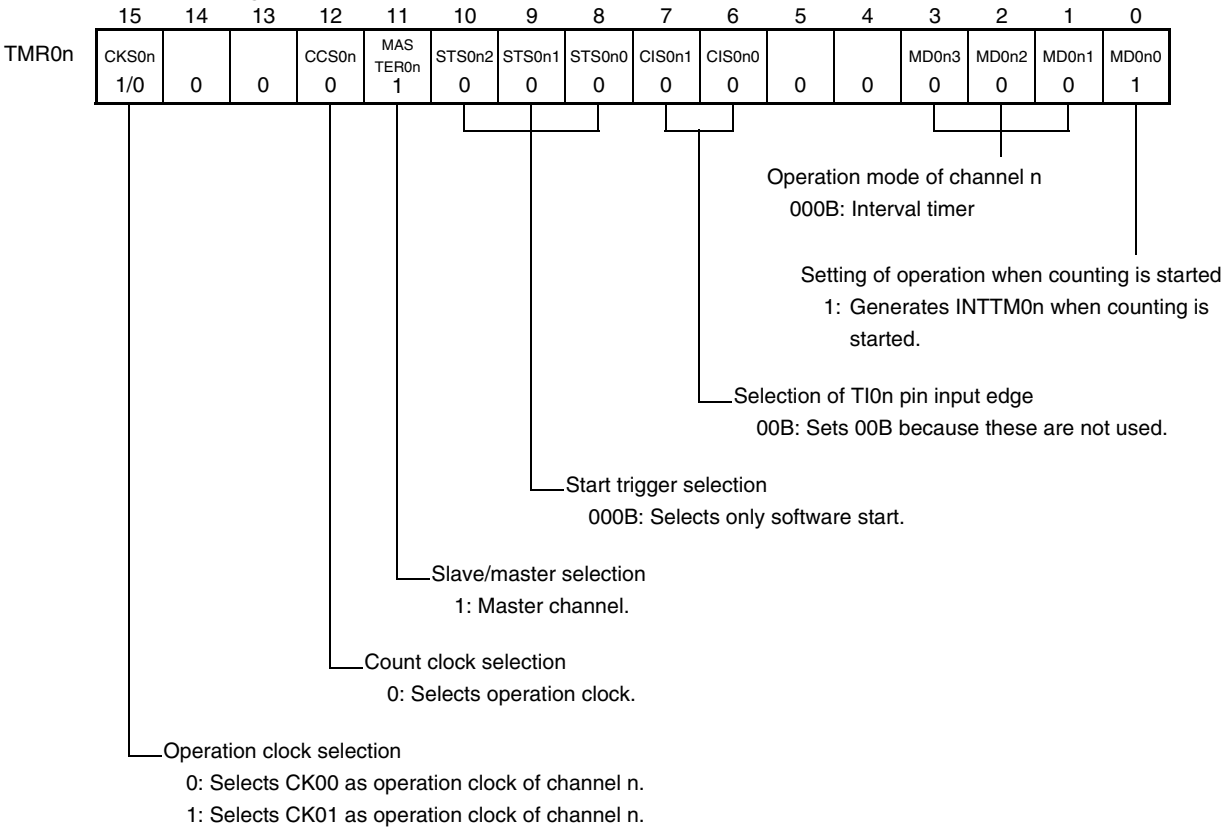
Figure 6-56. Example of Basic Timing of Operation as PWM Function



Remark n = 0, 2
m = n+1

Figure 6-57. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register m (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register m (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register m (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (toggle mode).

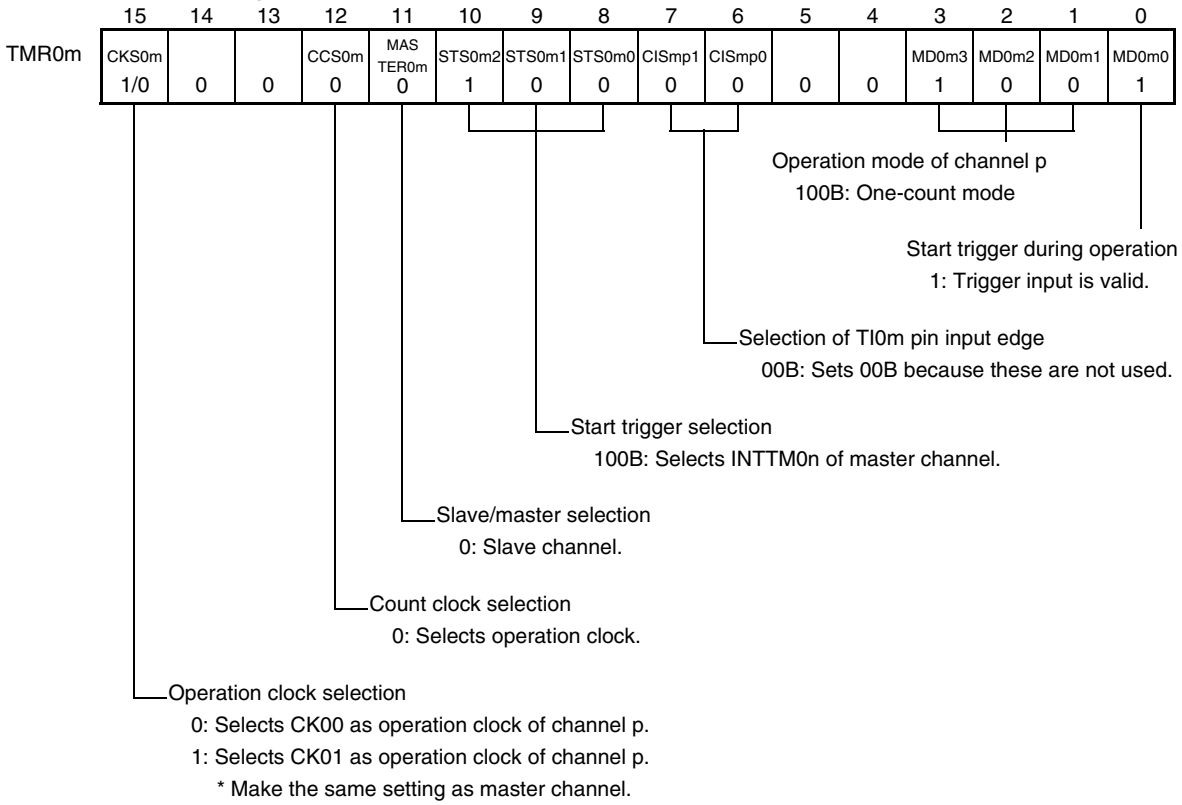
(e) Timer output mode register m (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets toggle mode.

Remark n = 0, 2

Figure 6-58. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMR0m)



(b) Timer output register m (TO0)

	Bit m	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register m (TOE0)

	Bit m	
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 m (TOL0)

	Bit m	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register m (TOM0)

	Bit m	
TOM0	TOM0p 1	1: Sets the combination-operation mode.

Remark n = 0, 2
m = n+1

Figure 6-59. 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 TMR0m 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 TDR0m register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0m bit of the TOM0 register is set to 1 m(combination-operation mode). Sets the TOL0m bit. Sets the TO0m bit and determines default level of the TO0m output. →	The TO0m pin goes into Hi-Z output state. The TO0m default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0m to 1 and enables operation of TO0m. →	TO0m does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0m pin outputs the TO0m set level.
Operation start	Sets TOE0m (slave) to 1 (only when operation is resumed). The TS0m (master) and TS0m (slave) bits of the TS0 register are set to 1 at the same time. → The TS0n and TS0m bits automatically return to 0 because they are trigger bits.	TE0n = 1, TE0m = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n and TMR0m registers, TOM0n, TOM0p, TOL0n, TOL0m bits cannot be changed. Set values of the TDR0n and TDR0m registers can be changed after INTTM0n of the master channel is generated. The TCR0n and TCR0m registers can always be read. The TSR0n and TSR0m 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 TDR0m is loaded to TCR0m, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0m becomes active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0m = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0m (slave) bits are set to 1 at the same time. → The TT0n and TT0m bits automatically return to 0 because they are trigger bits.	TE0n, TE0m = 0, and count operation stops. TCR0n and TCR0m hold count value and stops. The TO0m output is not initialized but holds current status.
	TOE0m of slave channel is cleared to 0 and value is set to the TO0m bit. →	The TO0m pin outputs the TO0m set level.

Operation is resumed.

Remark n=0, 2
m =n+1

Figure 6-59. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TO0m pin output levels Clears TO0m bit to 0 after the value to be held is set to the port register. →	The TO0m pin output levels is held by port function.
	When holding the TO0p pin output levels is not necessary Switches the port mode register to input mode. →	The TO0m pin output levels go are into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0m bit is cleared to 0 and the TO0m pin is set to port mode.)

Remark n=0, 2
m =n+1

6.8.2 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the **TI0n** pin.

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 **TDR0m** (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.

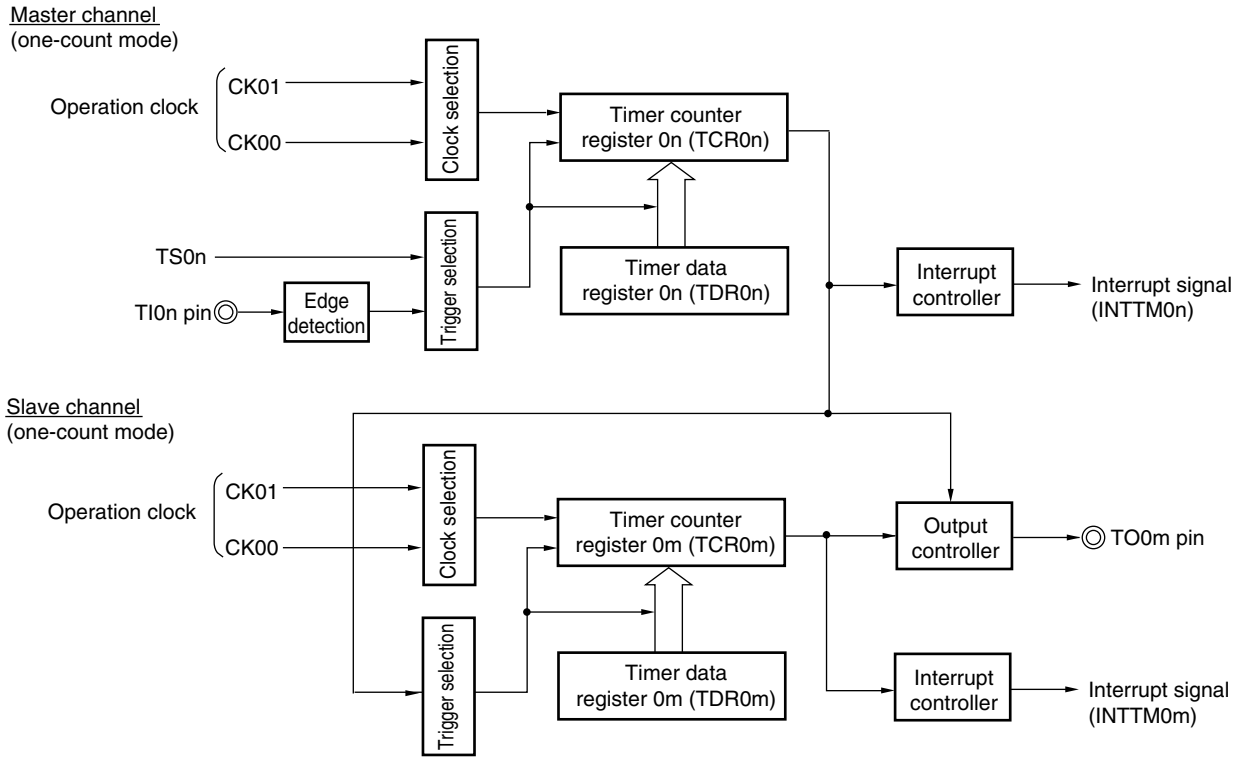
The slave channel operates in the one-count mode and counts the pulse width. TCR0m of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the TDR0m value. TCR0m counts down from the value of TDR0m it has loaded, in synchronization with the count value. When TCR0m = 0000H, it outputs INTTM0m and stops counting until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0m becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0m = 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 TDR0m of the slave channel. If TDR0n and TDR0m are rewritten during operation, therefore, an illegal waveform is output. Be sure to rewrite TDR0n and TDR0m after INTTM0n of the channel to be rewritten is generated.

Remark n=0, 2
m =n+1

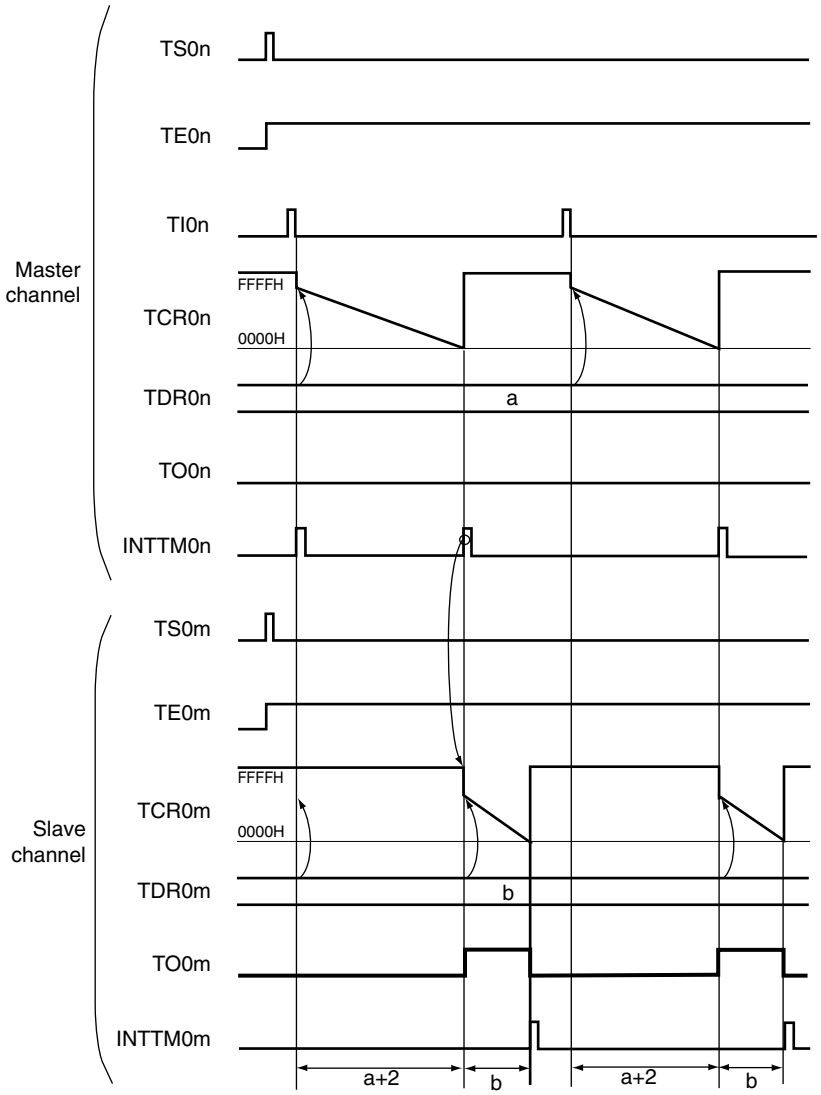
Figure 6-60. Block Diagram of Operation as One-Shot Pulse Output Function



Remark n=0, 2

m =n+1

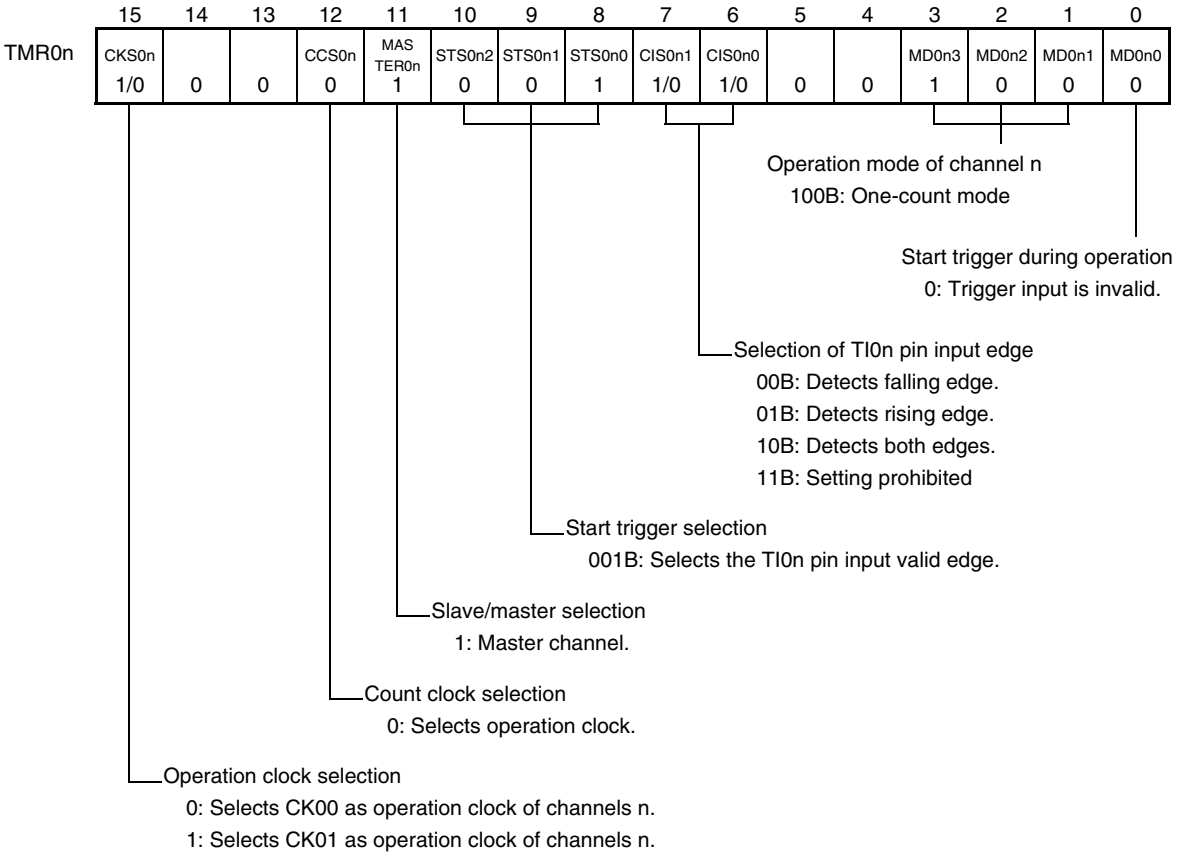
Figure 6-61. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remark n=0, 2
m =n+1

Figure 6-62. 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 m (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register m (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register m (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when TOM0n = 0 (toggle mode).
	0	

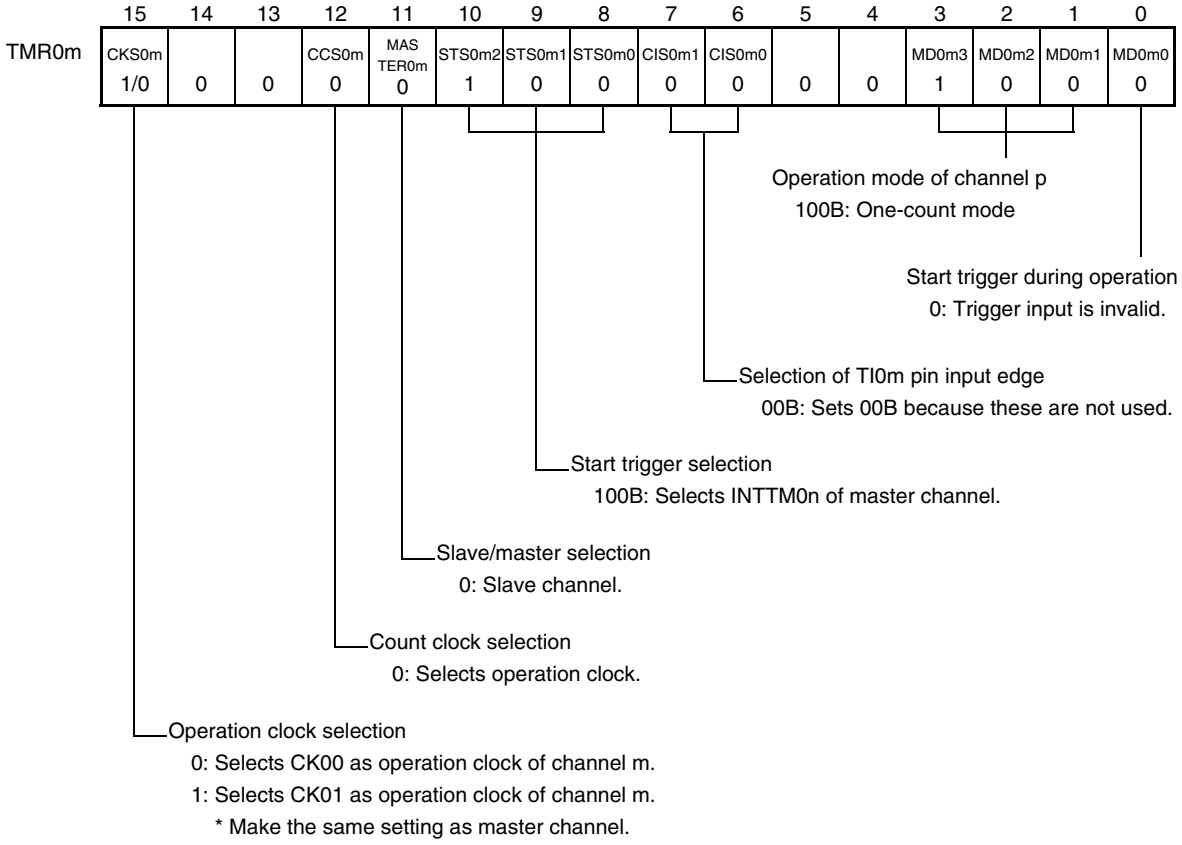
(e) Timer output mode register m (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets toggle mode.
	0	

Remark n=0, 2

Figure 6-63. Example of Set Contents of Registers when One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register 0m (TMR0m)



(b) Timer output register m (TO0)

	Bit m	
TO0	TO0m 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register m (TOE0)

	Bit m	
TOE0	TOE0m 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 m (TOL0)

	Bit m	
TOL0	TOL0m 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register m (TOM0)

	Bit m	
TOM0	TOM0m 1	1: Sets the combination-operation mode.

Remark n=0, 2
m =n+1

Figure 6-64. 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 TMR0m 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 TDR0m register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0m bit of the TOM0 register is set to 1 (combination-operation mode). Sets the TOL0m bit. Sets the TO0m bit and determines default level of the TO0m output. →	The TO0m pin goes into Hi-Z output state. The TO0m default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0m to 1 and enables operation of TO0m. →	TO0m does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0m pin outputs the TO0m set level.
Operation start	Sets TOE0m (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0m (slave) bits of the TS0 register are set to 1 at the same time. → The TS0n and TS0m bits automatically return to 0 because they are trigger bits.	TE0n =1,TE0m =1 , the master channel enters the TI0n input edge detection wait status. Counter stops operating.
	Detects the TI0n pin input valid edge of master channel. →	Master channel starts counting.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. Set values of the TMR0m, TDR0n, TDR0m registers, TOM0n, TOM0m, TOL0n, TOL0m bits cannot be changed. The TCR0n and TCR0m registers can always be read. The TSR0n and TSR0m 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 TI0n pin valid input edge 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 TDR0m to TCR0m, and the counter starts counting down. The output level of TO0m becomes active one count clock after generation of INTTM0n from the master channel. It becomes inactive when TCR0m = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0m (slave) bits are set to 1 at the same time. → The TT0n and TT0m bits automatically return to 0 because they are trigger bits.	TE0n, TE0m = 0, and count operation stops. TCR0n and TCR0m hold count value and stops. The TO0m output is not initialized but holds current status.
	TOE0m of slave channel is cleared to 0 and value is set to the TO0m bit. →	The TO0m pin outputs the TO0m set level.

Operation is resumed.

Remark n-0, 2
m=n+1

Figure 6-64. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TO0m pin output levels Clears TO0m bit to 0 after the value to be held is set to the port register. →	The TO0m pin output level is held by port function.
	When holding the TO0m pin output level is not necessary Switches the port mode register to input mode. →	The TO0m pin output levels go are into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0m bit is cleared to 0 and the TO0m pin is set to port mode.)

Remark n=0, 2
m=n+1

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 **TDR0q** (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 TCRmp, 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.

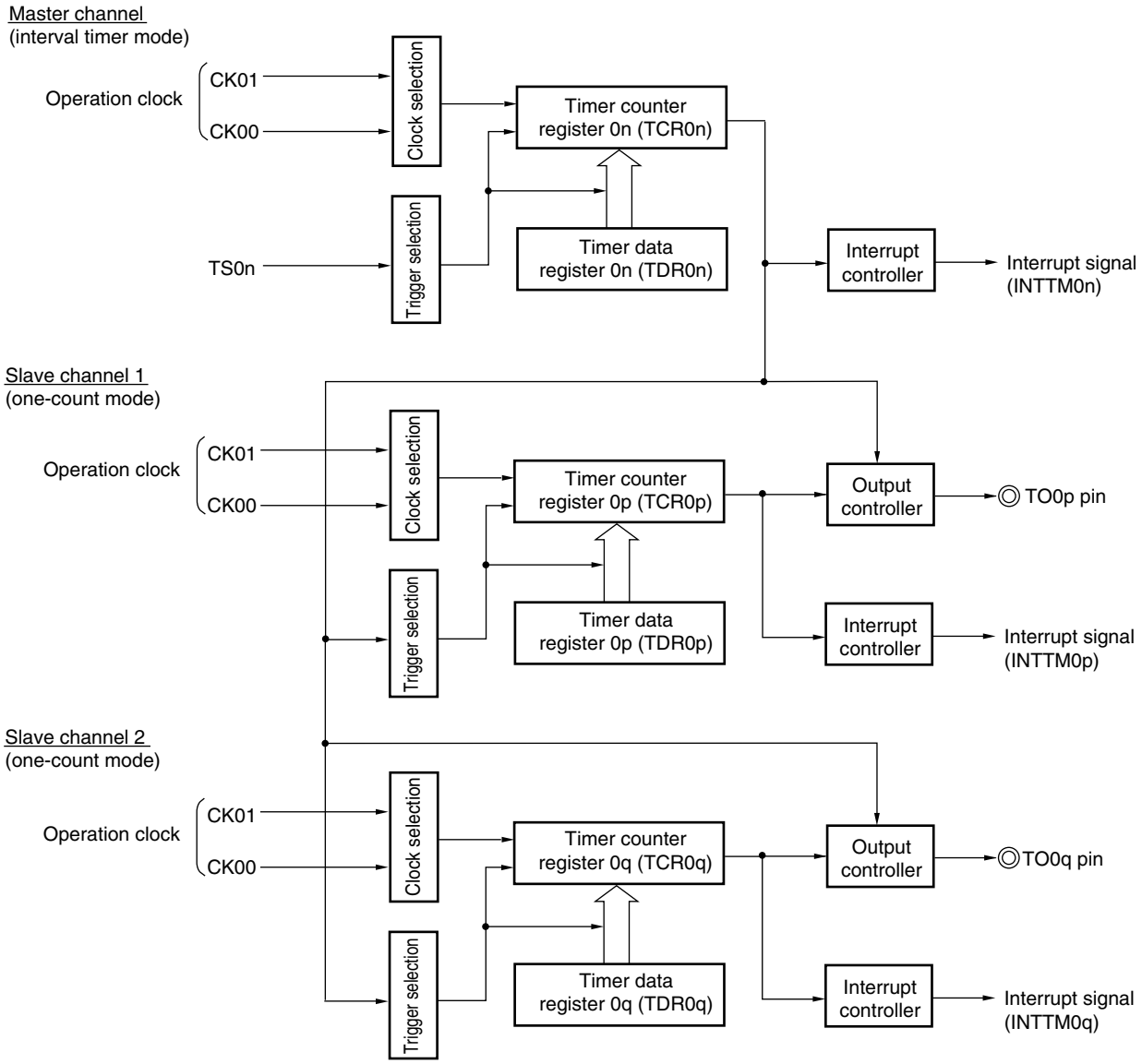
In the same way as TCR0p of the slave channel 1, TCRmq of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0q pin. TCRmq loads the value of TDRmq to TCR0q, using INTTM0n of the master channel as a start trigger, and starts counting down. When 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 above, as for the timer array unit 0, up to seven types of PWM, signals can be generated, while as for the timer array unit 1, up to three types of PWM signals can be generated.

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 TO00 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
n < p < q ≤ 4 (n < p < q ≤ 3 for 78K0R/KC3-L)
However p and q are consecutive integers greater than n) (p=n+1, q=n+2)

Figure 6-65. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)

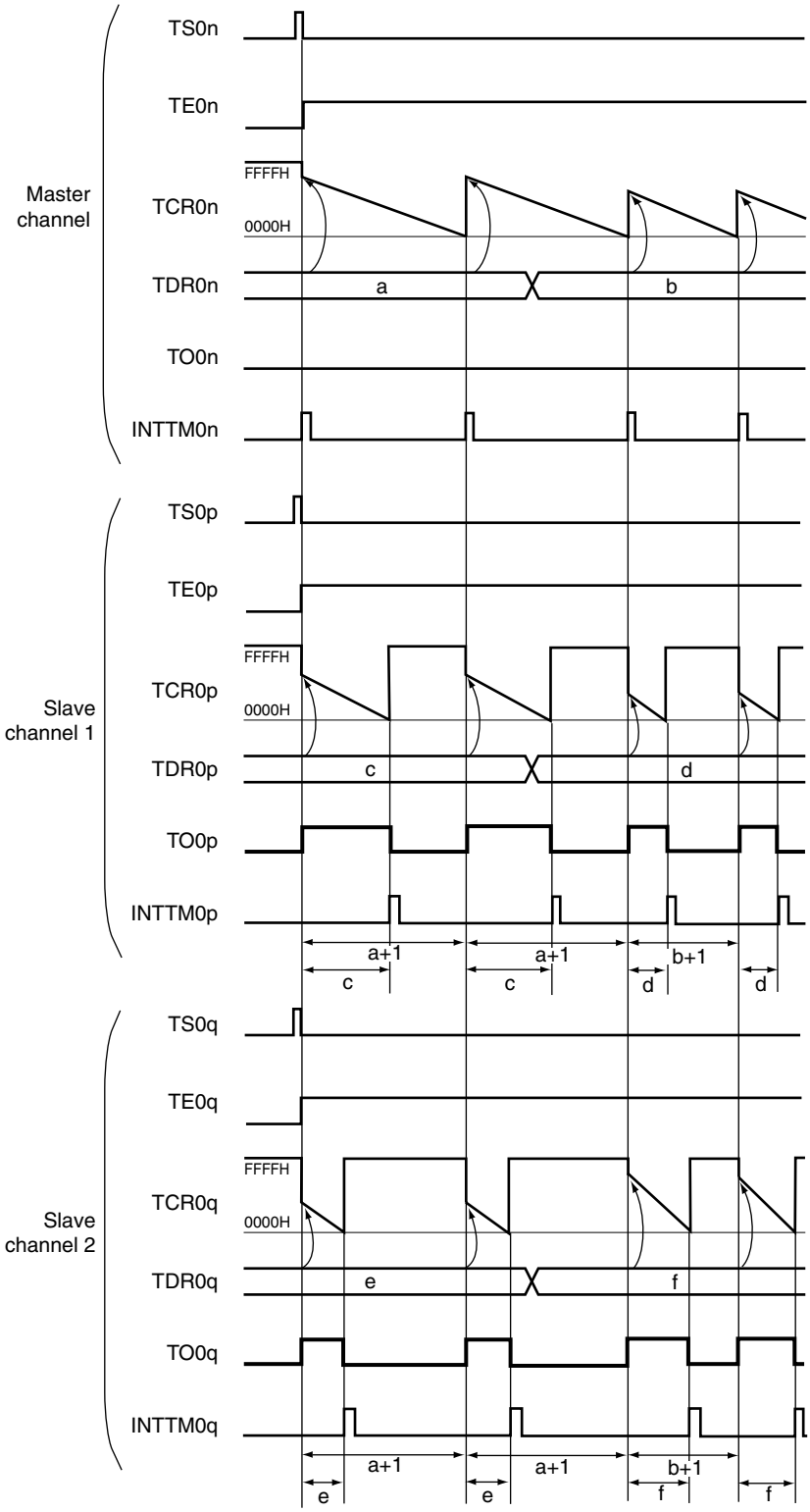


Remark 1. n=0, 2

2. p=n+1

q=n+2

Figure 6-66. Example of Basic Timing of Operation as Multiple PWM Output Function (output two types of PWMs)



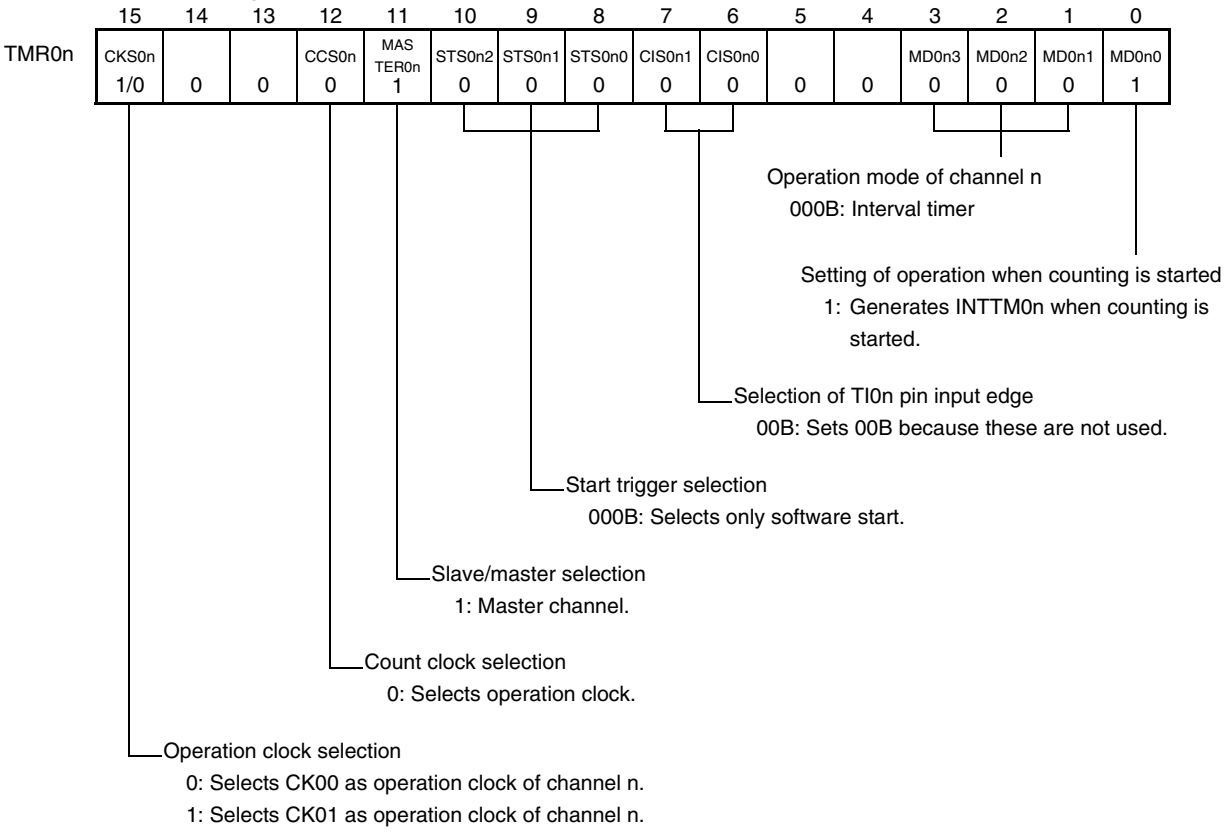
Remark 1. n=0, 2

2. p=n+1

q=n+2

Figure 6-67. 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 m (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register m (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register m (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (toggle mode).

(e) Timer output mode register m (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets toggle mode.

Remark n = 0, 2

Figure 6-68. 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	STS002	STS001	STS000	CIS0p1	CIS0p0			MD003	MD002	MD001	MD000
	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 channel 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: Slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel p, q.
1: Selects CK01 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TO0)

	Bit q	Bit p
TO0	TO0q	TO0m
	1/0	1/0

0: Outputs 0 from TO0p or TO0q.
1: Outputs 1 from TO0p or TO0q.

(c) Timer output enable register m (TOE0)

	Bit q	Bit p
TOE0	TOE0q	TOE0m
	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 m (TOL0)

	Bit q	Bit p
TOL0	TOL0q	TOL0m
	1/0	1/0

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

(e) Timer output mode register m (TOM0)

	Bit q	Bit p
TOM0	TOM0q	TOM0m
	1	1

1: Sets the combination-operation mode.

Remark n=0, 2; p=n+1; q=n+2

Figure 6-69. 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 registers 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 is 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 TO0p, TO0q pins go 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 and 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.

Operation is resumed (on the next page).

Remark 1. n = 0, 2

2. p = n+1; q = n+2

Figure 6-69. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOE0p and TOE0q (slave) to 1 (only when operation s resumed). The TS0n bit (master), and TS0p and TS0q (slave) bits of he TS0 register are set to 1 at the same time. The TS0n, TS0p, and TS0q bits automatically return to 0 because they are trigger bits.	TE0n = 1, TE0p, TE0q = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n, TMR0p, TMR0q registers, 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 TCR0q, 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. The TT0n, TT0p, and TT0q bits automatically return to 0 because they are trigger bits.	TE0n, TE0p and TE0q = 0, and count operation stops. TCR0n, TCR0p and TCR0q hold count value and stops. The TO0p and TO0q output is not initialized but holds current status.
	TOE0p or TOE0q of slave channel is cleared to 0 and value is set to the TO0p and TO0q bits.	The TO0p and TO0q pins output the TO0p and TO0q set levels.
TAU stop	To hold the TO0p and TO0q pin 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 pin output levels are held by port function.
	When holding the TO0p and TO0q pin output level is not necessary Switches the port mode register to input mode.	The TO0p and TO0q pin output levels go into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p and TO0q bits are cleared to 0 and the TO0p and TO0q pins are set to port mode.)

Operation is resumed (on the before page).

Remark 1. n=0, 2

2. p=n+1; q=n+2

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 512 Hz or 16.384 kHz or 32.768 kHz

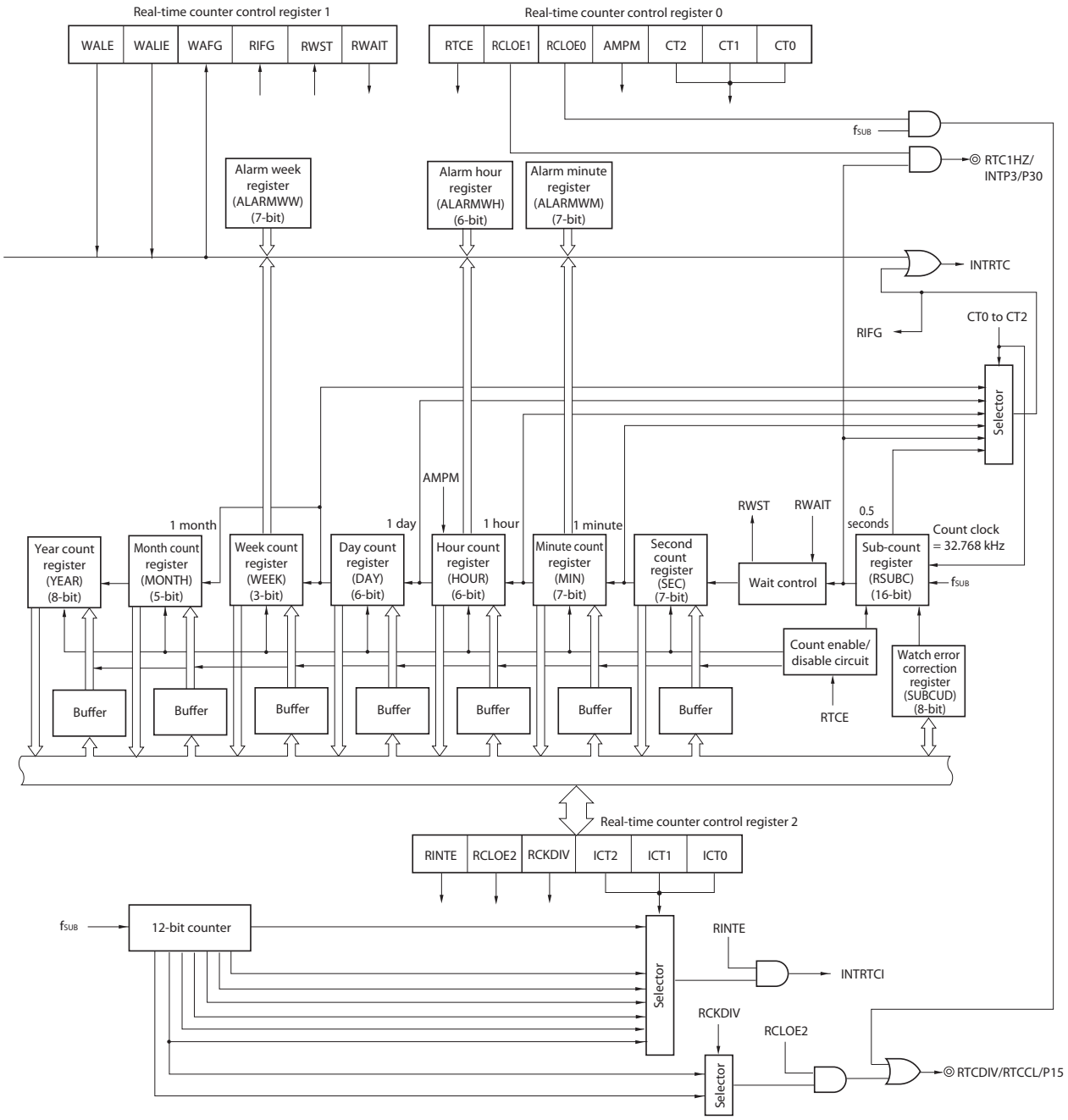
7.2 Configuration of Real-Time Counter

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)

Figure 7-1. Block Diagram of Real-Time Counter



7.3 Registers Controlling Real-Time Counter

The real-time counter is controlled by the following 16 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)

(1) Peripheral enable register 0 (PER0)

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

When the real-time 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	0	ADCEN	IICAEN	SAU1EN	SAU0EN	0	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock supply[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	Enables input clock supply. • SFR used by the real-time counter (RTC) can be read/written.

Note RTCEN is used to supply or stop the clock used when accessing the real-time counter (RTC) register 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 other than the real-time counter can be stopped in HALT mode when the subsystem clock is used, by setting RTCLPC of the operation speed mode control register (OSMC) to 1. In that case, set RTCEN to 1 and bits 0 to 6 of PER0 to 0.

3. Be sure to clear bit 1 and 6 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 pin, 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	0	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

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
• Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of real-time counter control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system. • Table 7-2 shows the displayed time digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of 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 is changed when RTCE = 1, the last waveform of the 32.768 kHz 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, the RSUBC register is cleared.	

Caution The RIFG and WAFG flags may be cleared when the RTCC1 register is written by using a 1-bit manipulation instruction. Use, therefore, an 8-bit manipulation instruction in order to write to the RTCC1 register. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting “1” to the corresponding bit. 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	$2^6/f_{XT}$ (1.953125 ms)
1	0	0	1	$2^7/f_{XT}$ (3.90625 ms)
1	0	1	0	$2^8/f_{XT}$ (7.8125 ms)
1	0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{XT}$ (125 ms)

RCLOE2[Note]	RTCDIV pin output control
0	Output of RTCDIV pin is disabled.
1	Output of RTCDIV pin is enabled.

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_{XT} 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.

(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 affected 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. If a value outside the range is set, the register value returns to the normal value after 1 period.

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. If a value outside the range is set, the register value returns to the normal value after 1 period.

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 value of 00 to 23, or 01 to 12 and 21 to 32, in decimal format by using a BCD code, according to the time system set by using bit 3 (AMPM) of real-time counter control register 0 (RTCC0). If the value of the AMPM bit is changed, the HOUR value will be automatically changed according to the specified time system. If a value outside the range is set, the register value returns to the normal value after 1 period.

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 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 shows the relationship between the setting value of the AMPM bit, the HOUR register value, and time.

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. If a value outside the range is set, the register value returns to the normal value after 1 period.
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. If a value outside the range is set, the register value returns to the normal value after 1 period.
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 (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

(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. If a value outside the range is set, the register value returns to the normal value after 1 period.

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 count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (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. If a value outside the range is set, the register value returns to the normal value after 1 period.

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 (SEC).

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 Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(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 is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 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

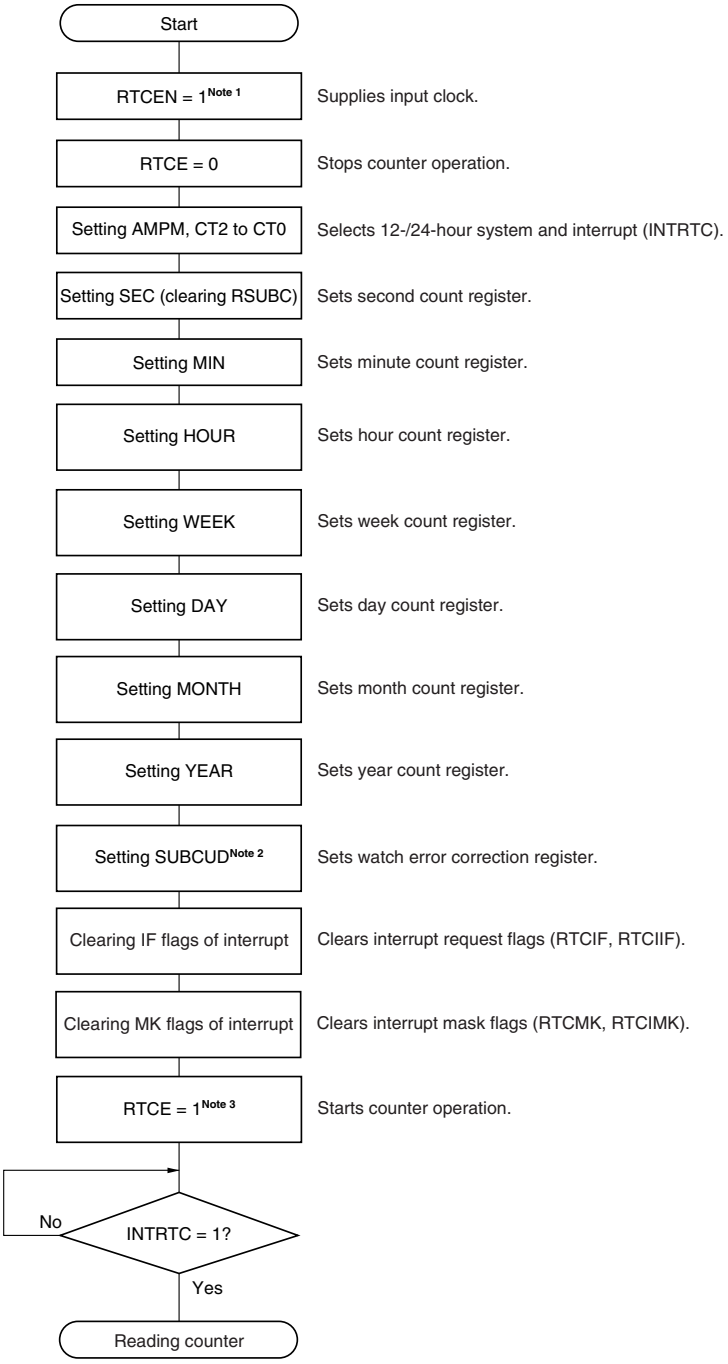
Here is an example of setting the alarm.

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

7.4 Real-Time Counter Operation

7.4.1 Starting operation of real-time counter

Figure 7-18. Procedure for Starting Operation of Real-Time Counter



<R>

Notes 1. First set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

<R> **2.** Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **7.4.7 Example of watch error correction of real-time counter**.

3. 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.

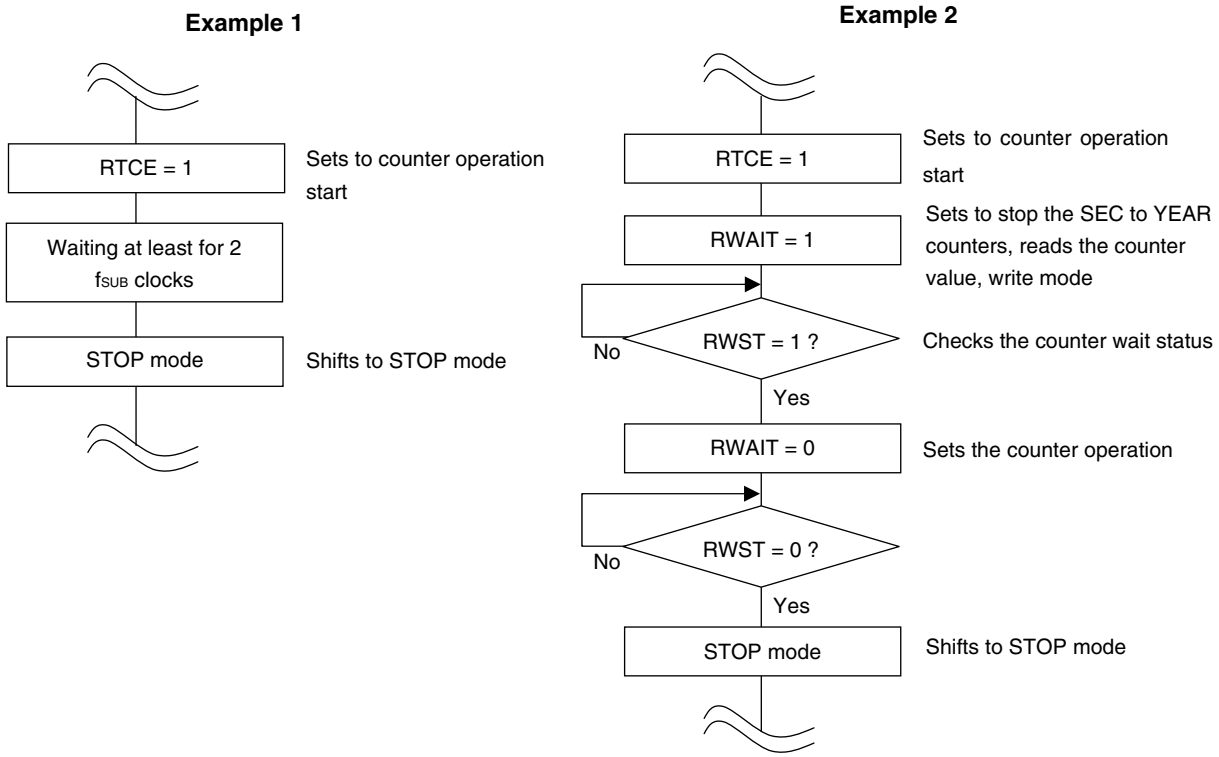
7.4.2 Shifting to STOP mode after starting operation

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 7-19**, **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 7-19**, **Example 2**).

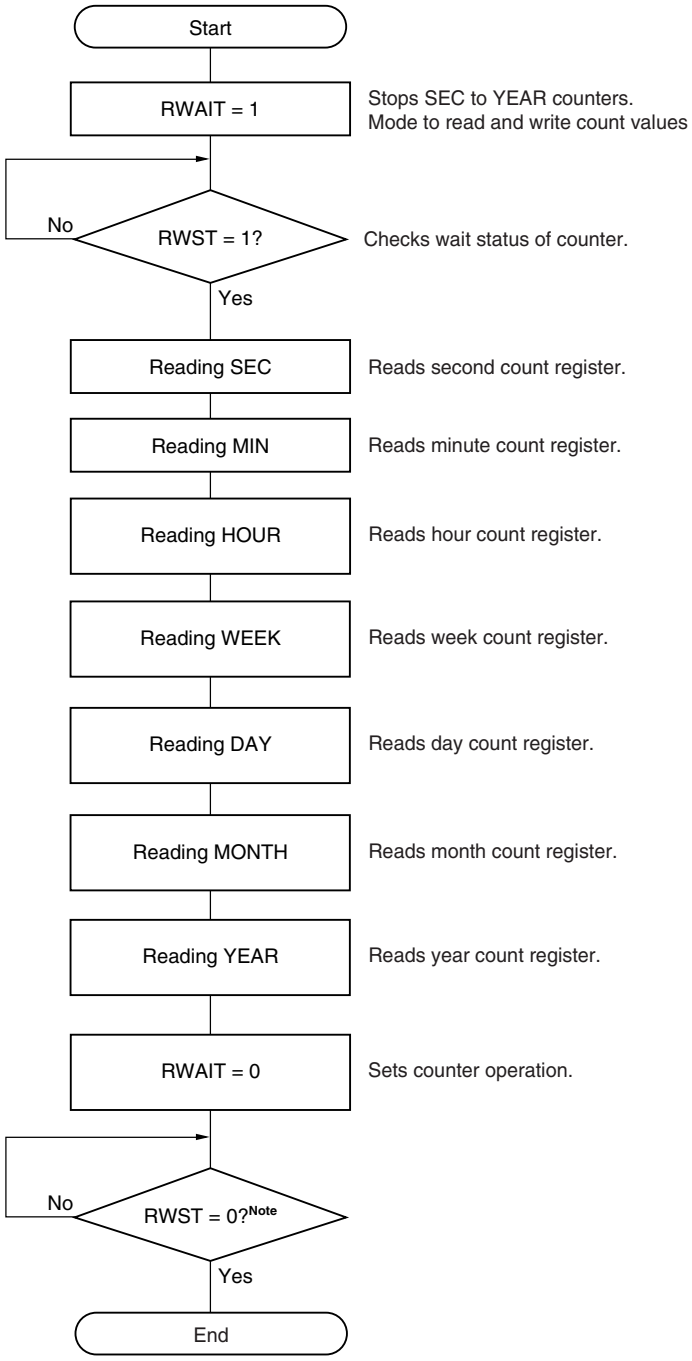
Figure 7-19. Procedure for Shifting to STOP Mode After Setting RTCE to 1



7.4.3 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 7-20. Procedure for Reading Real-Time Counter

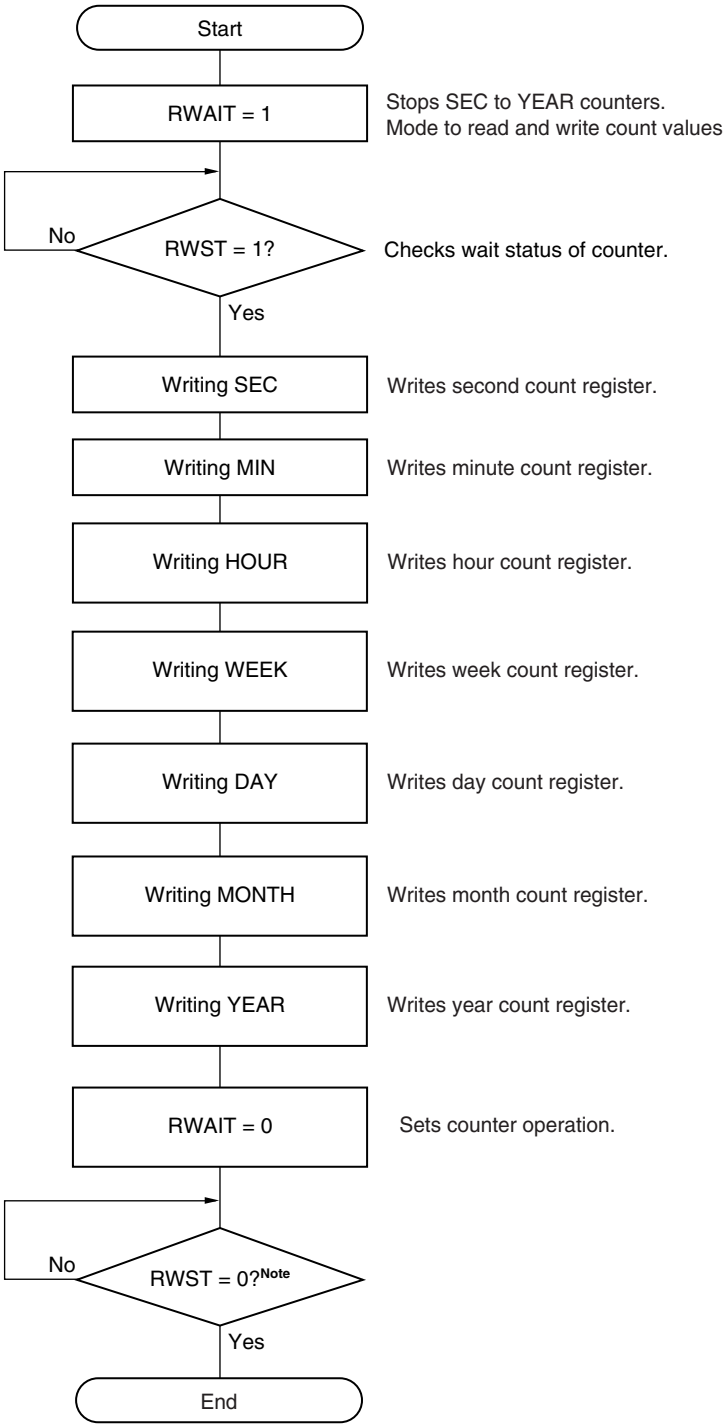


Note Be sure to confirm that RWST = 0 before setting STOP mode.

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 7-21. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

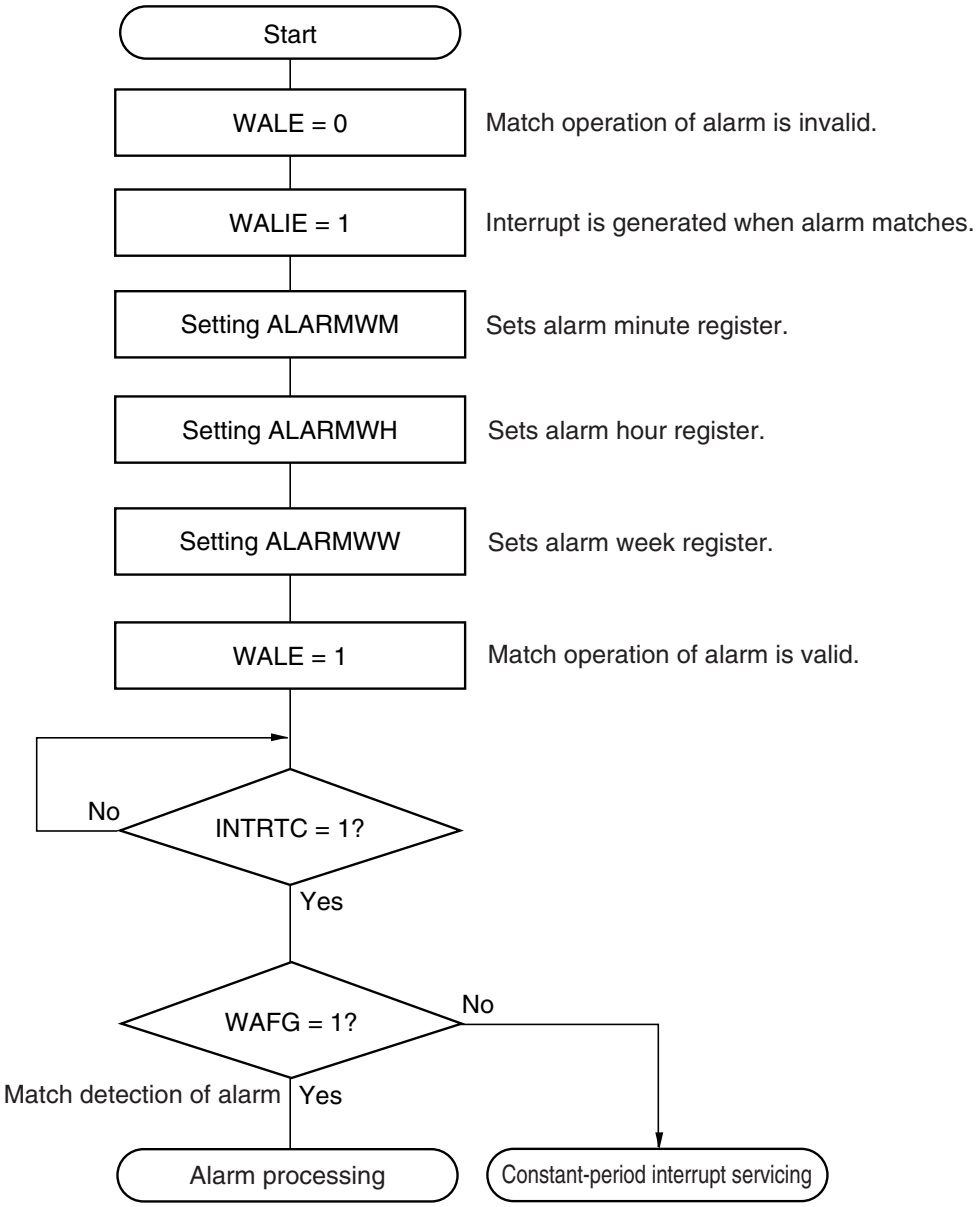
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

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 7-22. Alarm Setting Procedure

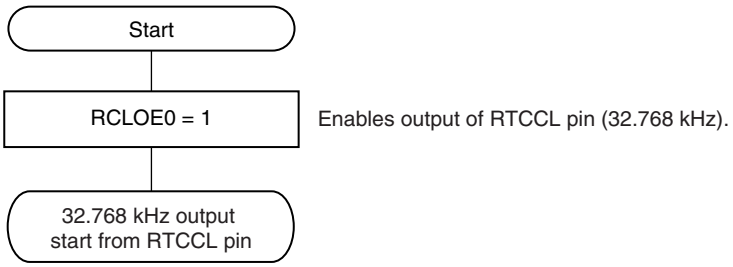


Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

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.

<R> 7.4.5 32.768 kHz output of real-time counter

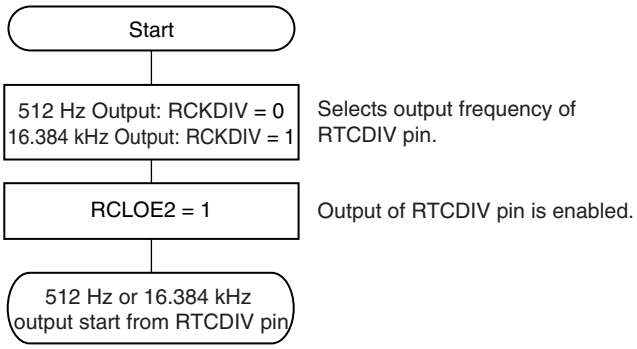
Figure 7-23. 32.768 kHz Output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

<R> 7.4.6 512 Hz, 16.384 kHz output of real-time counter

Figure 7-24. 512 Hz, 16.384 kHz output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

7.4.7 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 kHz output frequency of the RTCCL pin is set to its initial value.

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 32.768 kHz from the RTCCL pin is set to its initial value (00H).

Note See setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **7.4.5 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.

$$\begin{aligned}\text{Correction value} &= \text{Number of correction counts in 1 minute} \div 3\\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \div 3\\ &= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3\\ &= 86\end{aligned}$$

[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.

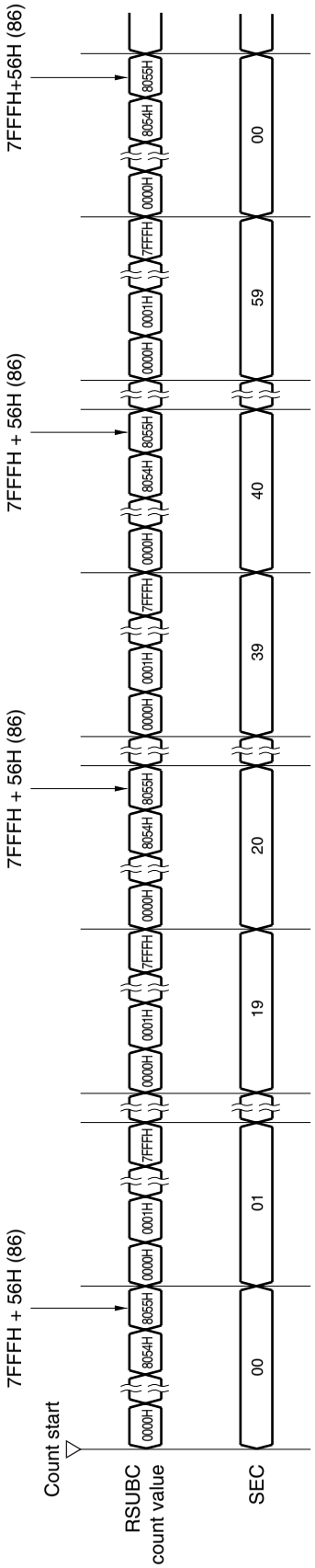
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86\\ (F5, F4, F3, F2, F1, F0) &= 44\\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)\end{aligned}$$

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz − 131.2 ppm), setting the correction register such that DEV = 0 and the correction value is 86 (bits 6 to 0 of SUBCUD: 0101100) results in 32768 Hz (0 ppm).

Figure 7-25 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 7-25. 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 32.768 kHz from the RTCCL pin is set to its initial value (00H).

Note See **7.4.5 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 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.

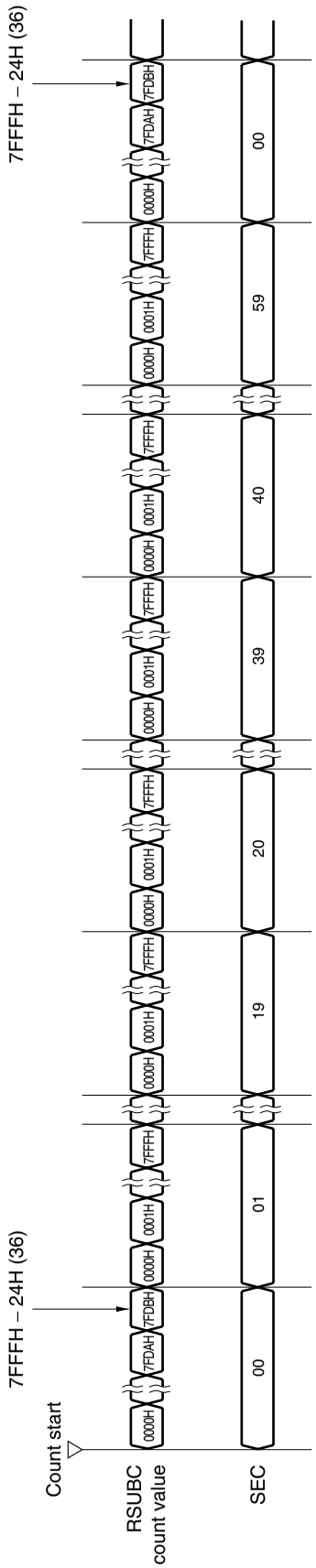
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

− {(/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 7-26 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 7-26. 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 19 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)

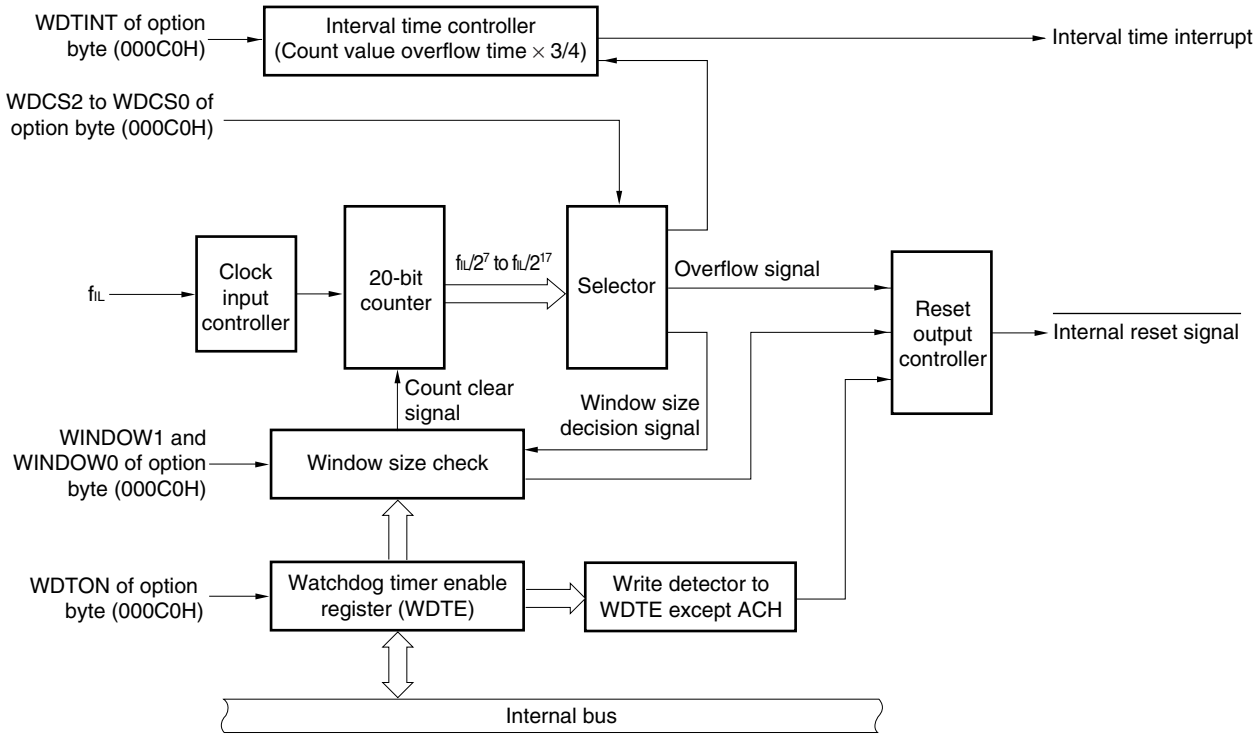
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

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 22 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 Operation 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 22**).

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 Setting overflow time of watchdog timer** and **CHAPTER 22**).
   - 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 Setting window open period of watchdog timer** and **CHAPTER 22**).

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.

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 times can be set.

<R>

Table 8-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 34.5 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.71 ms)
0	0	1	$2^8/f_{IL}$ (7.42 ms)
0	1	0	$2^9/f_{IL}$ (14.84 ms)
0	1	1	$2^{10}/f_{IL}$ (29.68 ms)
1	0	0	$2^{12}/f_{IL}$ (118.72 ms)
1	0	1	$2^{14}/f_{IL}$ (474.90 ms)
1	1	0	$2^{15}/f_{IL}$ (949.80 ms)
1	1	1	$2^{17}/f_{IL}$ (3799.19 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.

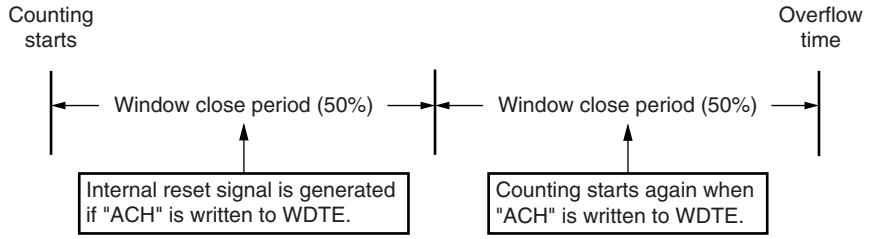
Remark f_{IL}: Internal low-speed oscillation clock frequency

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.

<R> **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 can be set is as follows.

Table 8-4. Setting Window Open Period of Watchdog Timer

<R>

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.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^{10}/f_{IL}$ (MAX.) = 2^{10}/34.5 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) = 0 to 2^{10}/25.5 kHz (MIN.) × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) to $2^{10}/f_{IL}$ (MAX.) = 2^{10}/25.5 kHz (MIN.) × 0.5 to 2^{10}/34.5 kHz (MAX.) = 20.08 to 29.68 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.

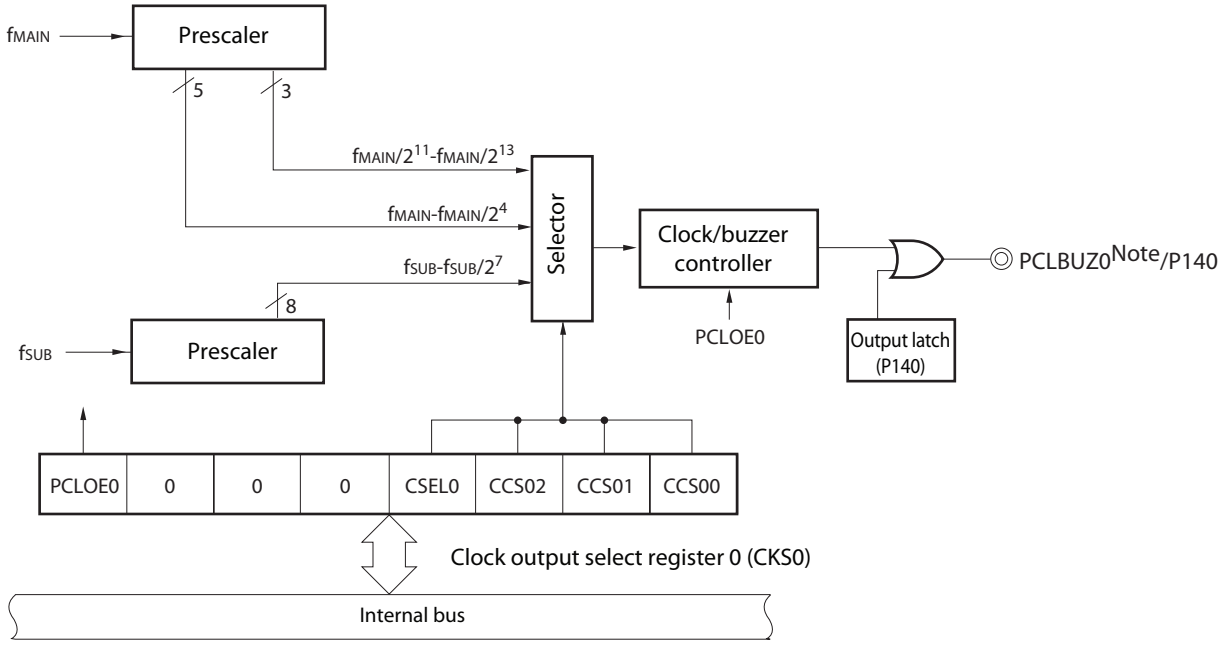
Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

PCLBUZ0 outputs a clock selected by clock output select register n (CKS0).

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 pin 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.

Remark f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

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 (CKS0) Port mode register 14 (PM14) Port register 14 (P14)

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 register 0 (CKS0)
- Port mode registers 14 (PM14)

(1) Clock output select register 0 (CKS0)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0), and set the output clock.

Select the clock to be output from PCLBUZ0 by using CKS0.

CKS0 is 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 (CKS0)

Address: FFFA5H (CKS0) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKS0	PCLOE0	0	0	0	CSEL0	CCS02	CCS01	CCS00

PCLOE0	PCLBUZ0 output enable/disable specification
0	Output disable (default)
1	Output enable

CSEL0	CCS02	CCS01	CCS00	PCLBUZ0 output clock selection			
					f_{MAIN} = 12 MHz	f_{MAIN} = 16 MHz	f_{MAIN} = 20 MHz
0	0	0	0	f_{MAIN}	12 MHz	16 MHz	Setting prohibited[Note]
0	0	0	1	f_{MAIN}	6 MHz	8 MHz	10 MHz[Note]
0	0	1	0	$f_{MAIN}/2$	3 MHz	4 MHz	5 MHz
0	0	1	1	$f_{MAIN}/2^2$	1.5 MHz	2 MHz	2.5 MHz
0	1	0	0	$f_{MAIN}/2^3$	0.75 MHz	1 MHz	1.25 MHz
0	1	0	1	$f_{MAIN}/2^4$	5.86 kHz	7.81 kHz	9.76 kHz
0	1	1	0	$f_{MAIN}/2^{11}$	2.93 kHz	3.91 kHz	4.88 kHz
0	1	1	1	$f_{MAIN}/2^{12}$	1.46 kHz	1.95 kHz	2.44 kHz
1	0	0	0	$f_{MAIN}/2^{13}$	32.768 kHz		
1	0	0	1	f_{SUB}	16.384 kHz		
1	0	1	0	$f_{SUB}/2$	8.192 kHz		
1	0	1	1	$f_{SUB}/2^2$	4.096 kHz		
1	1	0	0	$f_{SUB}/2^3$	2.048 kHz		
1	1	0	1	$f_{SUB}/2^4$	1.024 kHz		
1	1	1	0	$f_{SUB}/2^5$	512 Hz		
1	1	1	1	$f_{SUB}/2^6$	256 Hz		

Note Use the output clock within a range of 10 MHz.

Cautions 1. Change the output clock after disabling clock output (PCLOE0 = 0).

2. To shift to STOP mode when the main system clock is selected (CSEL0 = 0), set PCLOE0 = 0 before executing the STOP instruction. When the subsystem clock is selected (CSEL0 = 1), PCLOE0 = 1 can be set because the clock can be output in STOP mode.

Remarks 1. f_{MAIN}: Main system clock frequency

2. f_{SUB}: Subsystem clock frequency

(2) Port mode register 14 (PM14)

This register sets input/output of port 14 in 1-bit units.

When using the P140/INTP6/PCLBUZ0 pin for clock output/buzzer output, clear PM140 and the output latch of P140 to 0.

PM14 can be 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 14 (PM14)

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	PM144 note	PM143 note	PM142 note	1	PM140

PM14n	P14n pin I/O mode selection (n = 0,2-4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Only for 78K0R/KE3-L

9.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

PCLBUZ0 outputs a clock/buzzer selected by clock output select register 0 (CKS0).

9.4.1 Operation as output pin

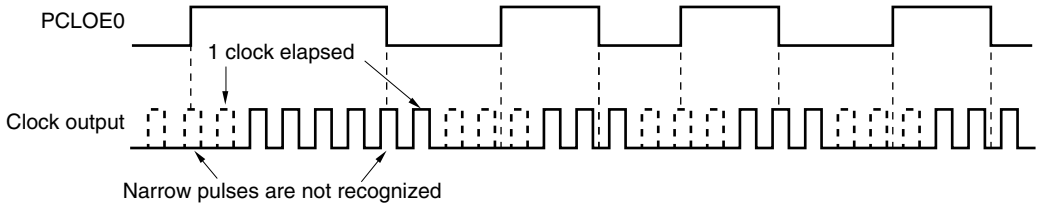
PCLBUZ0 is output as the following procedure.

<1> Select the output frequency with bits 0 - 3 (CCS00 - CCS02, CSEL0) of the clock output select register (CKS0) of the PCLBUZ0 pin (output in disabled status).

<2> Set bit 7 (PCLOE0) of CKS0 to 1 to enable clock/buzzer output.

Remarks The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOE0) is switched. At this time, pulses with a narrow width are not output. Figure 9-4 shows enabling or stopping output using PCLOE0 and the timing of outputting the clock.

Figure 9-4 Remote Control Output Application Example



CHAPTER 10 A/D CONVERTER

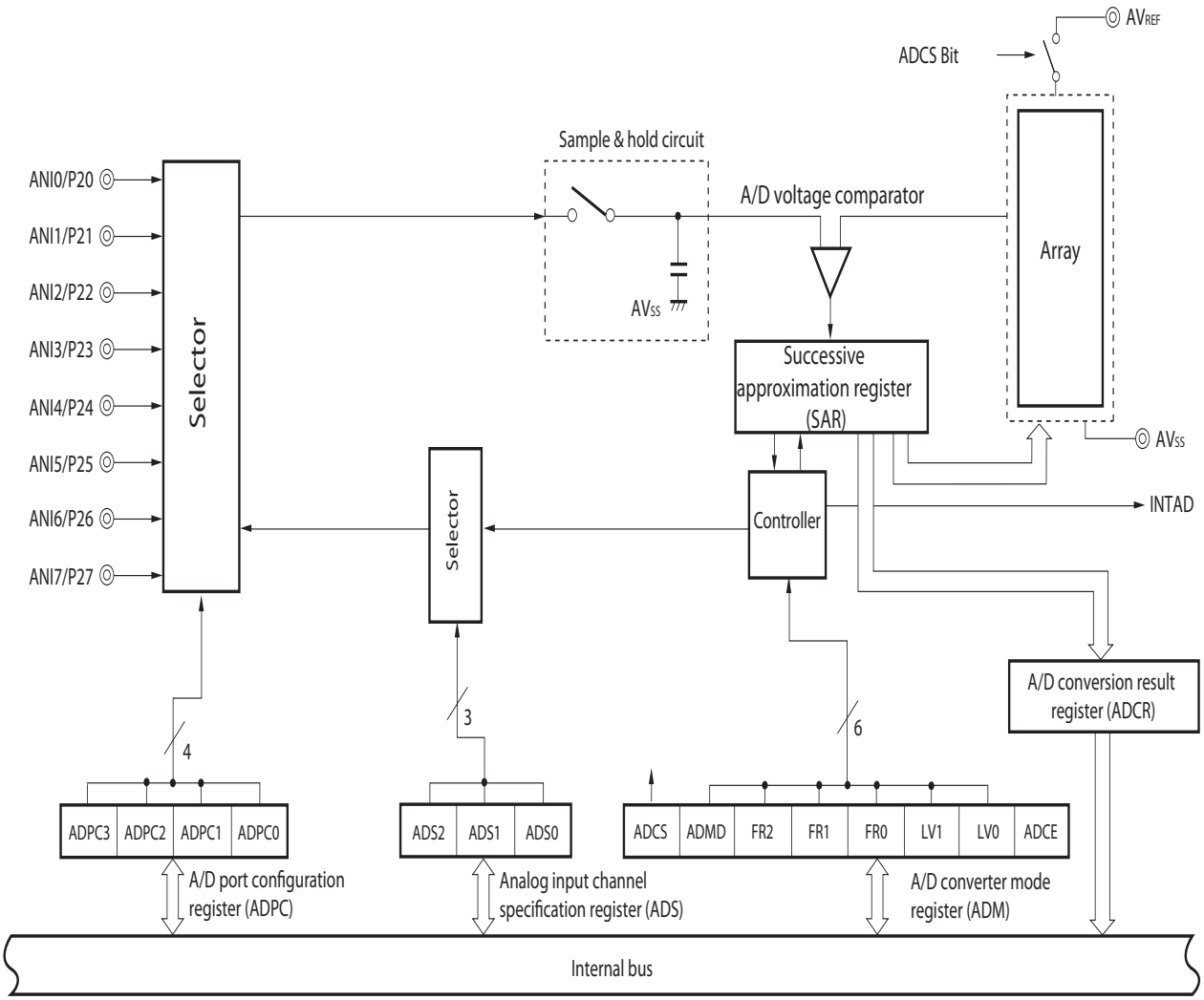
10.1 Function of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of up to 12 channels (ANI0 - ANI17) with a resolution of 10 bits.

The A/D converter has the following function.

- **10-bit resolution A/D conversion**
  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 - ANI11. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

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 - ANI17 pins

These are the analog input pins of the 8 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the array with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB of the SAR is reset.

After that, bit 10 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the array is selected by the value of bit 11, to which the result has been already set.

Bit 11 = 0: (1/4 AV_{REF})
Bit 11 = 1: (3/4 AV_{REF})

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 10 = 1
Analog input voltage ≤ Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR register.

(4) Array

The array generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a 12-bit register that sets voltage tap data whose values from the array match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

(9) AV_{REF} pin

This pin inputs the reference voltage of the A/D converter, the power supply pins and A/D converter of the comparator. When all pins of ports 2 are used as the analog port pins, make the potential of AV_{REF} be such that 1.8 V ≤ AV_{REF} ≤ V_{DD}. When one or more of the pins of ports 2 are used as the digital port pins, make AV_{REF} the same potential as V_{DD}. The analog signal input to ANI0 - ANI17 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(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.

10.3 Registers Used in A/D Converter

The A/D converter uses the following seven registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- Port mode registers 2 (PM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

(1) Peripheral enable register 0 (PER0)

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

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 10-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

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

Cautions 1. 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. Be sure to clear bit 1 and 6 to 0.

(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 10-3. 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	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADMD	A/D conversion operation mode specification
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation (A/D voltage comparator: 1/2AV_{REF} operation)

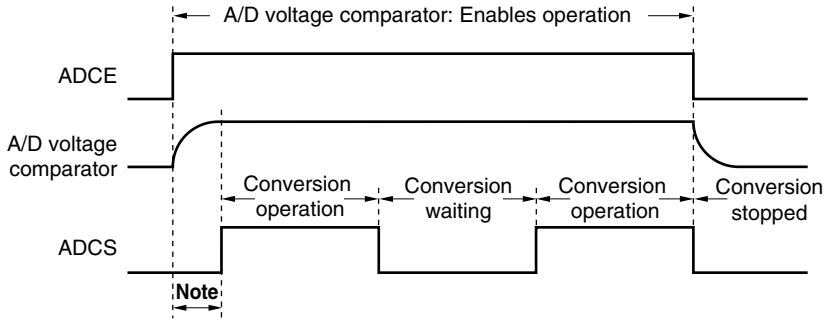
Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 10-2 A/D Conversion Time Selection**.

2. 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, by waiting for at least 1 μs to elapse before setting ADCS to 1 after ADCE has been set to 1, the conversion results are valid from the first 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 (only A/D voltage comparator consumes power))
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator: enables operation)

Figure 10-4. 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 rising of the ADCS bit must be 1 μs or longer.

Caution A/D conversion must be stopped before rewriting bits FR0 to FR2, LV1, and LV0 to values other than the identical data.

Table 10-2. A/D Conversion Time Selection

(1) 1.8 V ≤ AV_{REF} ≤ 3.6 V

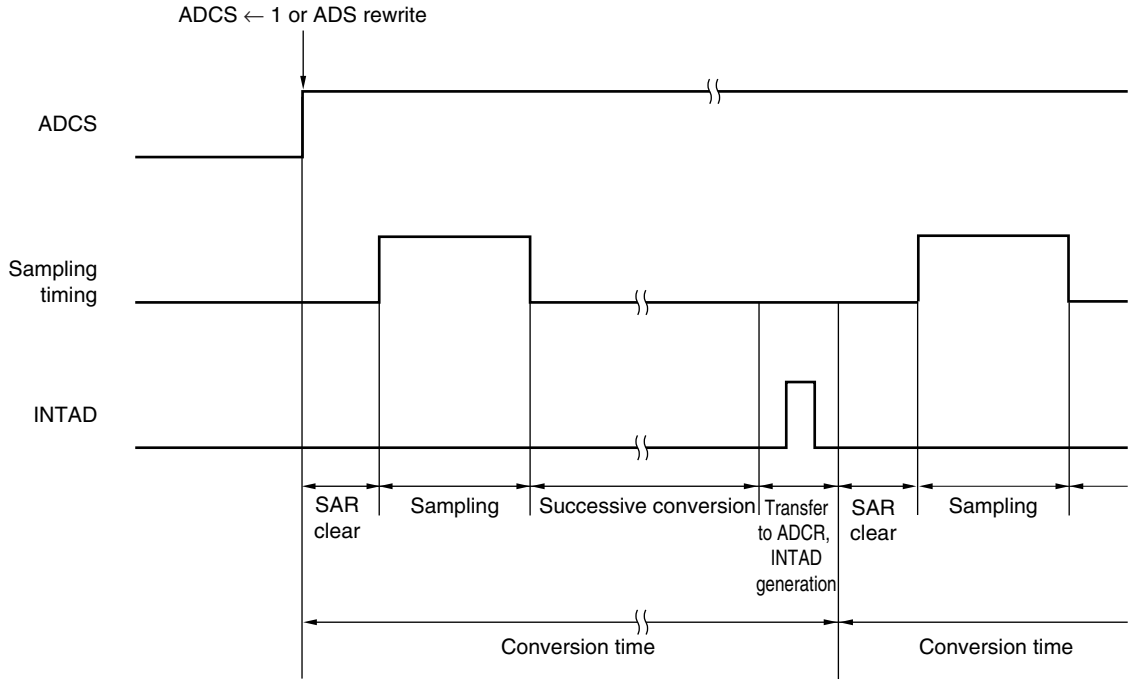
A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	Standard	34.2 μs	28.5 μs	21.4 μs	17.1 μs	f_{CLK}/20
0	0	1				17.2 μs	14.4 μs	10.8 μs	8.6 μs	f_{CLK}/10
0	1	0				13.8 μs	11.5 μs	8.6 μs	Setting prohibited	f_{CLK}/8
0	1	1				10.4 μs	8.7 μs	Setting prohibited		f_{CLK}/6
1	0	0				Setting prohibited	Setting prohibited			f_{CLK}/4
1	0	1								f_{CLK}/3
1	1	0								f_{CLK}/2
1	1	1								f_{CLK}
0	0	0	0	1	Low-voltage	34.2 μs	28.5 μs	Setting prohibited	Setting prohibited	f_{CLK}/20
0	0	1				Setting prohibited	Setting prohibited			f_{CLK}/10
0	1	0								f_{CLK}/8
0	1	1								f_{CLK}/6
1	0	0								f_{CLK}/4
1	0	1								f_{CLK}/3
1	1	0								f_{CLK}/2
1	1	1								f_{CLK}
×	×	×	1	0	–	Setting prohibited				–
0	0	0	1	1	High speed	34.2 μs	28.5 μs	21.4 μs	17.1 μs	f_{CLK}/20
0	0	1				17.2 μs	14.4 μs	10.8 μs	8.6 μs	f_{CLK}/10
0	1	0				13.8 μs	11.5 μs	8.6 μs	6.9 μs	f_{CLK}/8
0	1	1				10.4 μs	8.7 μs	6.5 μs	5.2 μs	f_{CLK}/6
1	0	0				7.0 μs	5.8 μs	4.4 μs	3.5 μs	f_{CLK}/4
1	0	1				5.3 μs	4.4 μs	Setting prohibited	Setting prohibited	f_{CLK}/3
1	1	0				3.6 μs	Setting prohibited			f_{CLK}/2
1	1	1				Setting prohibited				f_{CLK}

Cautions 1. When rewriting FR2 - FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

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

Figure 10-5. A/D Converter Sampling and A/D Conversion Timing



(3) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10-6. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	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.

(4) 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Figure 10-7. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to A/D converter mode register (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.

(5) 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 10-8. Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	ANI0	P20/ANI0 pin
0	0	1	ANI1	P21/ANI1 pin
0	1	0	ANI2	P22/ANI2 pin
0	1	1	ANI3	P23/ANI3 pin
1	0	0	ANI4	P24/ANI4 pin
1	0	1	ANI5	P25/ANI5 pin
1	1	0	ANI6	P26/ANI6 pin
1	1	1	ANI7	P27/ANI7 pin

Cautions 1. Be sure to clear bits 3 - 7 to "0".

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

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

Figure 10-8. Format of Analog Input Channel Specification Register (ADS) (2/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

○ Scan mode (ADMD = 1)

ADS2	ADS1	ADS0	Analog input channel			
			Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	1	ANI1	ANI2	ANI3	ANI4
0	1	0	ANI2	ANI3	ANI4	ANI5
0	1	1	ANI3	ANI4	ANI5	ANI6
1	0	0	ANI4	ANI5	ANI6	ANI7
Other than the above			Setting prohibited			

Cautions 1. Be sure to clear bits 3 - 7 to "0".

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

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

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

This register switches the ANI0/P20 - ANI7/P27 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 10-9. 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 2							
					ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	D	D	D	D	D	D
0	0	1	1	1	A	D	D	D	D	D	D	D
0	1	0	0	0	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited								

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

2. Do not set the pin that is set by ADPC as digital I/O by ADS.

<R> **3. Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.**

(7) Port mode registers 2 (PM2)

When using the ANI0/P20 - ANI7/P27 pins for analog input port set PM20 - PM27 to 1. The output latches of P20 - P27 and at this time may be 0 or 1.

If PM20 - PM27 are set to 0, they cannot be used as analog input port pins.

PM2 can be set by a 1-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 10-10. Formats of Port Mode Registers 2 (PM2)

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

PM2n	P2n pin I/O mode selection (n=7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The ANI0/P20 - ANI7/P27 pins are as shown below depending on the settings of ADPC, ADS, and PM2.

Table 10-3. Setting Functions of ANI0/P20 - ANI7/P27 Pins

ADPC	PM2	ADS	ANI0/P20 - ANI7/P27 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	Selects ANI.	Setting prohibited
		Does not select ANI.	

10.4 A/D Converter Operations

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 - 1 (FR2 - FR0, LV1, and LV0) of ADM, and set the operation mode by using bit 6 (ADMD) of ADM.

<3> Set bit 0 (ADCE) of A/D converter mode register (ADM) to 1 to start the operation of the A/D voltage comparator.

<4> 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).

<5> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<6> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.
A timer trigger wait state is entered if the timer trigger mode is set in step <7>.
(<7> to <13> are operations performed by hardware.)

<7> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<8> 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.

<9> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<10> 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.

<11> Next, bit 8 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 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of SAR is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<12> Comparison is continued in this way up to bit 0 of SAR.

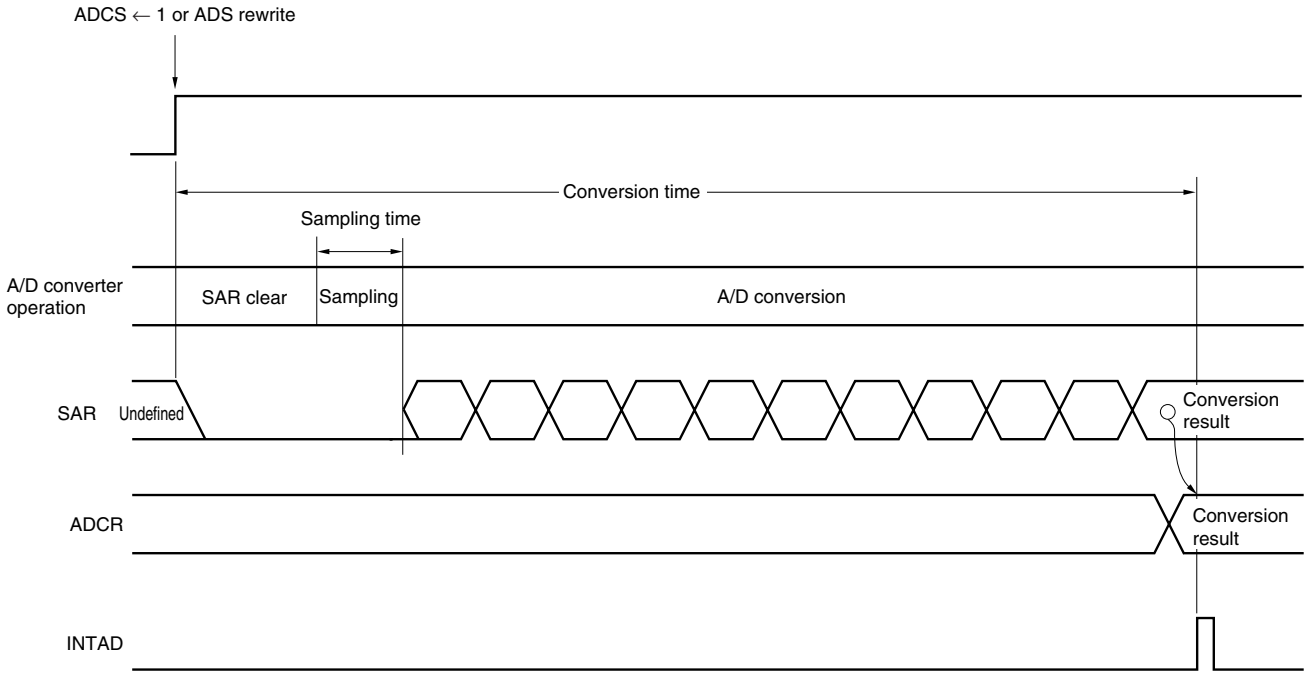
<13> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<14> Repeat steps <7> to <13>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <6>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <6>. To change a channel of A/D conversion, start from <5>.

Caution Make sure the period of <3> to <6> is 1 μs or more.

Remark Two types of A/D conversion result registers are available.
- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

<R> **Figure 10-11. Basic Operation of A/D Converter**



A/D conversion operations are performed continuously until bit 7 (ADCS) of A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 - ANI17) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT ( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

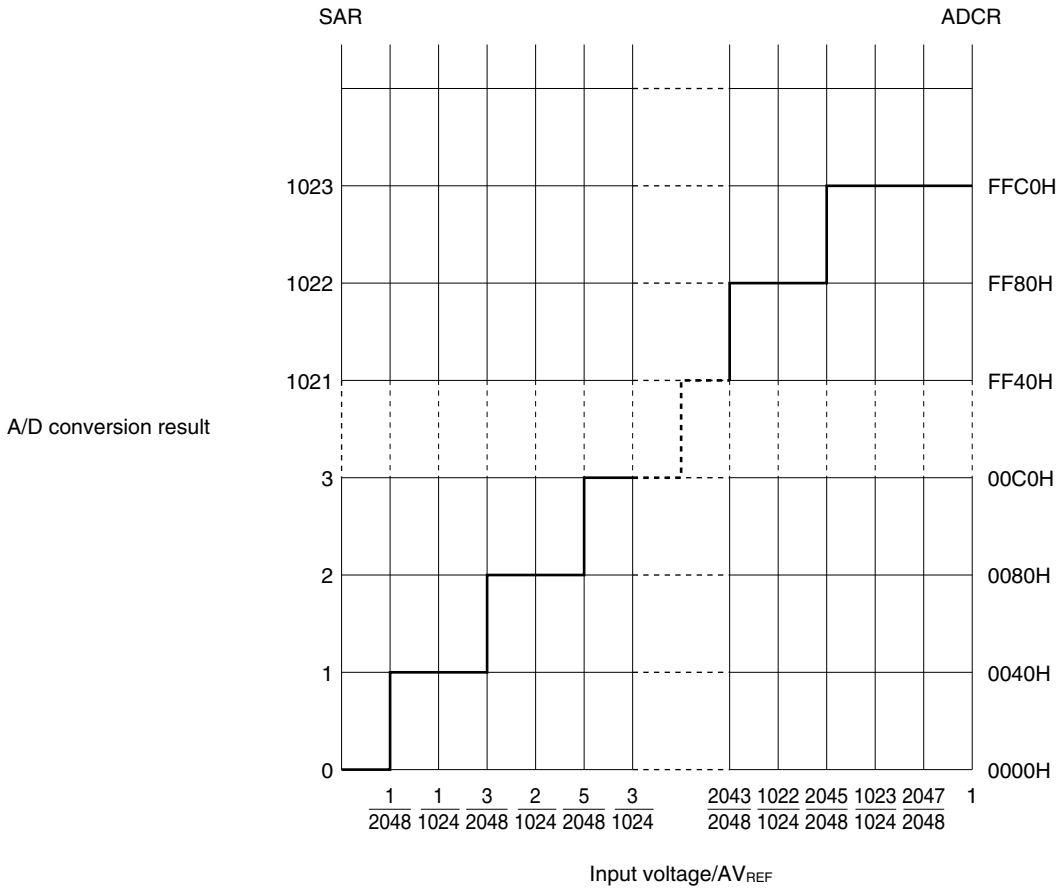
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 10-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-12. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 A/D converter operation modes

The select mode and scan mode are provided as the A/D converter operation modes.

(1) Select mode

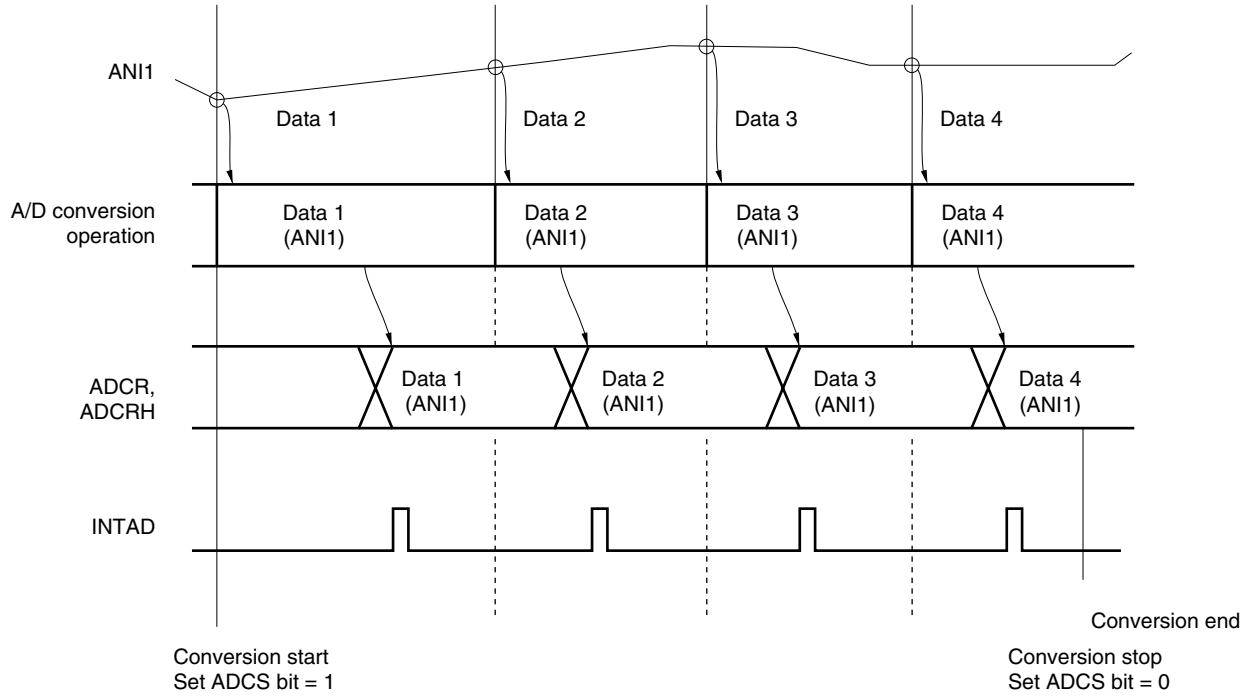
One analog input specified by the analog input channel specification register (ADS), while the ADMD bit of A/D converter mode register (ADM) is 0, is A/D converted.

When A/D conversion is complete, the conversion result is stored in the A/D conversion result register (ADCR) and the A/D conversion end interrupt request signal (INTAD) is generated.

After A/D conversion has been completed, A/D conversion is repeated successively, unless the ADCS bit is set to 0.

If anything is written to ADM or ADS during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the beginning.

Figure 10-13. Example of Select Mode Operation Timing



(2) Scan mode

The four analog input channels of scan 0 – scan 3, which are specified by the analog input channel specification register (ADS), while the ADMD bit of A/D converter mode register (ADM) is 1, are A/D converted successively. A/D conversion is performed in sequence, starting from the analog input channel specified by scan 0.

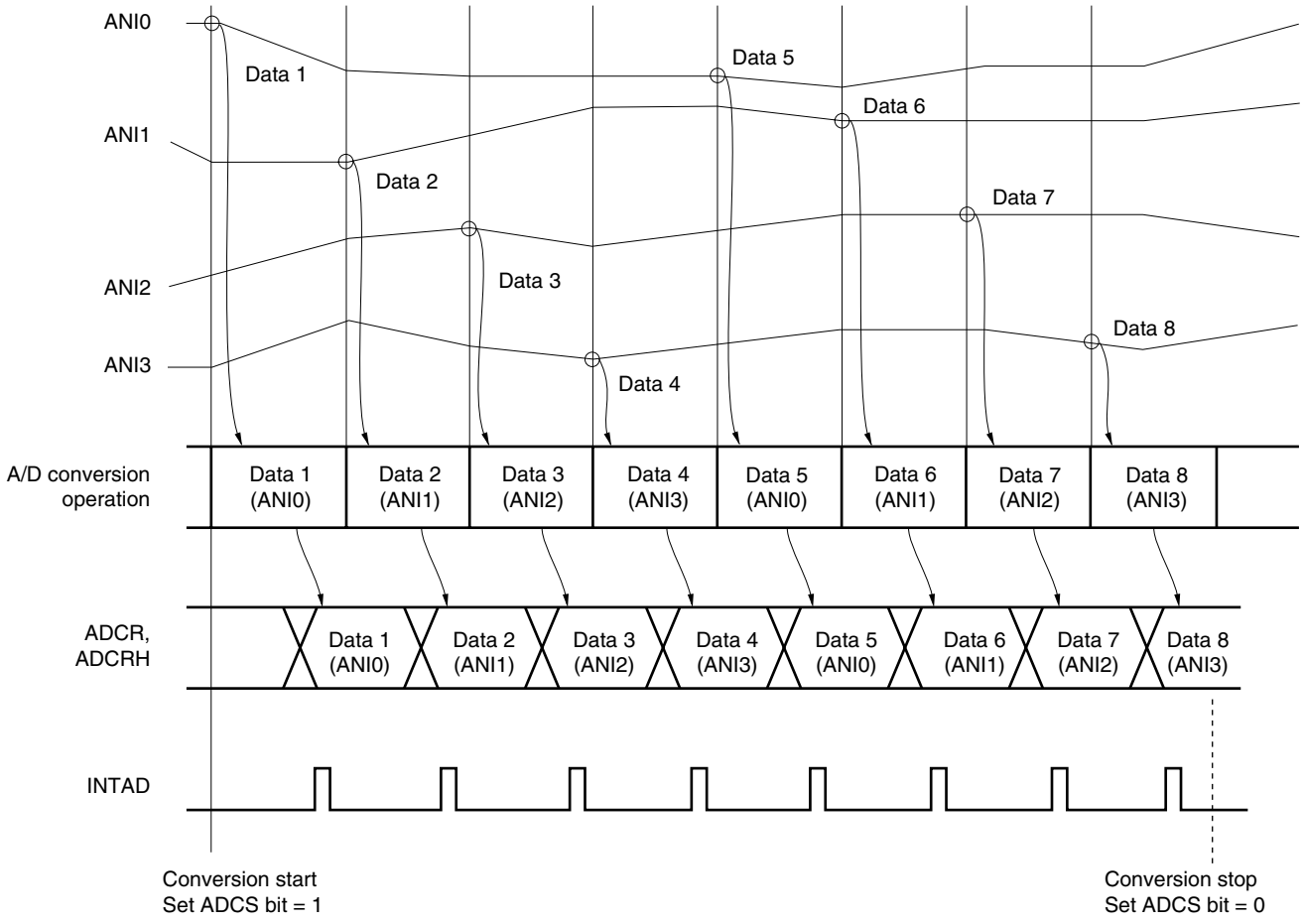
When A/D conversion of one analog input is complete, the conversion result is stored in the A/D conversion result register (ADCR) and the A/D conversion end interrupt request signal (INTAD) is generated.

The A/D conversion results of all the analog input channels are stored in ADCR. It is therefore recommended to save the contents of ADCR to RAM, once A/D conversion of one analog input channel has been completed.

When one A/D conversion ends, the next A/D conversion is started successively, until ADCS bit is set to 0.

If anything is written to ADM or ADS during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the analog input channel of scan 0.

Figure 10-14. Example of Scan Mode Operation Timing



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 - FR0, LV1, and LV0) of ADM, and select the operation mode by using bit 6 (ADMD) of ADM.
<3> Set bit 0 (ADCE) of A/D converter mode register (ADM) to 1.
<4> Set the channel to be used in the analog input mode by using bits 3 - 0 (ADPC3 - ADPC0) of the A/D port configuration register (ADPC), bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2), and bits 3 to 0.
<5> Select a channel to be used by using bits 2 to 0 (ADS2 - ADS0) of the analog input channel specification register (ADS).
<6> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.
<7> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<8> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<9> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS to start A/D conversion.
<10> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<11> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<12> Clear ADCS to 0.
<13> Clear ADCE to 0.
<14> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 0.

Cautions 1. Make sure the period of <3> to <6> is 1 μs or more.
2. <3> may be done between <4> and <5>.
3. <3> can be omitted. However, ignore data of the first conversion after <7> in this case.
4. The period from <7> to <10> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM. The period from <9> to <10> is the conversion time set using FR2 to FR0, LV1, and LV0.

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 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 10-15. Overall Error

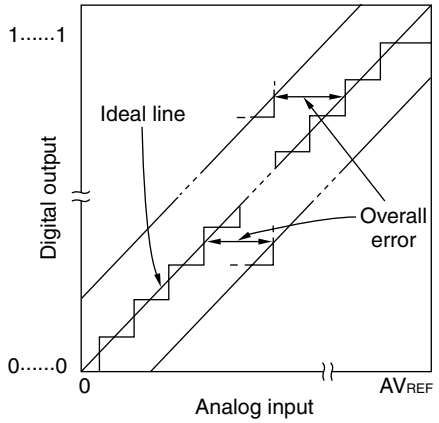
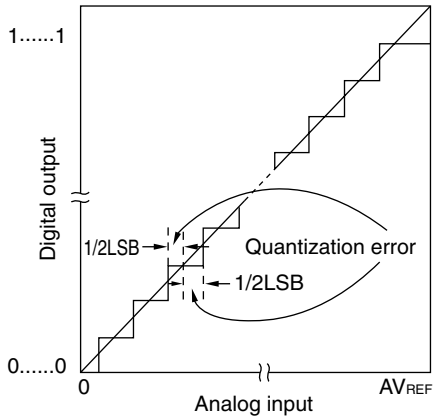


Figure 10-16. Quantization Error



(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.

Figure 10-17. Zero-Scale Error

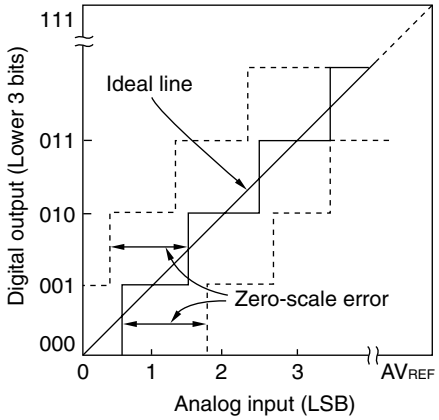
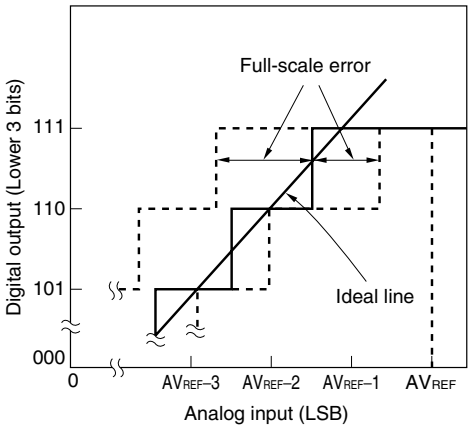


Figure 10-18. Full-Scale Error



(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 10-19. Integral Linearity Error

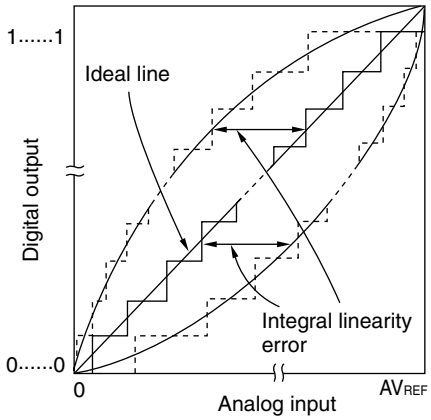
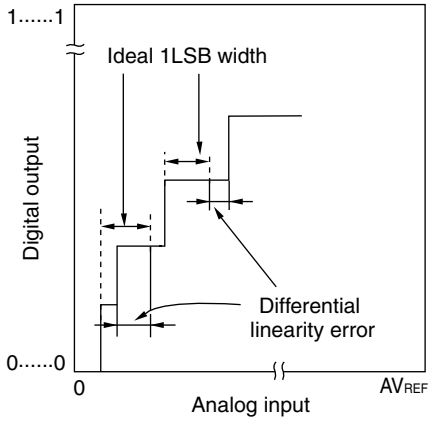


Figure 10-20. Differential Linearity Error



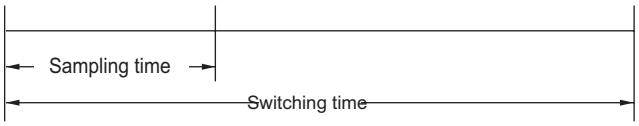
(8) Conversion time

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.

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 - ANI17

Observe the rated range of the ANI0 - ANI17 input voltage. If a voltage of AV_{REF} 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 10-bit resolution, attention must be paid to noise input to the AV_{REF} pin and pins ANI0 - ANI17.

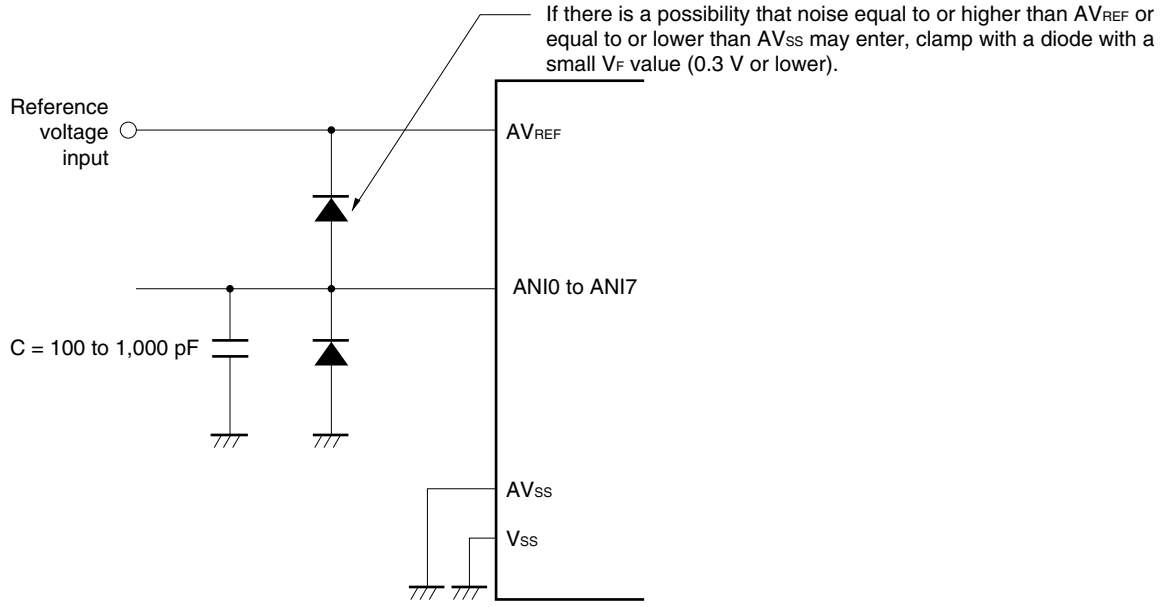
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 10-21 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 10-21. Analog Input Pin Connection



(5) ANI0/P20 - ANI7/P27

<1> The analog input pins (ANI0 - ANI7) are also used as input port pins (P20 - P27).
The analog input pins (ANI8 to ANI11) are also used as input port pins (P150 - P153).
When A/D conversion is performed with any of ANI0 - ANI17 selected, do not access P20 - P27 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 - P27 starting with the ANI0/P20 that is the furthest from AV_{REF}.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI0 - ANI17 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 accurate, 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 - ANI17 pins (see **Figure 10-21**).
Be sure to use this setting specially in case of using high speed mode.

(7) AV_{REF} pin input impedance

A series resistor string of several tens of kΩ is connected between the AV_{REF} and AV_{SS} pins.

Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV_{REF} and AV_{SS} pins, resulting in a large reference voltage error.

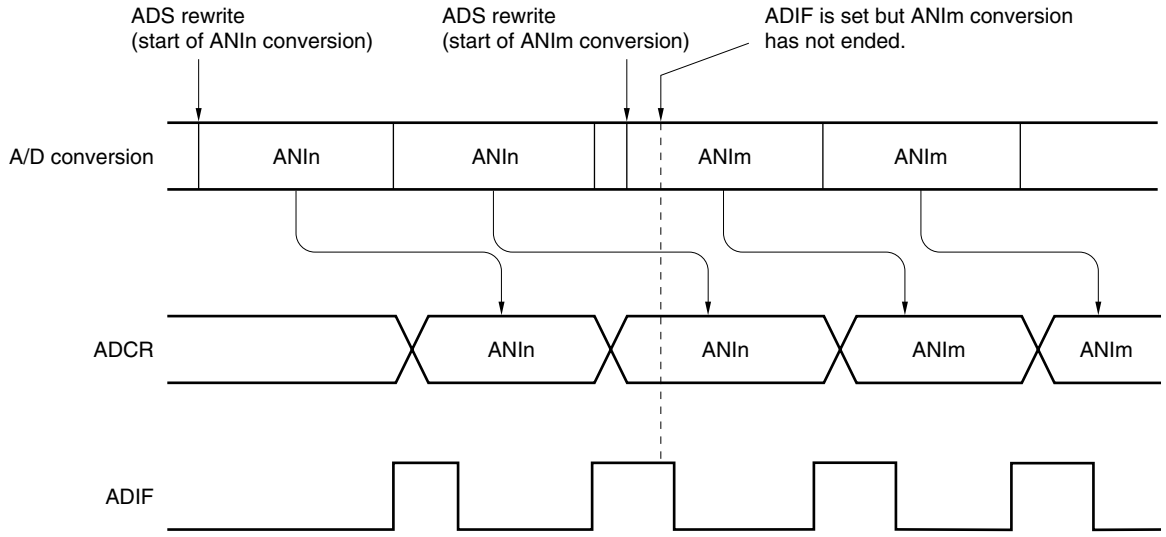
(8) Interrupt request flag (ADIF)

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 10-22. Timing of A/D Conversion End Interrupt Request Generation



Remark n = 0 - 7
m = 0 - 7

(9) Conversion results just after A/D conversion start

<R> The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register (ADM), analog input channel specifications 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, ADS, or ADPC. Using 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 10-23. Internal Equivalent Circuit of ANIn Pin

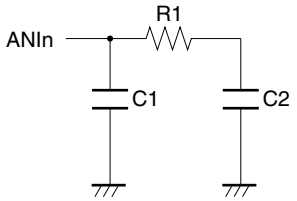


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

<R>

AV_{REF}	Mode	R1	C1	C2
1.8 V ≤ V_{DD} < 3.6 V	**Standard**	18.6 kΩ	8 pF	6.3 pF
	High speed 2	7.8 kΩ		

Remarks 1. The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

2. n = 0 - 7

<R> **(12) Starting the A/D converter**

Start the A/D converter after the AV_{REF} voltage stabilize.

CHAPTER 11 SERIAL ARRAY UNIT

Each serial array unit has four serial channels, each of which can be used for 3-wire serial (CSI), UART, and simplified I2C communication.

Function assignment of each channel supported by the 78K0R/KC3-L, KE3-L is as shown below.

In case of 78K0R/KC3-L

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	-	-	-
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

In case of 78K0R/KE3-L

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 note	UART2 note	IIC20 note
	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.

In this chapter, related to control register 11.2 serial array unit, configuration of Serial Array Unit of 78K0R/KC3-L and KE3-L has been taken as example and explained.

11.1 Functions of Serial Array Unit

Each serial interface supported by the 78K0R/KC3-L, KE3-L has the following features.

11.1.1 3-wire serial I/O (CSI00, CSI01, CSI20)

Data is transmitted or received in synchronization with the serial clock ($\overline{SCK}$) output from the master channel.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock ($\overline{SCK}$), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI20) Communication.**

[Data transmission/reception]

- Data length of 7 , 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
- Maximum transfer rate

  During master communication: Max. $f_{CLK}/4$, during slave communication: Max. $f_{MCK}/6$ Note

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

11.1.2 UART (UART0 - UART3)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit 0 with an external interrupt (INTP0).

For details about the settings, see **11.6 Operation of UART (UART0 to UART3) Communication.**

[Data transmission/reception]

- Data length of 5, 7, 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

} Using the external interrupt (INTP0) and timer array unit 0

11.1.3 Simplified I^2C (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 I^2C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **11.7 Operation of Simplified I^2C (IIC10, IIC20)**

[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 I^2C]

- 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 **11.7.3 (2) Processing flow** for details.

Remarks 1. To use the full-function I^2C bus, see CHAPTER 12 SERIAL INTERFACE IICA.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

11.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

Table 11-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn)[Note1]
Serial clock I/O	$\overline{SCK00}$, $\overline{SCK10}$, $\overline{SCK20}$[Note2] pins (for 3-wire serial I/O), SCL10, SCL20[Note2] pins (for simplified I^2C)
Serial data input	SI00, SI10, SI20[Note2] pins (for 3-wire serial I/O), RxD0, RxD1, RxD2[Note2] pins (for UART), RxD3 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO10, SO20[Note2] pins (for 3-wire serial I/O), TxD0, TxD1, TxD2[Note2] pins (for UART), TxD3 pin (for UART supporting LIN-bus), output controller
Serial data I/O	SDA10, SDA20[Note2] 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 registers 0, 1, 14[Note2] (PIM0, PIM1, PIM14[Note2]) • Port output mode registers 0, 1, 14[Note2] (POM0, POM1, POM14[Note2]) • Port mode registers 0, 1, 4, 14[Note2] (PM0, PM1, PM4, PM14[Note2]) • Port registers 0, 1, 4, 14[Note2] (P0, P1, P4, P14[Note2])

Notes 1. 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)

2. 78K0R/KE3-L Only

Remark In case of 78K0R/KC3-L

m: Unit number (m = 0, 1), n: Channel number (n = 0 - 3), mn = 00 - 03, 12, 13

p: CSI number (p = 00, 10) q:UARTNumber(q = 0, 1, 3) r:IIC Number(r = 10)

In case of 78K0R/KE3-L

m: Unit number (m = 0, 1) n:Channel number(n = 0-3) mn = 00-03, 10-13

p: CSI Number (p = 00, 10, 20), q: UART number (q = 0-3), r: IIC number (r = 10, 20)

Figure 11-1 shows the block diagram of serial array unit 0.

Figure 11-1. Block Diagram of Serial Array Unit 0

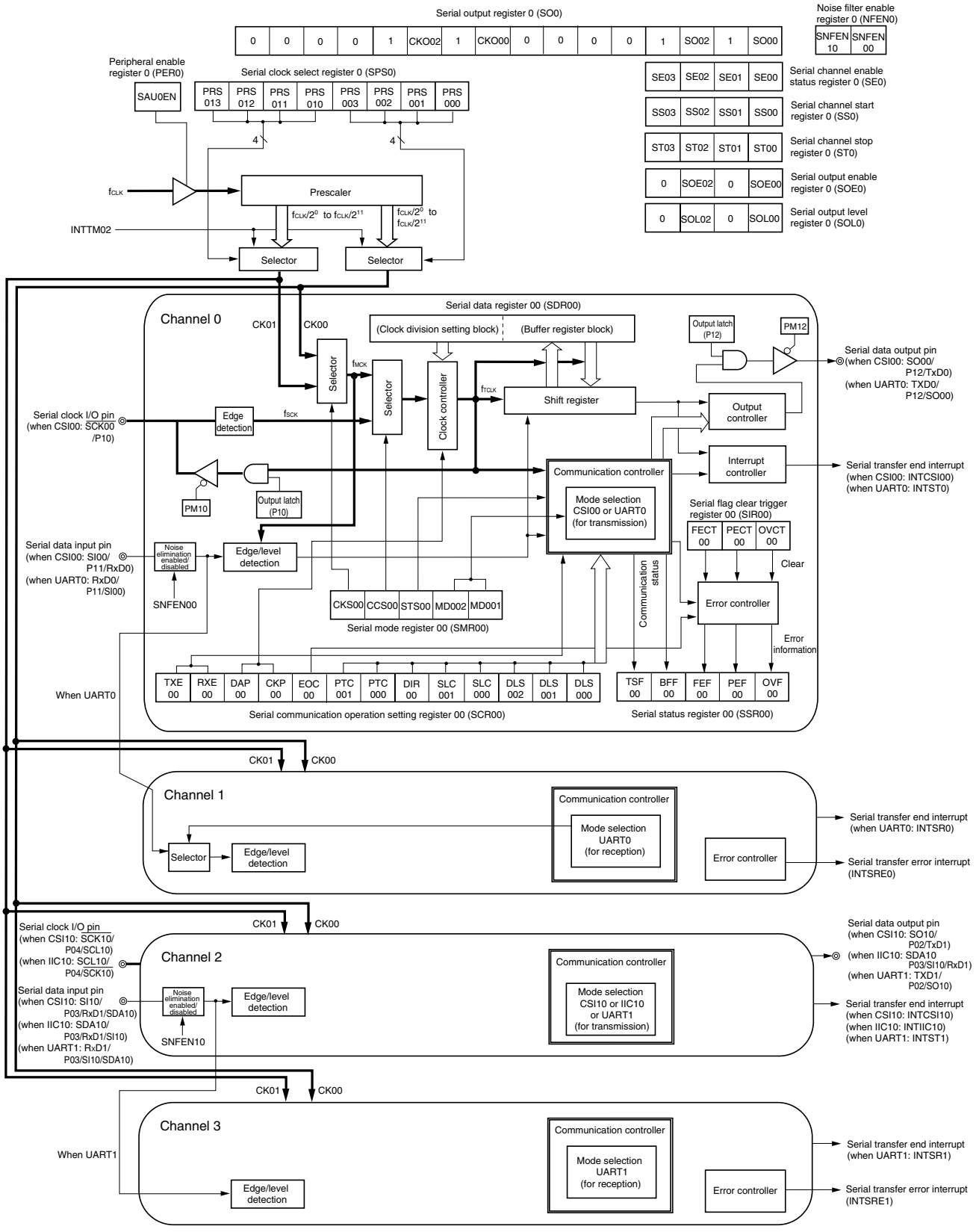


Figure 11-2, 11-3 shows the block diagram of serial array unit 1.

Figure 11-2. Block Diagram of Serial Array Unit 1 In case of 78K0R/KC3-L

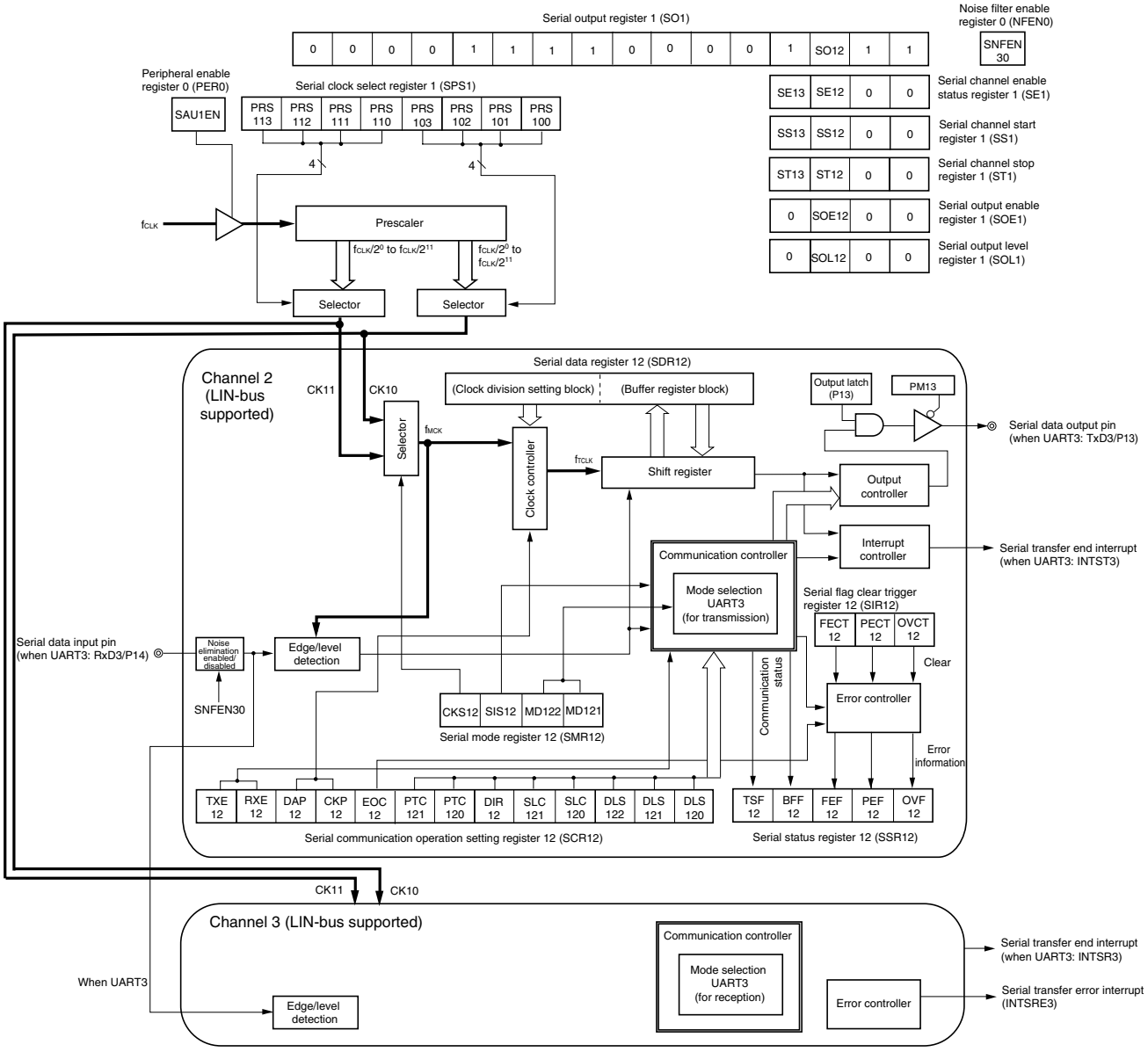
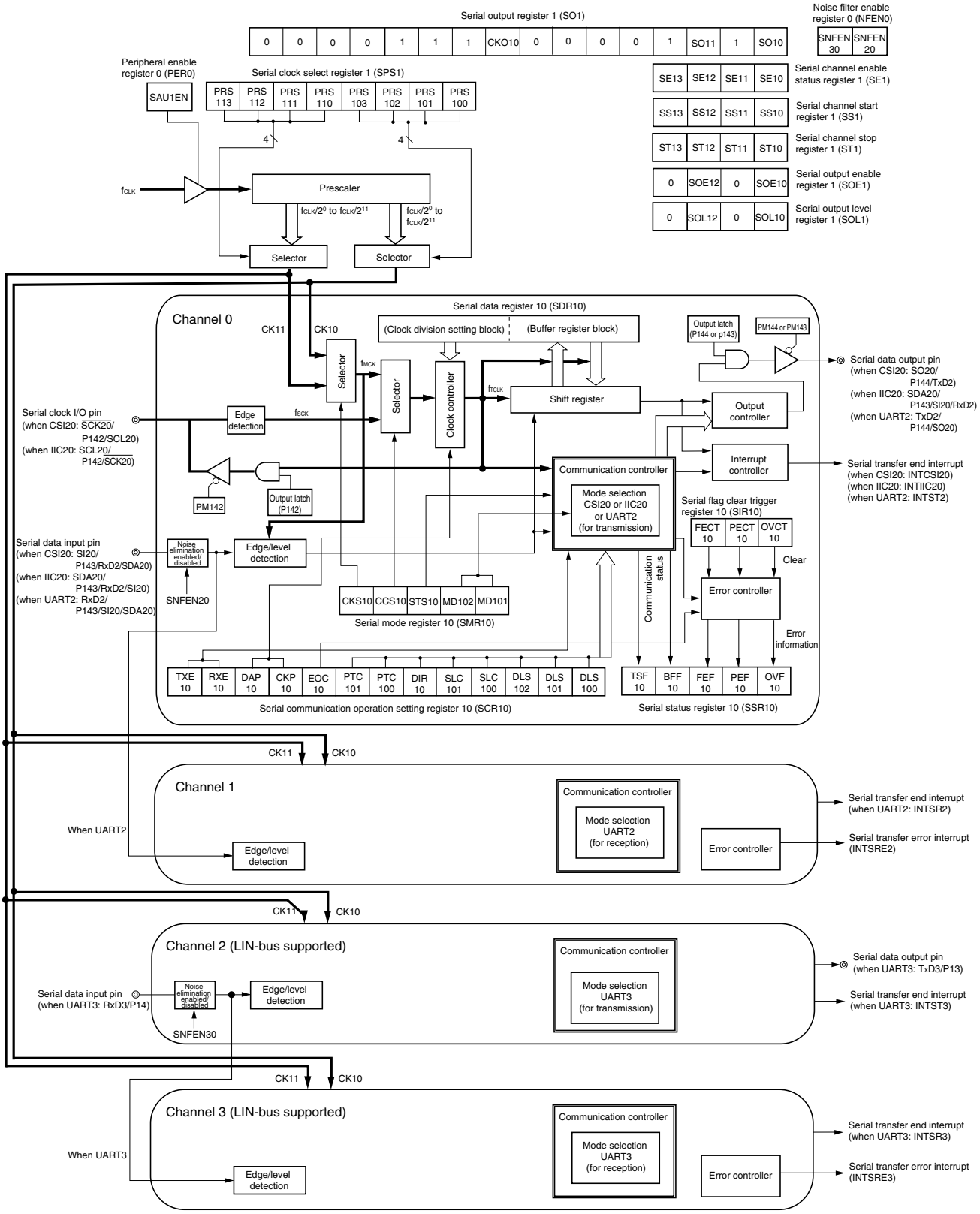


Figure 11-3. Block Diagram of Serial Array Unit 1 In case of 78K0R/KE3-L



(1) Shift register

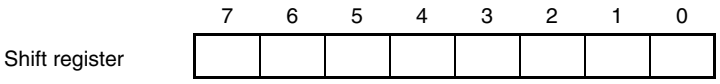
This is an 8-bit register that converts parallel data into serial data or vice versa.

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 (f_{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 - 4 of SDRmn register) (settable in UART mode only)
- 7-bit data length (stored in bits 0 - 6 of SDRmn register)
- 8-bit data length (stored in bits 0 - 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 **In case of 78K0R/KC3-L**

m: Unit number (m = 0, 1), n: Channel number (n = 0 - 3), mn = 00 - 03, 12, 13

p: CSI number (p = 00, 10) q:UARTNumber(q = 0, 1, 3) r:IIC Number(r = 10)

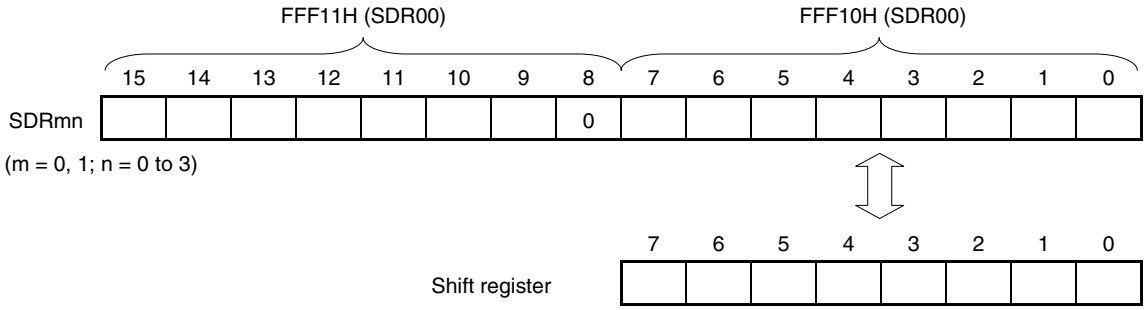
In case of 78K0R/KE3-L

m: Unit number (m = 0, 1) n:Channel number(n = 0-3) mn = 00-03, 10-13

p: CSI Number (p = 00, 10, 20), q: UART number (q = 0-3), r: IIC number (r = 10, 20)

Figure 11-4. 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 11.3 Registers Controlling Serial Array Unit.

2. In case of 78K0R/KC3-L

m: Unit number (m = 0, 1), n: Channel number (n = 0 - 3), mn = 00 - 03, 12, 13

In case of 78K0R/KE3-L

m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

11.3 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 registers 0, 1, 14 (PIM0, PIM1, PIM14)
- Port output mode registers 0, 1, 14 (POM0, POM1, POM14)
- Port mode registers 0, 1, 14 (PM0, PM1, PM14)
- Port registers 0, 1, 14 (P0, P1, P14)

Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)
mn = 00 to 03, 10 to 13

(1) Peripheral enable register 0 (PER0)

PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

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 11-5. 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	0	ADCEN	IICAEN	SAU1EN	SAU0EN	0	TAU0EN

SAUmEN	Control of serial array unit m input clock supply
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Enables input clock supply. • 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 noise filter enable register (NFEN0), port input mode registers (PIM0, PIM1, PIM14), port output mode registers (POM0, POM1, POM14), port mode registers (PM0, PM1, PM4, PM14), and port registers (P0, P1, P4, P14)).

2. After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

3. Be sure to clear bit 6 to 0.

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 11-6. 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} = 10 MHz	f_{CLK} = 12 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	10 MHz	12 MHz	16 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	5 MHz	6 MHz	8 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	2.5 MHz	3 MHz	4 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	1.25 MHz	1.5 MHz	2 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	625 kHz	750 kHz	1 MHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	313 kHz	375 kHz	500 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	156 kHz	187.5 kHz	250 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	78.1 kHz	93.7 kHz	125 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	39.1 kHz	46.8 kHz	62.5 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	19.5 kHz	23.4 kHz	31.2 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	9.77 kHz	11.7 kHz	15.6 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	4.88 kHz	5.85 kHz	7.81 kHz	9.77 kHz
1	1	1	1	INTTM02 if m = 0 Note 2. In the case of m = 1, setting is prohibited.				
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 (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, sub system clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting the TIS02 bit of timer input select register 0 (TIS0) to 1) and selecting INTTM02 by using the SPS0 register in channel 2 of TAU0. When changing f_{CLK}, however, SAU0 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 bit 2 (SAU0EN) of the PER0 register and bit 3 (SAU1EN) of the PER0 register to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} 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 (f_{MCK}), specify whether the serial clock (f_{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 11-7. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) - F0116H, F0117H (SMR03), After reset: 0020H R/W
F0150H, F0151H (SMR10) - 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

CKS mn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock 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.	

CCS mn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by CKSmn bit
1	Clock input from f_{SCK} pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STS mn	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of RxDq 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 - 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 - 3), mn = 00 - 03, 10 - 13

Figure 11-7. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) - F0116H, F0117H (SMR03) , F0150H, F0151H (SMR10) - 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	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 (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 - 03, 10 - 13

(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.

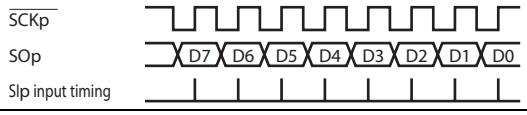
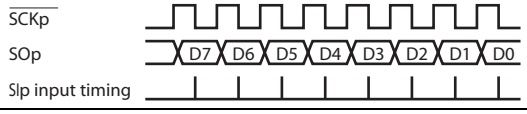
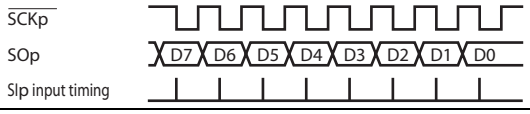
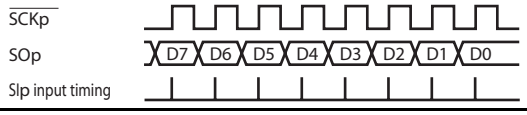
Reset signal generation sets this register to 0087H.

Figure 11-8. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) - F011EH, F011FH (SCR03), After reset: 0087H R/W
F0158H, F0159H (SCR10) - 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	DLS mn2	DLS mn1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in 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.			

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.	

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), mn = 00 - 03, 10 -13,
p: CSI number (p = 00, 01, 10, 20)

Figure 11-8. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) - F011EH, F011FH (SCR03), F0158H, F0159H (SCR10) - 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	DLS mn2	DLS mn1	DLS mn0

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity. Note.	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.		

DLS mn2	DLS mn1	DLS mn0	Setting of data length in CSI and UART modes
1	0	0	5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
1	1	0	7-bit data length (stored in bits 0 to 6 of SDRmn register)
1	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.			

<R> **Note** 0 is always added regardless of the data contents.

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), mn = 00 to 03, 10 to 13

(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 (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of SDRmn is used as the transfer clock.

The lower 8 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 bits.

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 11-9. 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 (f_{MCK})
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

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 simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.**

<R> **4. Do not write eight bits to the lower eight bits if operation is stopped (SEmn = 0). (If these bits are written to, the higher seven bits are cleared to 0.)**

Remarks 1. For the function of the lower 8 bits of SDRmn, see 11.2 Configuration of Serial Array Unit.

2. m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)
mn = 00 to 03, 10 to 13

(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 11-10. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), F0140H, F0141H (SSR10) to 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	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Caution If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

Figure 11-10. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R
F0140H, F0141H (SSR10) to 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	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

FEF mn	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).

<Clear condition>
- 1 is written to the FECTmn bit of the SIRmn register.

<Set condition>
- A stop bit is not detected when UART reception ends.

PEF mn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).

<Clear condition>
- 1 is written to the PECTmn bit of the SIRmn register.

<Set condition>
- The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).
- No ACK signal is returned from the slave channel at the ACK reception timing during I^2C transmission (ACK is not detected).

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs

<Clear condition>
- 1 is written to the OVCTmn bit of the SIRmn register.

<Set condition>
- Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- Transmit data is not ready for slave transmission or transmission and reception in CSI mode.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

(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 11-11. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H(SIR00), F010AH, F010BH(SIR01), F010CH, F010DH(SIR02), F010EH, F010FH (SIR03), F0148H, F0149H (SIR10) - 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	FEC Tmn	PEC Tmn	OVC Tmn

FEC Tmn	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Caution Be sure to clear bits 15 - 3 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

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 m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of 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 an 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears this register to 0000H.

Figure 11-12. 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
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

(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 an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears this register to 0000H.

Figure 11-13. 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).

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

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 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 an 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears this register to 0000H.

Figure 11-14. 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), mn = 00 - 03, 10 - 13

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 an 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears this register to 0000H.

Figure 11-15. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	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 of SOE0, and bits 1 and 15 to 3 of SOE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 - 3),
mn = 00 - 03, 10 - 13

(12) Serial output register m (SOm)

SOmn 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 CKOmn 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 P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$, P12/SO00/TxD0, P13/TxD3, P45/SO01, P142/$\overline{\text{SCK20}}$/SCL20, P143/SI20/SDA20/RxD2, or P144/SO20/TxD2 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 11-16. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	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, 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 - 3), mn = 00 - 03, 10 - 13

(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 11-17. 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 bit 15-3, 1 of SOL and bits 15 – 3 of SOL1, 1 to “0”.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00 - 03, 10 - 13

(14) Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to realize a LIN-bus communication operation by UART3 in coordination with an external interrupt and the timer array unit TAU0.

When bit 0 is set to 1, the input signal of the serial data input (RxD3) pin is selected as an external interrupt (INTP0), so that wake up signal can be detected, the low width of the sync break field, and the pulse width of the sync field can be measured by the timer.

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 11-18. 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	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit (TAU0)
0	Do not use the input of the Timer array unit channel 7 pin as a timer input (normal operation).
1	Input signal of RxD3 pin is used as timer input (detects the wakeup signal and measures the low width of the sync break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD3 pin as an external interrupt (detects the wakeup signal).

Caution Be sure to clear bits 7 - 2 to "0".

(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 hardware 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 11-19. 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 pin (RxD3/P14)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the P14 pin.	

SNFEN20	Use of noise filter of RxD2 pin (RxD2/SDA20/SI20/P143)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the other than RxD2 pin.	

SNFEN10	Use of noise filter of RxD1 pin (RxD1/SDA10/SI10/P03)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN10 to 1 to use the RxD1 pin. Clear SNFEN10 to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin (RxD0/SI00/P11)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7, 5, 3, and 1 to "0".

(16) Port input mode registers 0, 1, and 14 (PIM0, PIM1, PIM14)

These registers set the input buffer of ports 0, 1, and 14 in 1-bit units.

PIM0, PIM1, and PIM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 11-20. Format of Port Input Mode Registers 0, 1, and 14 (PIM0, PIM1, PIM14)

Address F0040H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM0	0	0	0	PIM04	PIM03	0	0	0

Address F0041H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM1	0	0	0	0	0	0	PIM11	PIM10

Address F004EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM14	0	0	0	0	PIM143	PIM142	0	0

PIMmn	Pmn pin input buffer selection (m = 0, 1, 14; n = 0 - 4)
0	Normal input buffer
1	TTL input buffer

(17) Port output mode registers 0, 1, 14 (POM0, POM1, POM14)

These registers set the output mode of ports 0, 1, and 14 in 1-bit units.

POM0, POM1, and POM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 11-21. Format of Port Output Mode Registers 0, 1, and 14 (POM0, POM1, POM14)

Address F0050H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM0	0	0	0	POM04	POM03	POM02	0	0

Address F0051H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM1	0	0	0	0	0	POM12	0	POM10

Address F005EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM14	0	0	0	POM144	POM143	POM142	0	0

POMmn	Pmn pin output buffer selection (m = 0, 1, 14; n = 0, 2 - 4)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(18) Port mode registers 0, 1, 4, 14 (PM0, PM1, PM4, PM14)

These registers set input/output of ports 0, 1, 4 and 14 in 1-bit units.

When using the P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/$\overline{SCK10}$/SCL10, P10/$\overline{SCK00}$/, P12/SO00/TxD0/, P13/TxD3/, P45/SO01, P142/$\overline{SCK20}$/SCL20, P143/SI20/SDA20/RxD2, and P144/SO20/TxD2 pins for serial data output or serial clock output, clear the PM02, PM03, PM04, PM10, PM12, PM13, PM142, PM143, and PM144 bits to 0, and set the output latches of P02, P03, P04, P10, P12, P13, P142, P143, and P144 to 1.

When using the P03/SI10/RxD1/SDA10, P04/$\overline{SCK10}$/SCL10, P10/$\overline{SCK00}$, P11/SI00/RxD0, P14/RxD3, P43/SCK01, P44/SI01, P142/SCK20/SCL20, pins for serial data input or serial clock input, set the PM03, PM04, PM10, PM11, PM14, PM142, and PM143 bits to 1. At this time, the output latches of P03, P04, P10, P11, P14, P43, P44, P142, and P143 may be 0 or 1.

PM0, PM1, PM4, and PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 11-22. Format of Port Mode Registers 0, 1, 4, and 14 (PM0, PM1, PM4, PM14)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	0	0	PM04	PM03	PM02	1	1

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	0	0	1	0	PM43	PM42	PM41	PM40

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	PM144	PM143	PM142	1	PM140

PMmn	Pmn pin I/O mode selection (m = 0, 1, 4, 14; n = 0 - 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Cauton 1. Be sure to set Bit 5, 6, of PM0 bit 4, 6, 7 of PM4 to 0.

Be sure to set it to 0, when it returns to default value after reset

2. Be sure to set bit 0, 1, 7 of PM0 bit5 PM4 bit1, 5-7 of PM14 to 1.

Remark Given above is a format of port mode register 0, 1 4, 14 of 78K0R/KE3-L. See 4. 3 (1) port mode register (PMxx), for port mode register format of78K0R/KC3-L.

11.4 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 P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/SCK10/SCL10, P10/SCK00, P11/SI00/RxD0, P12/SO00/TxD0, P13/TxD3, P14/RxD3, P142/SCK20/SCL20, P143/SI20/SDA20/RxD2, or P144/SO20/TxD2 pin can be used as ordinary port pins in this mode.

11.4.1 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 11-23. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCEN		ADCEN	IICAEN	SAU1EN	SAU0EN	0	TAU0EN
	×	0	×	×	0/1	0/1		×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Cautions 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for noise filter enable register (NFEN0), port input mode registers (PIM0, PIM1, PIM14), port output mode registers (POM0, POM1, POM14), port mode registers (PM0, PM1, PM4, PM14), and port registers (P0, P1, P4, P14)).

2. Be sure to clear bit 1 and 6 to 0.

Remark m: Unit number (m = 0, 1), □ : Setting disabled (fixed by hardware)
×: 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

11.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 11-24. 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 SO10 and SO12 value of the SO1 register can be set by software.

Figure 11-24. 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, SO10, and SO12 bits to “1”.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00-03, 10-13

2. : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

11.5 Operation of 3-Wire Serial I/O (CSI00, CSI10, CSI20) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- 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
- Maximum transfer rate

  During master communication: Max. fCLK/4, during slave communication: Max. fMCK/6 Note

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see CHAPTER 28 ELECTRICAL SPECIFICATIONS.

The channels supporting 3-wire serial I/O (CSI00, CSI10, CSI20) changes as below

In case of 78K0R/KC3-L

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	–	–	–
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

In case of 78K0R/KC3-L

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, CIS10, CSI20) performs the following six types of communication operations.

- Master transmission (See **11.5.1**.)
- Master reception (See **11.5.2**.)
- Master transmission/reception (See **11.5.3**.)
- Slave transmission (See **11.5.4**.)
- Slave reception (See **11.5.5**.)
- Slave transmission/reception (See **11.5.6**.)

11.5.1 Explains about master transmission, configuration of Serial Array Unit of 78K0R/KE3-L has been taken as example and explained. See 11.3 register Controlling Serial Array Unit for KC3-L and KE3-L.

11.5.1 Master transmission

Master transmission is that the 78K0R/KC3-L, KE3-L outputs a transfer clock and transmits 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	$\overline{\text{SCK00}}$, SO00	$\overline{\text{SCK10}}$, SO10	$\overline{\text{SCK20}}$, 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. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] 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 28 ELECTRICAL SPECIFICATIONS).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(1) Register setting

Figure 11-25. 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	CKOm1 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							

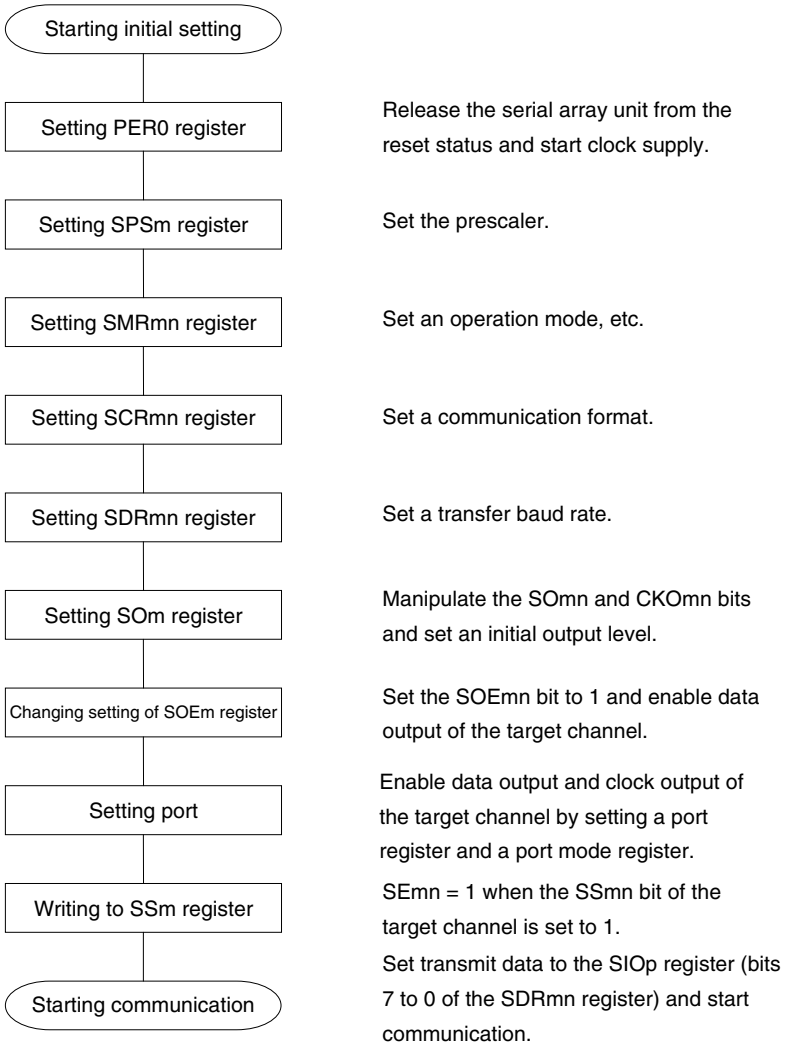
SIOp

Remarks 1 **m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10**
p: CSI number (p = 00, 10, 20)

2 **□: 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

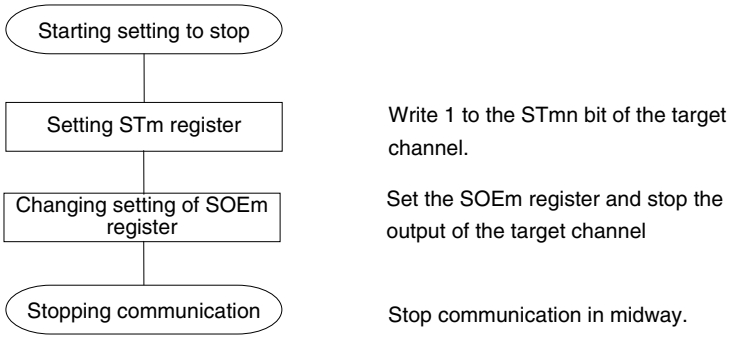
(2) Operation procedure

Figure 11-26. Initial Setting Procedure for Master Transmission



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

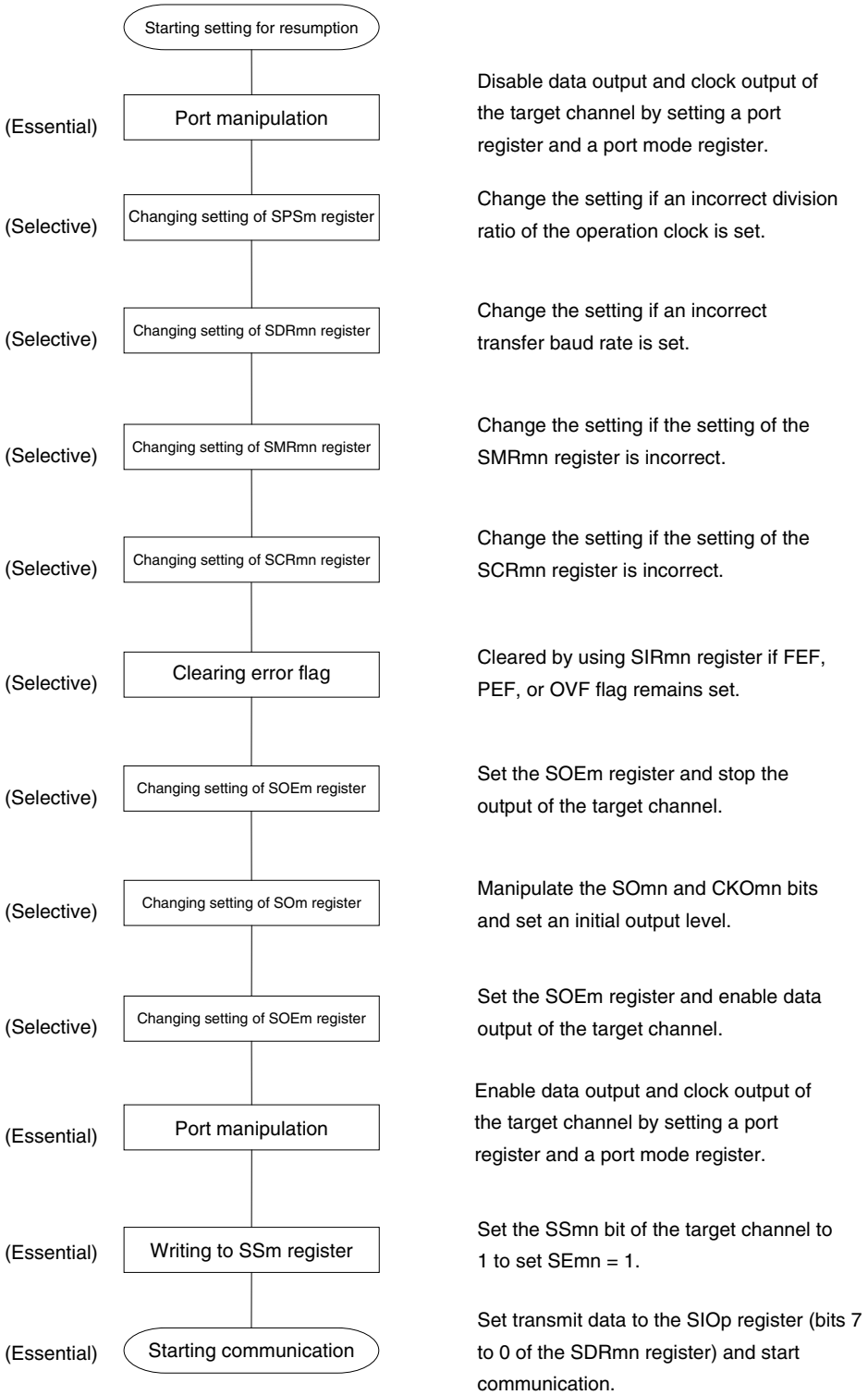
Figure 11-27. Procedure for Stopping Master Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, reset the SOm register (see Figure 11-27 Procedure for Resuming Master Transmission).

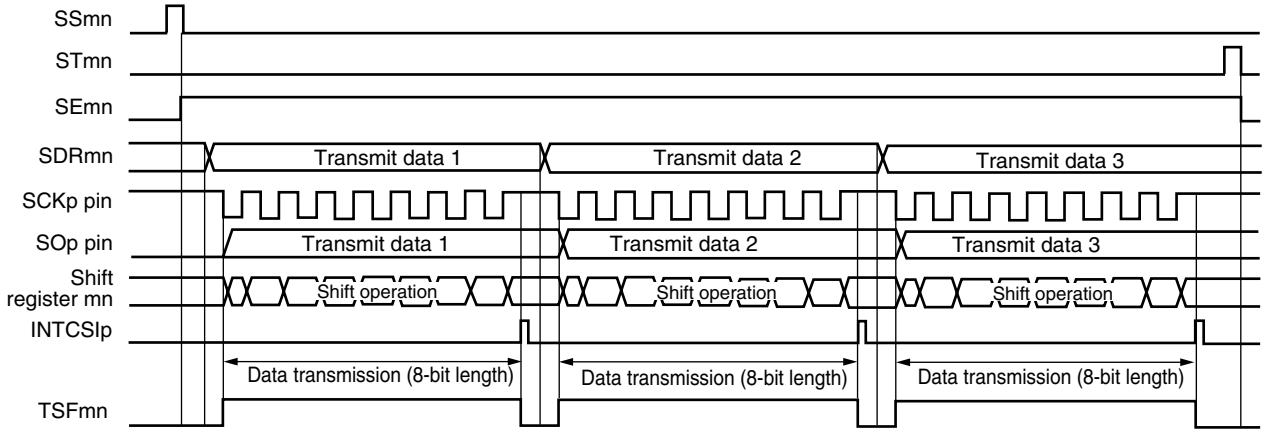
2. p: CSI number (p = 00, 10, 20)

Figure 11-28. Procedure for Resuming Master Transmission



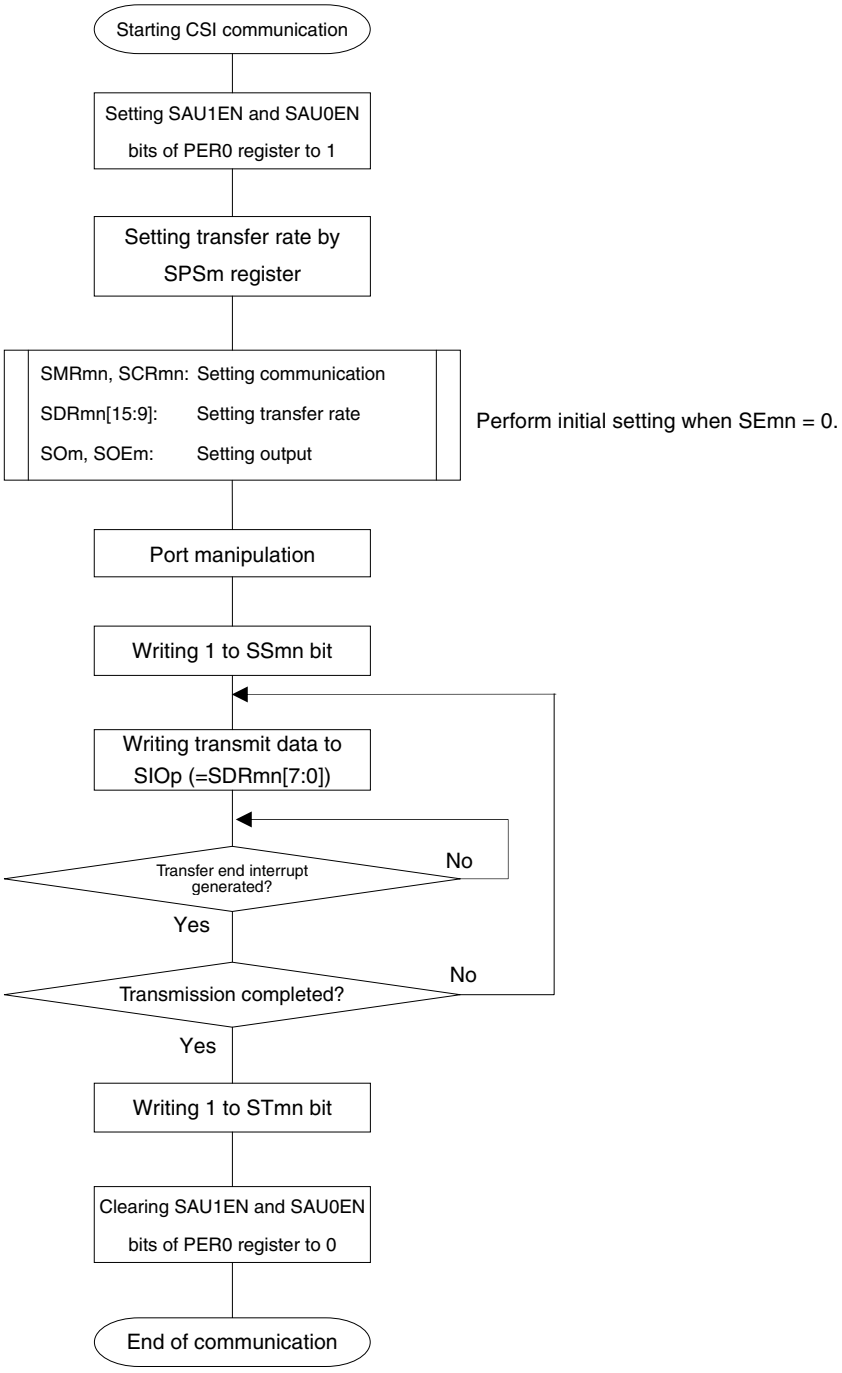
(3) Processing flow (in single-transmission mode)

Figure 11-29. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

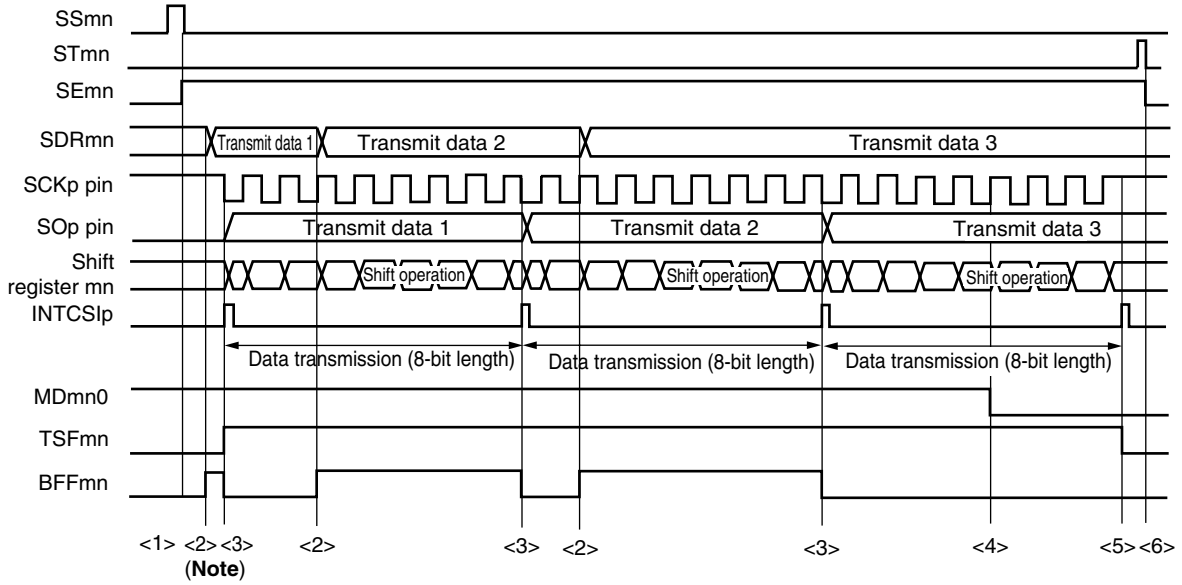
Figure 11-30. Flowchart of Master Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-31. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

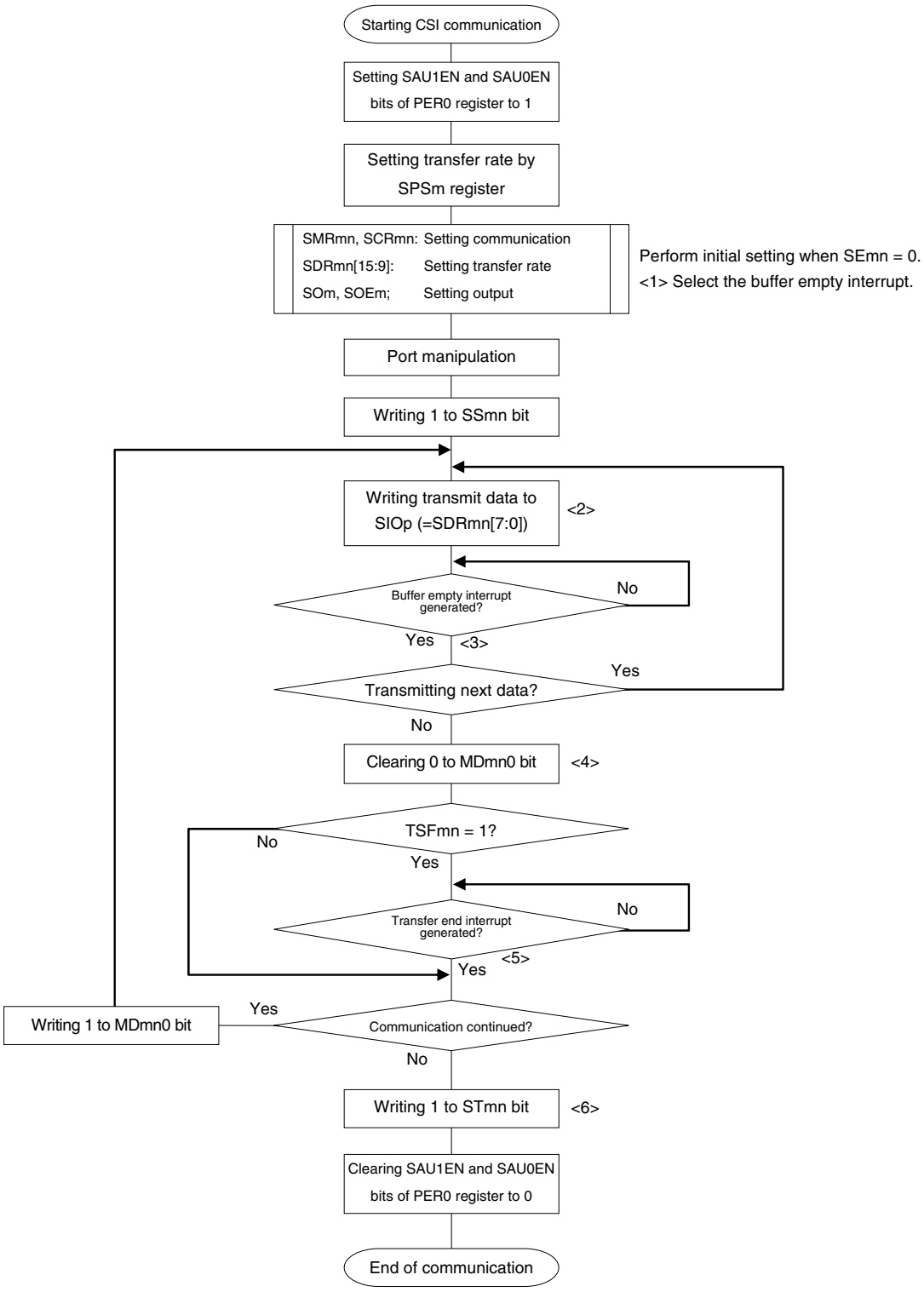


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, 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), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

Figure 11-32. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 11-30 Timing Chart of Master Transmission (in Continuous Transmission Mode).

11.5.2 Master reception

Master reception is that the 78K0R/KC3-L, KE3-L outputs a transfer clock and receives 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	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK10}}$, SI10	$\overline{\text{SCK20}}$, 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. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] 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 first 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 28 ELECTRICAL SPECIFICATIONS).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(1) Register setting

Figure 11-33. 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.

	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 ×	1	SOm0 ×

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)

	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 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

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)

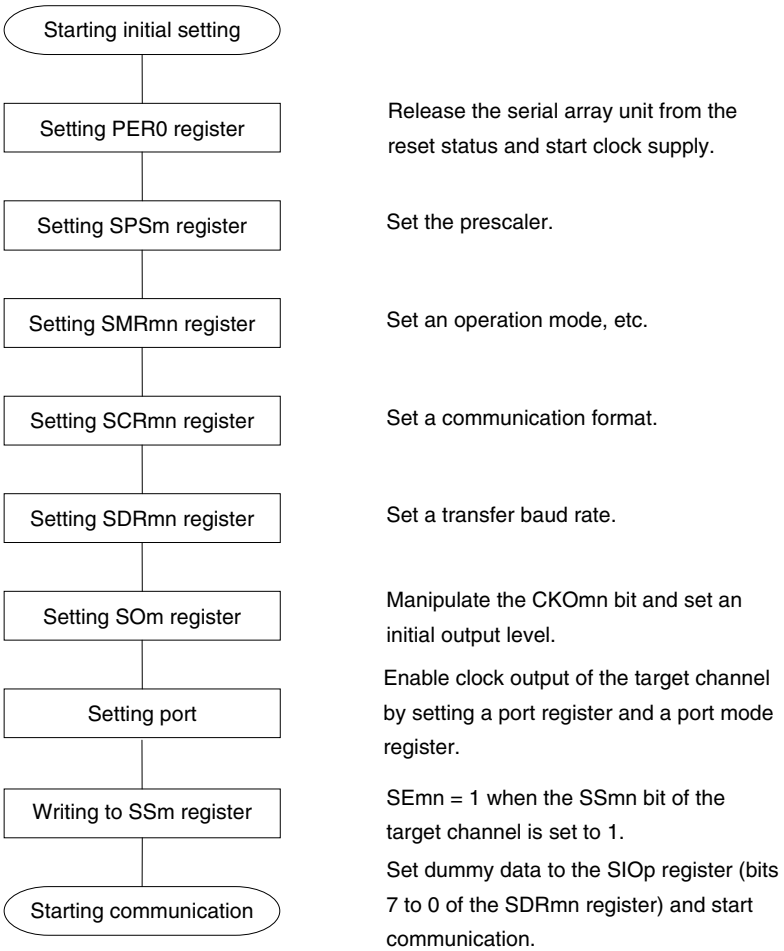
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Receive data register (Write FFH as dummy data.)							

SIOp

Remarks 1 m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)
2 □ : 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

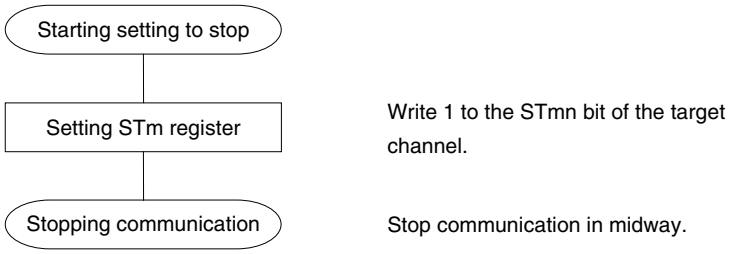
(2) Operation procedure

Figure 11-34. Initial Setting Procedure for Master Reception



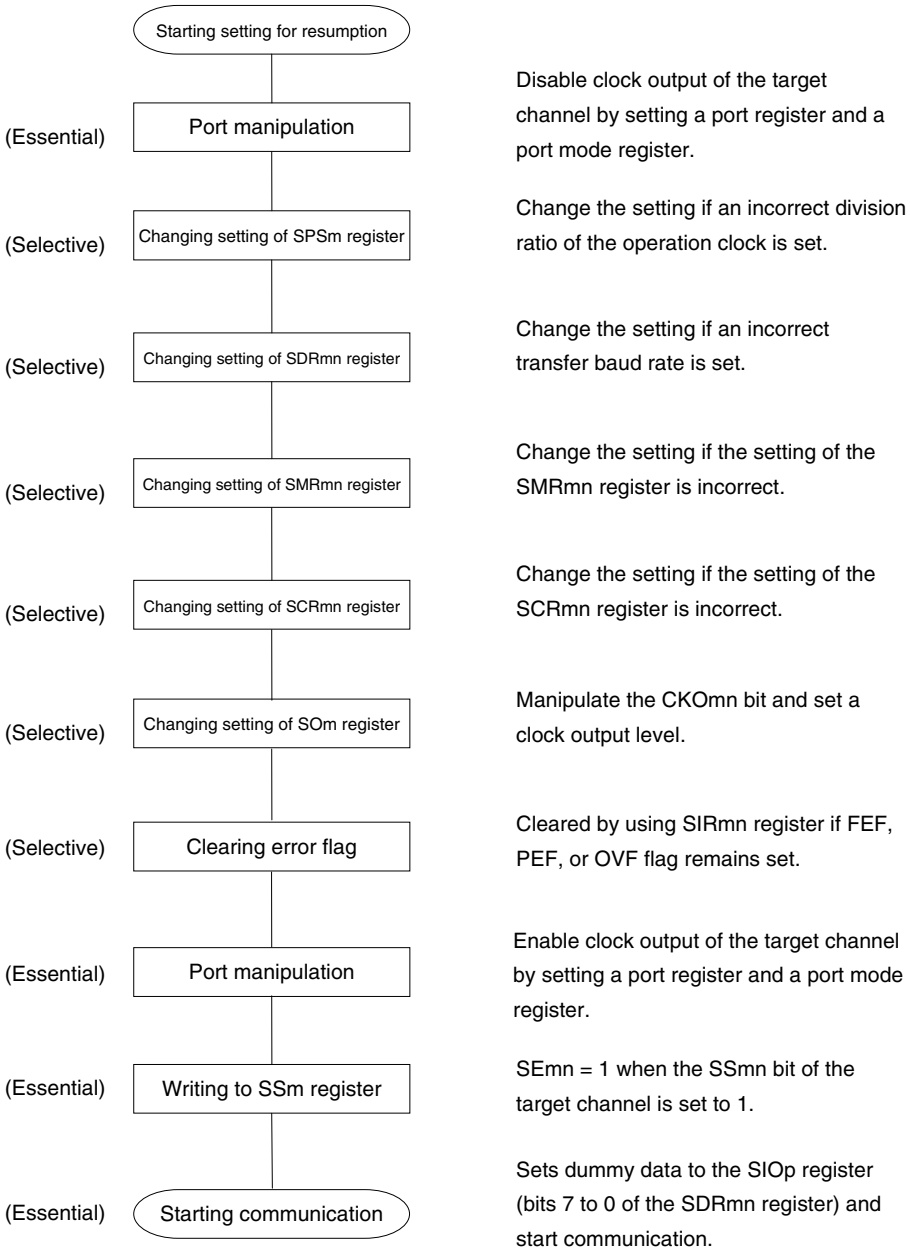
Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-35. Procedure for Stopping Master Reception



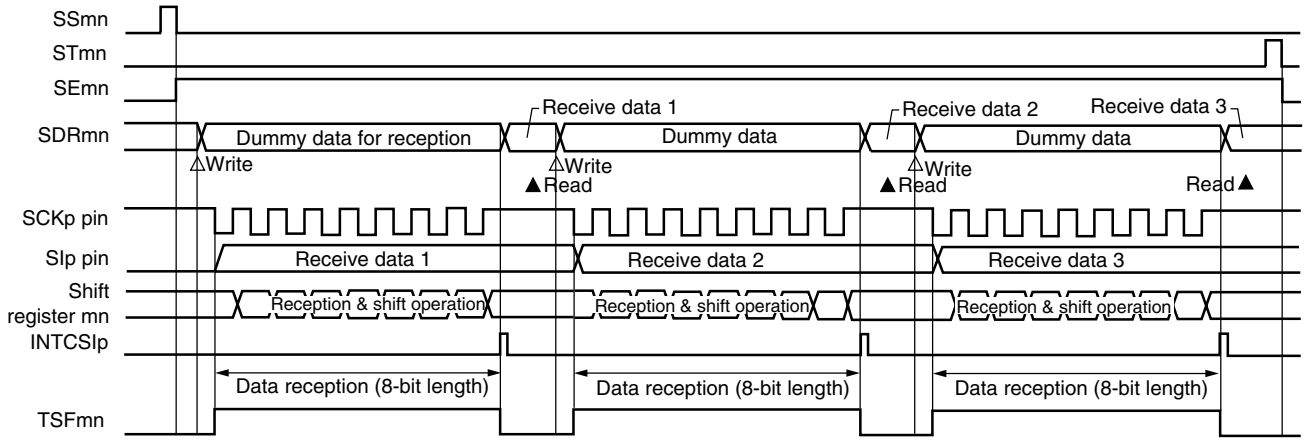
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see Figure 11-36 Procedure for Resuming Master Reception).

Figure 11-36. Procedure for Resuming Master Reception



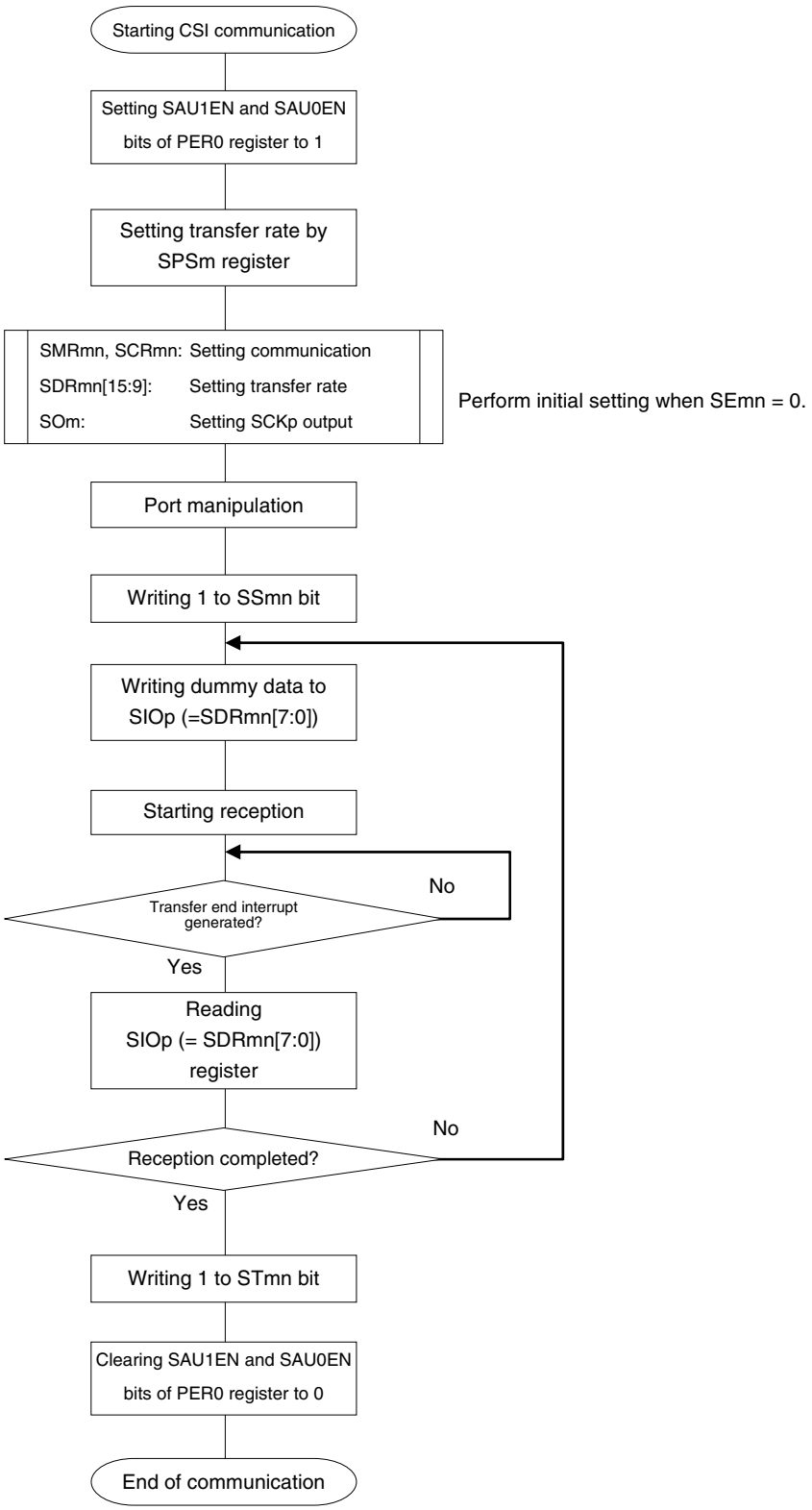
(3) Processing flow (in single-reception mode)

Figure 11-37. 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), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

Figure 11-38. Flowchart of Master Reception (in Single-Reception Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

11.5.3 Master transmission/reception

Master transmission/reception is that the 78K0R/KC3-L, KE3-L outputs a transfer clock and transmits/receives 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. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] 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		

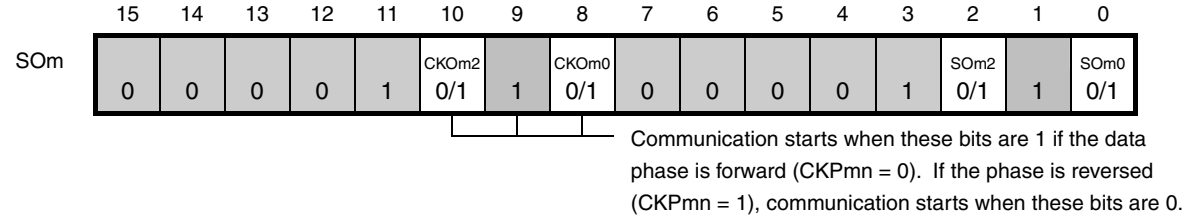
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

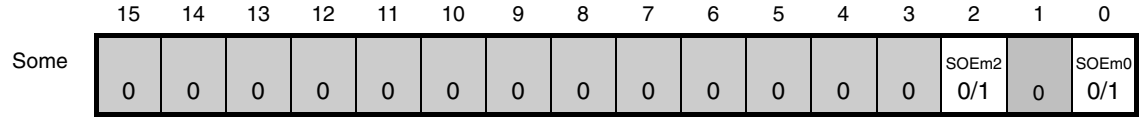
(1) Register setting

Figure 11-39. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10, CSI20) (1/2)

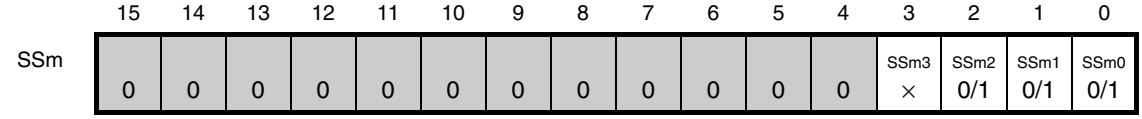
(a) Serial output register m (SOm) … Sets only the bits of the target channel.



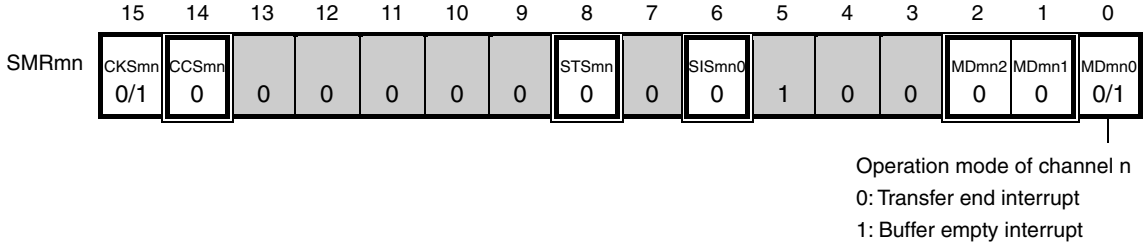
(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.



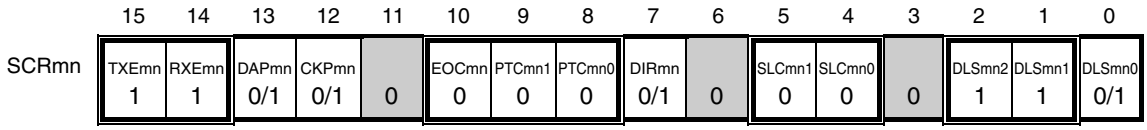
(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.



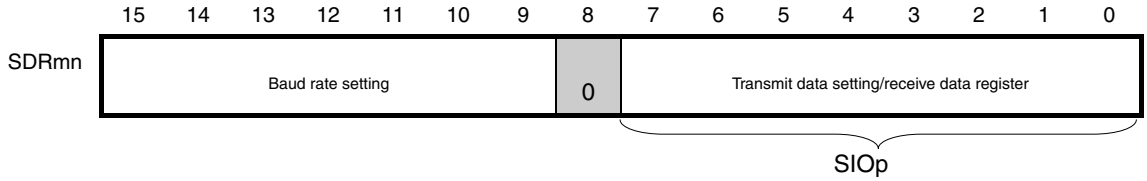
(d) Serial mode register mn (SMRmn)



(e) Serial communication operation setting register mn (SCRmn)



(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

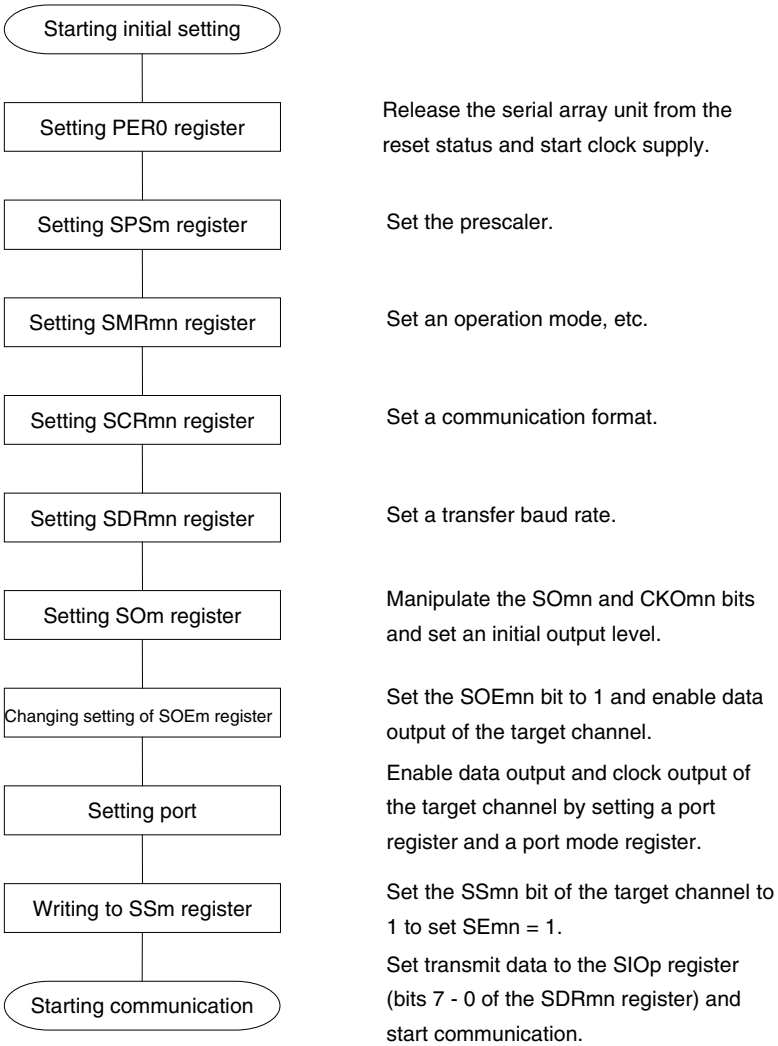


Remarks 1 m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

2 ☐ : 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

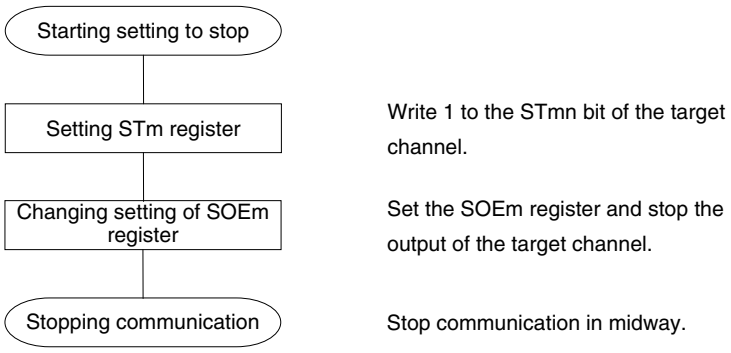
(2) Operation procedure

Figure 11-40. Initial Setting Procedure for Master Transmission/Reception



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-41. Procedure for Stopping Master Transmission/Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see Figure 11-42 Procedure for Resuming Master Transmission/Reception).

Figure 11-42. Procedure for Resuming Master Transmission/Reception

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

Set the SOEm register and stop the output of the target channel.

Manipulate the SOmn and CKOmn bits and set an initial output level.

Set the SOEm register and enable the output of the target channel.

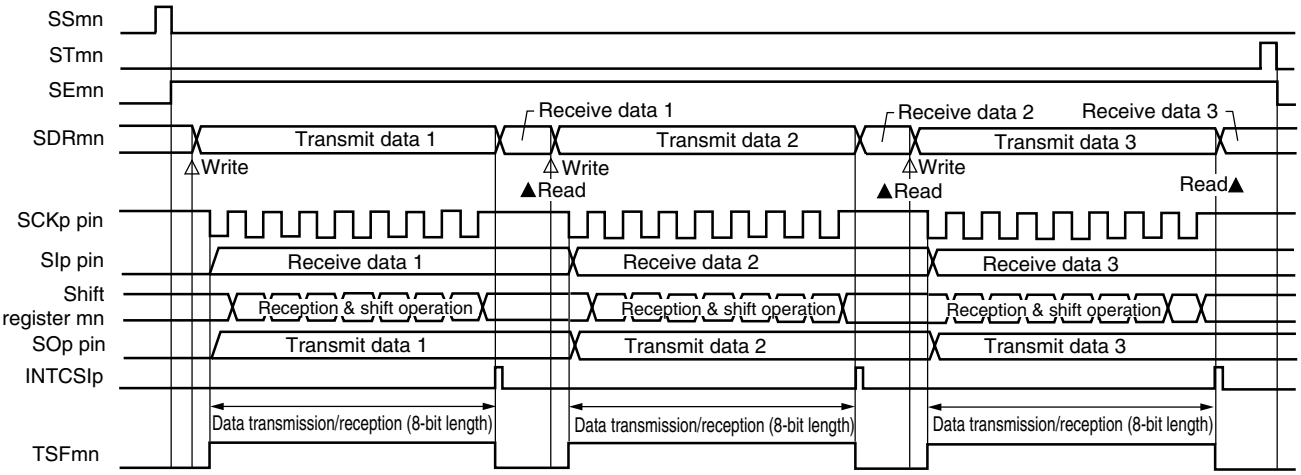
Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set SEmn to 1.

Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

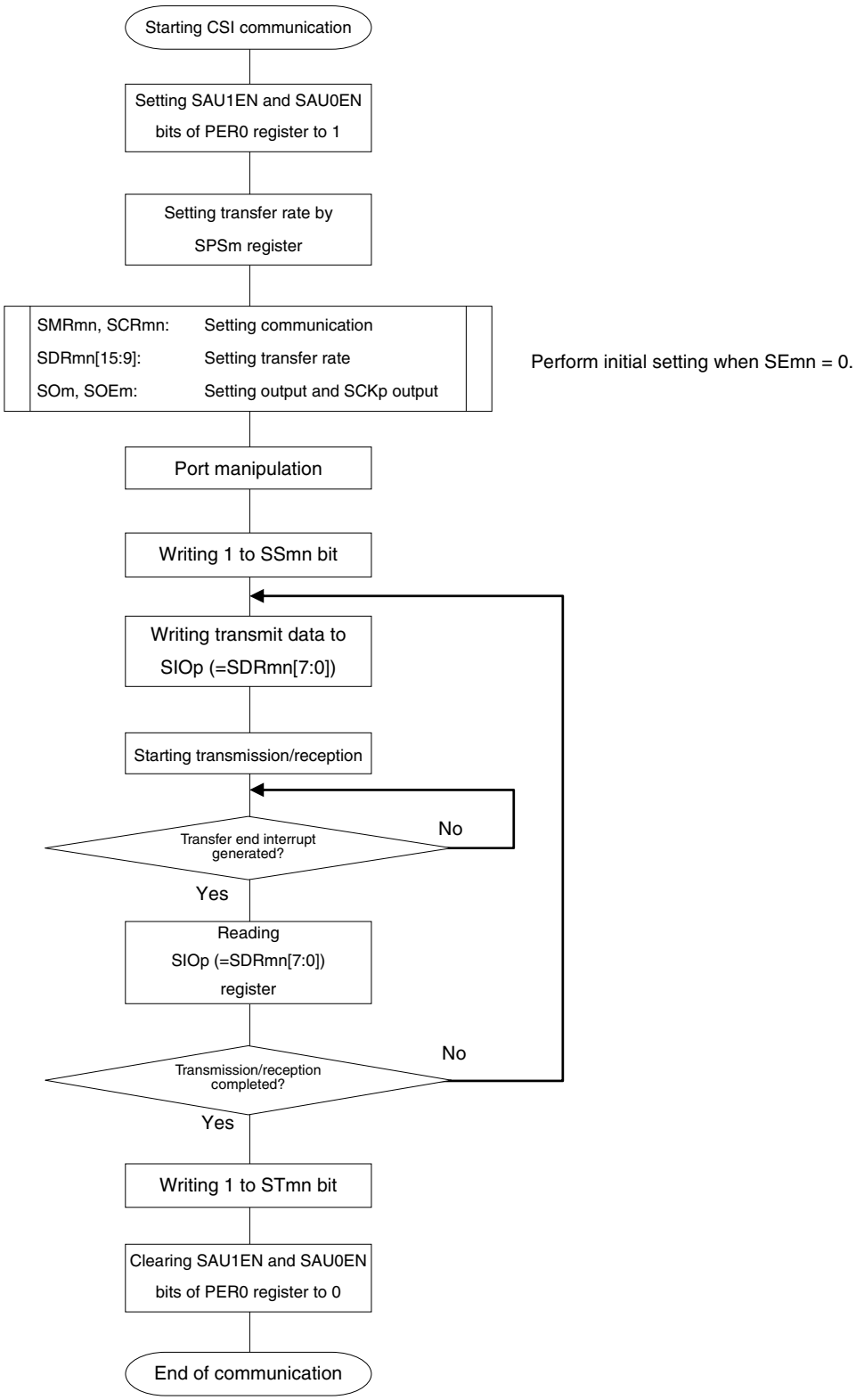
(3) Processing flow (in single-transmission/reception mode)

Figure 11-43. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

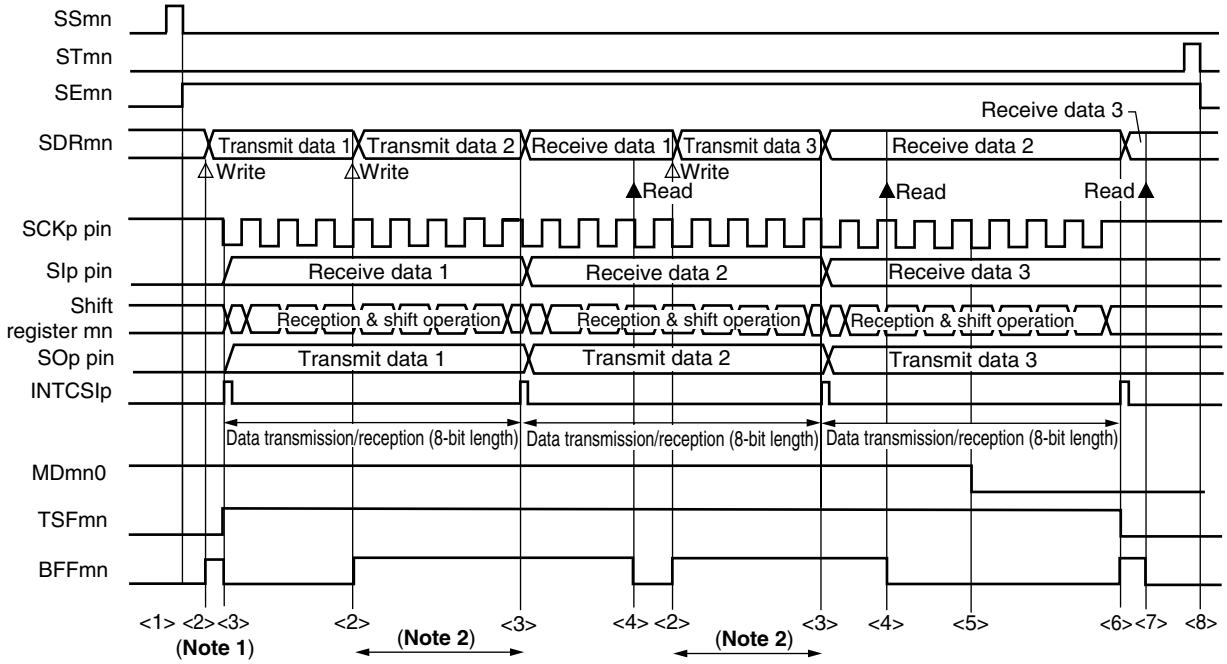
Figure 11-44. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-45. 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.

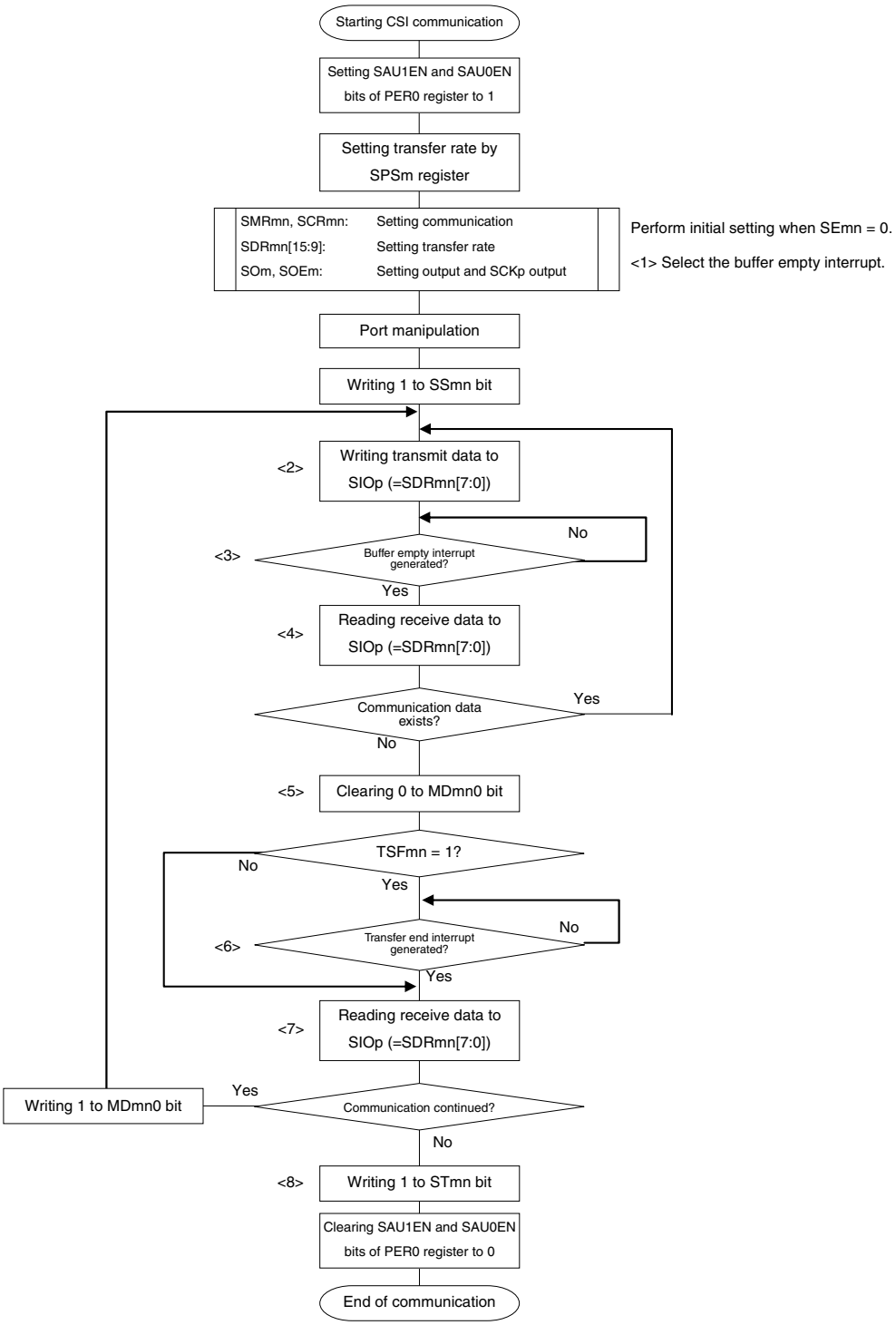
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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> - <8> in the figure correspond to <1> - <8> in Figure 11-46 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

Figure 11-46. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> - <8> in the figure correspond to <1> - <8> in Figure 11-44 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

11.5.4 Slave transmission

Slave transmission is that the 78K0R/KC3-L, KE3-L transmits 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 [Hz]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 first 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 maximum 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 28 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), mn = 00, 02, 10

(1) Register setting

Figure 11-47. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI10, CSI20) (1/2)

(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							

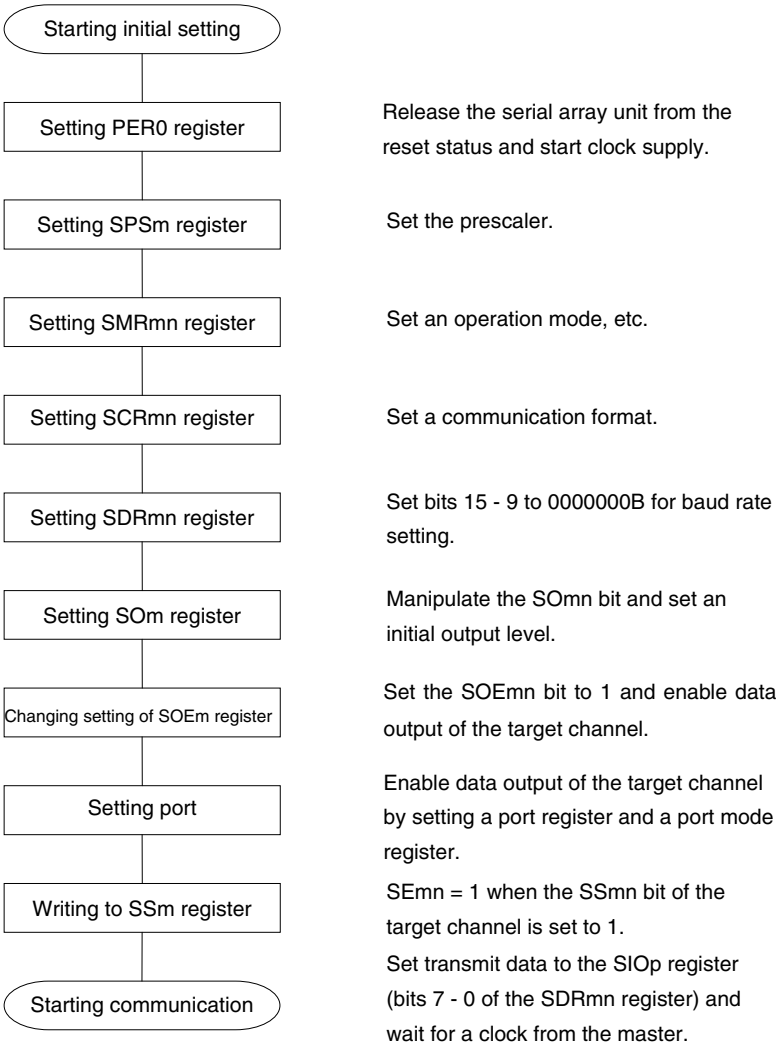
SIOp

Remark 1. **m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10**
p: CSI number (p = 00, 10, 20)

2. **□: 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

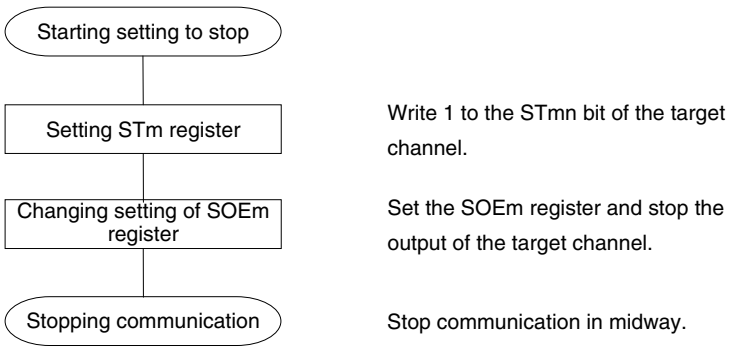
(2) Operation procedure

Figure 11-48. Initial Setting Procedure for Slave Transmission



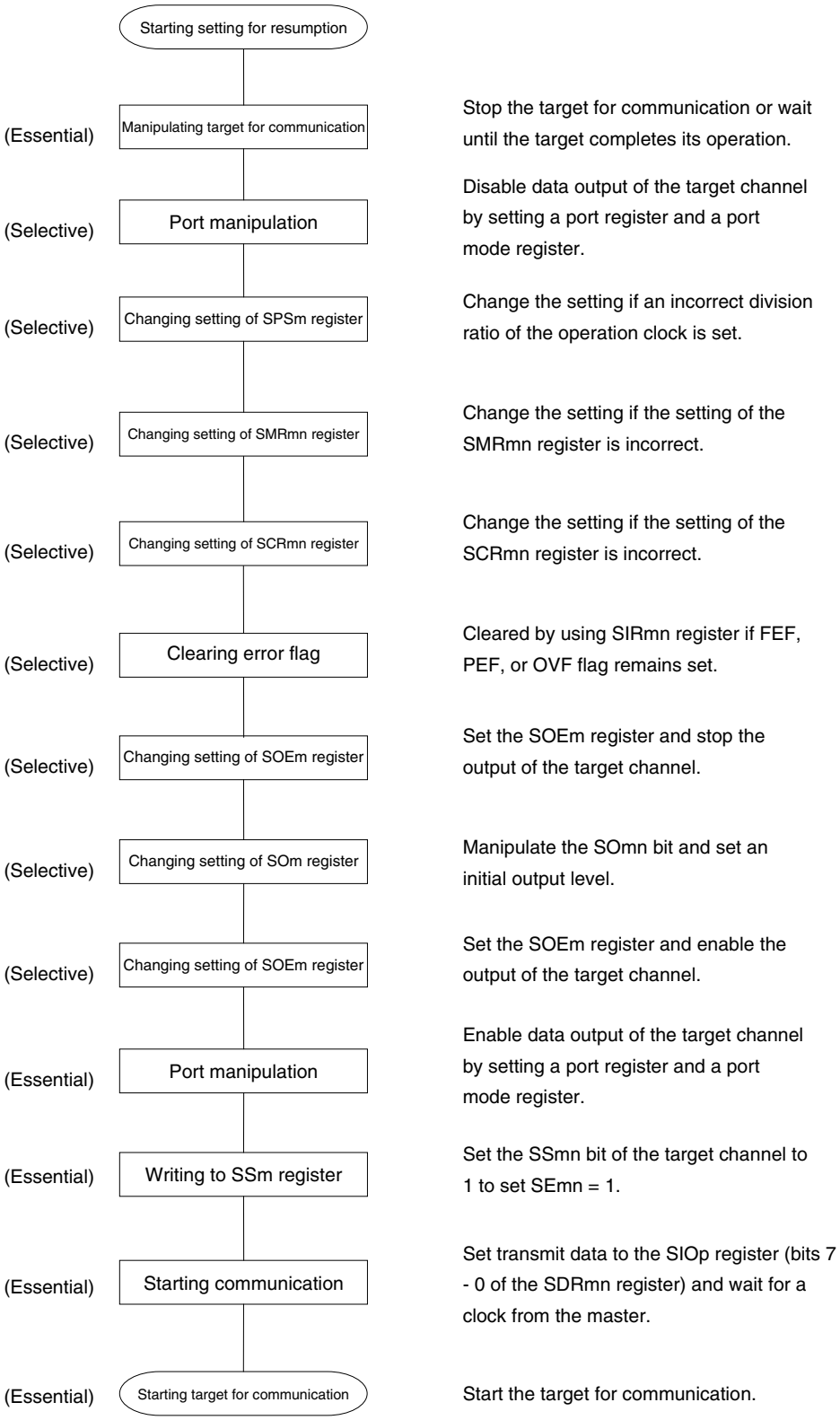
Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-49. Procedure for Stopping Slave Transmission



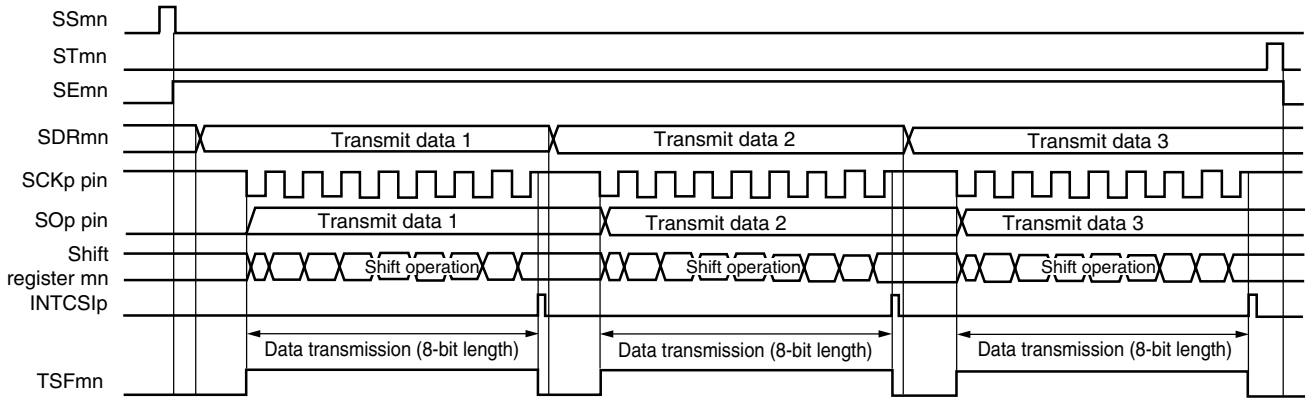
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see Figure 11-49 Procedure for Resuming Slave Transmission).

Figure 11-50. Procedure for Resuming Slave Transmission



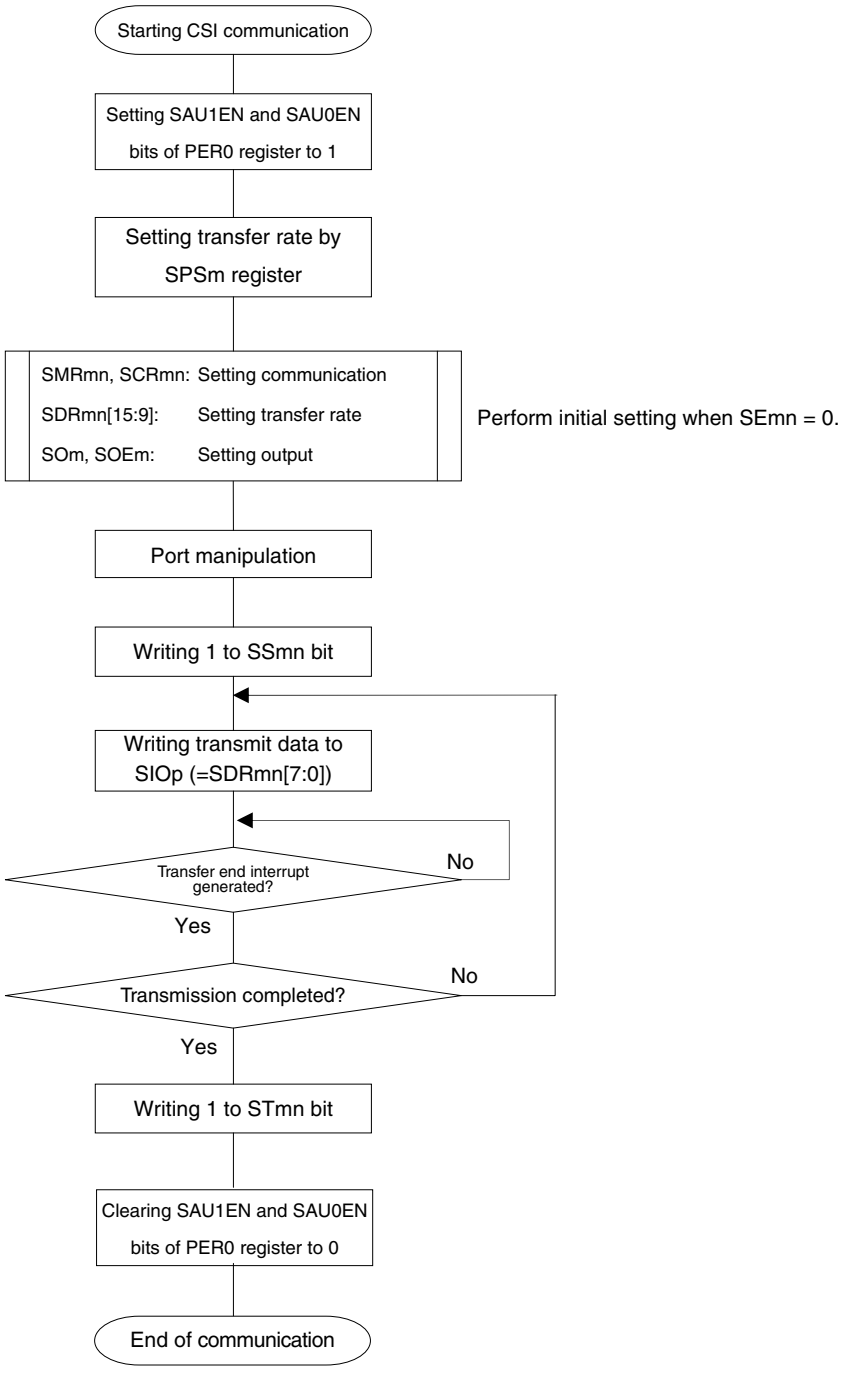
(3) Processing flow (in single-transmission mode)

Figure 11-51. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

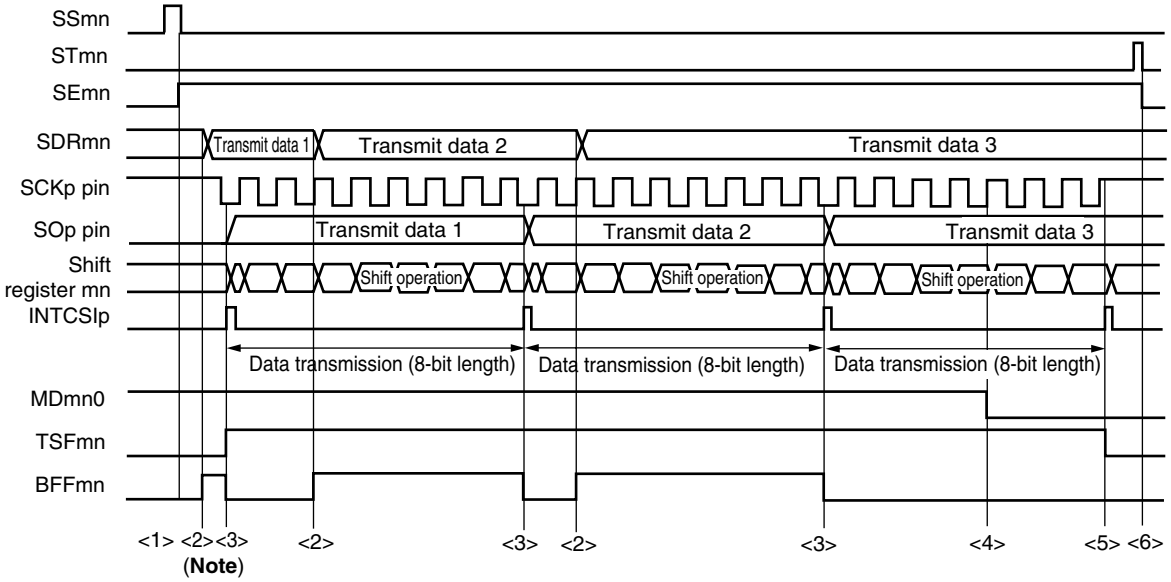
Figure 11-52. Flowchart of Slave Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

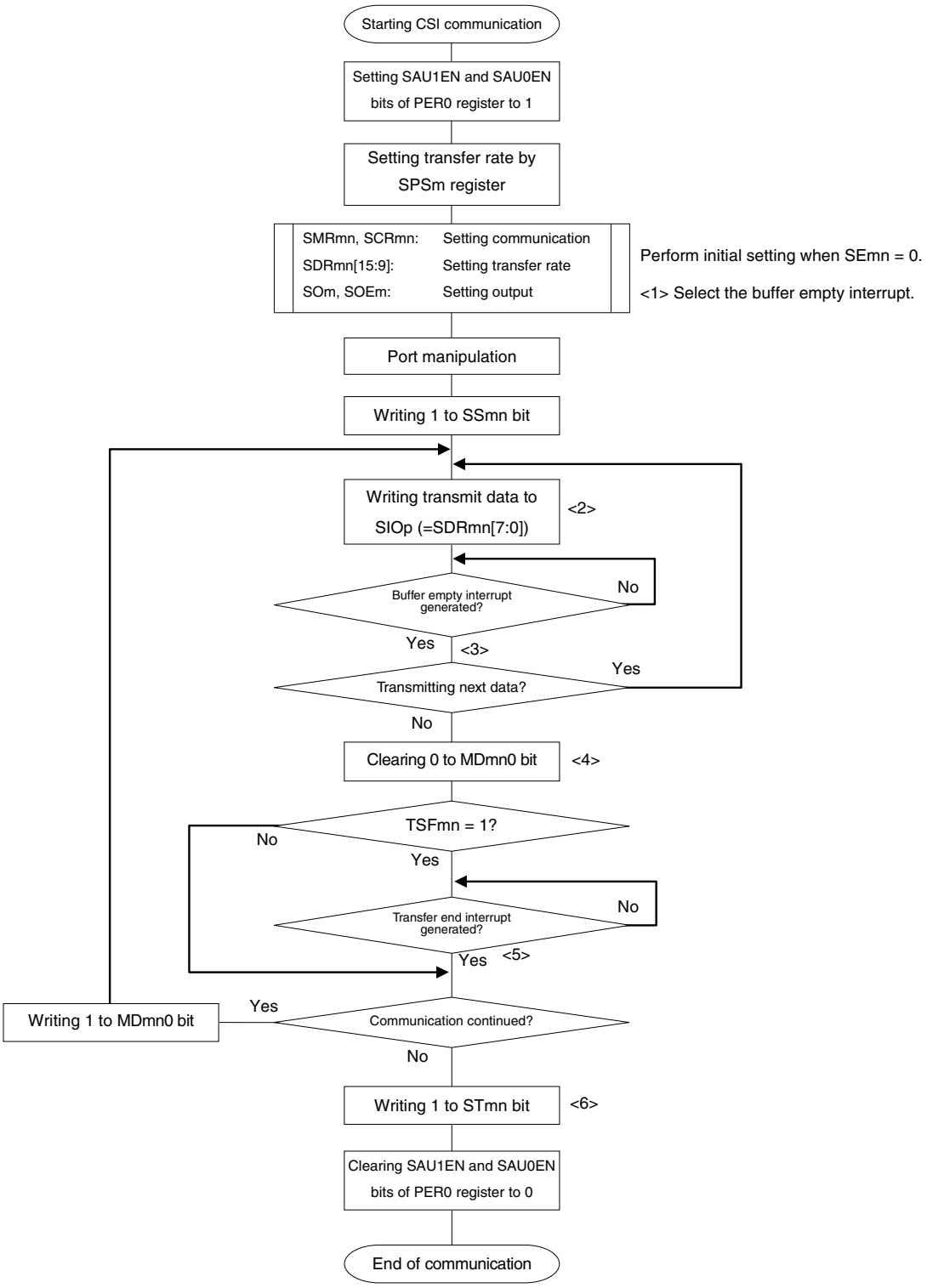
Figure 11-53. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



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.

Figure 11-54. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> - <6> in the figure correspond to <1> - <6> in Figure 11-52 Timing Chart of Slave Transmission (in Continuous Transmission Mode).

11.5.5 Slave reception

Slave reception is that the 78K0R/KC3-L, KE3-L receives 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 [Hz]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 maximum 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 28 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), mn = 00, 02, 10

(1) Register setting

Figure 11-55. 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)

	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 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							

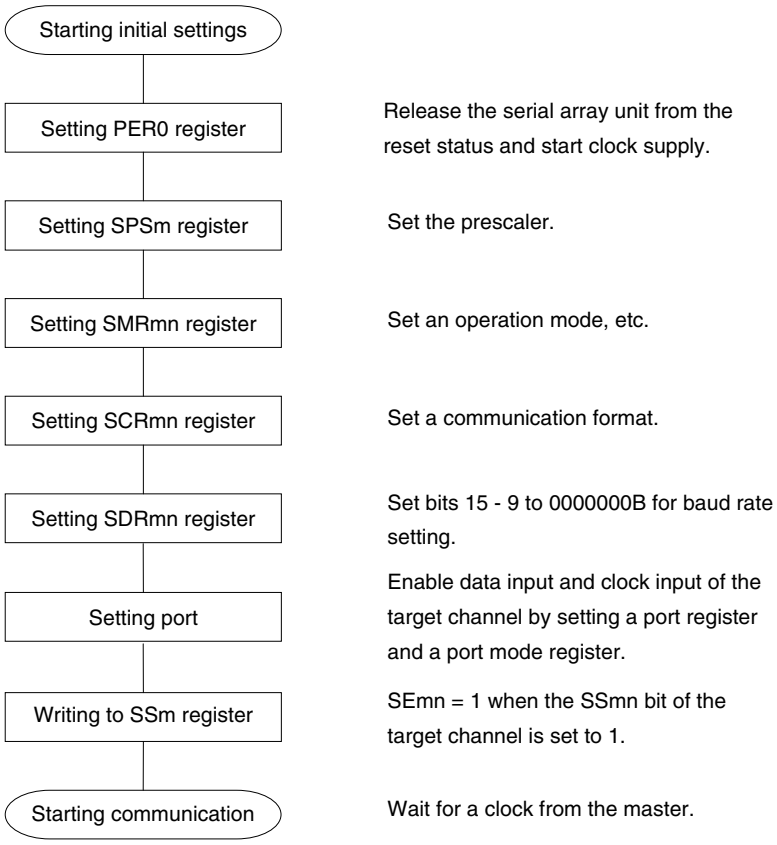
SIOp

Remark 1. **m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10**
p: CSI number (p = 00, 10, 20)

2. **□ : 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 11-56. Initial Setting Procedure for Slave Reception



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-57. Procedure for Stopping Slave Reception

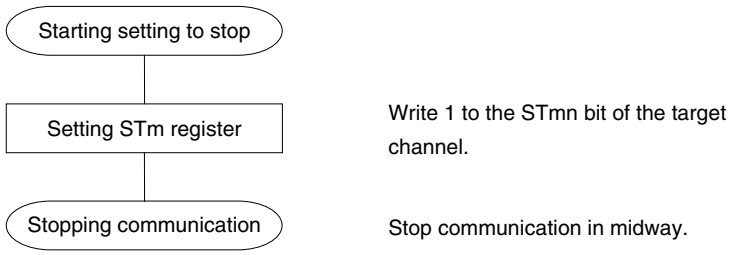
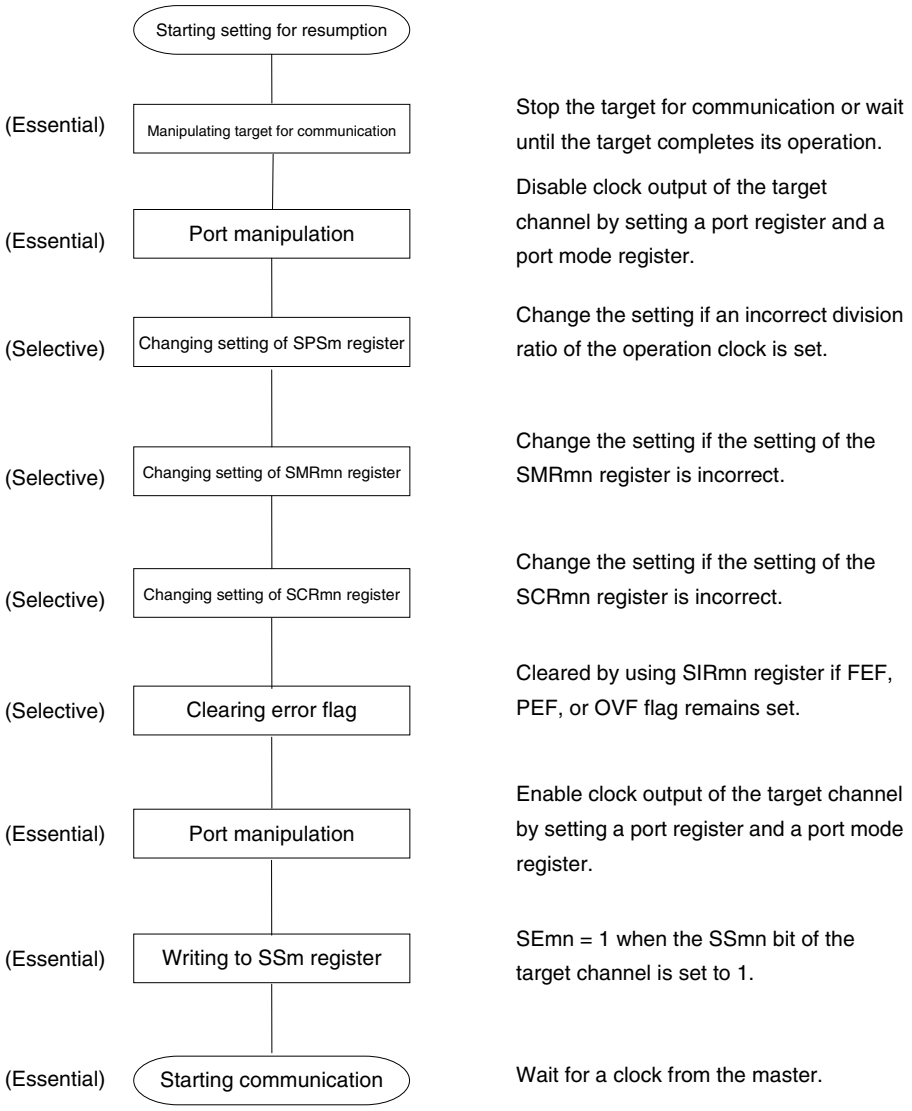
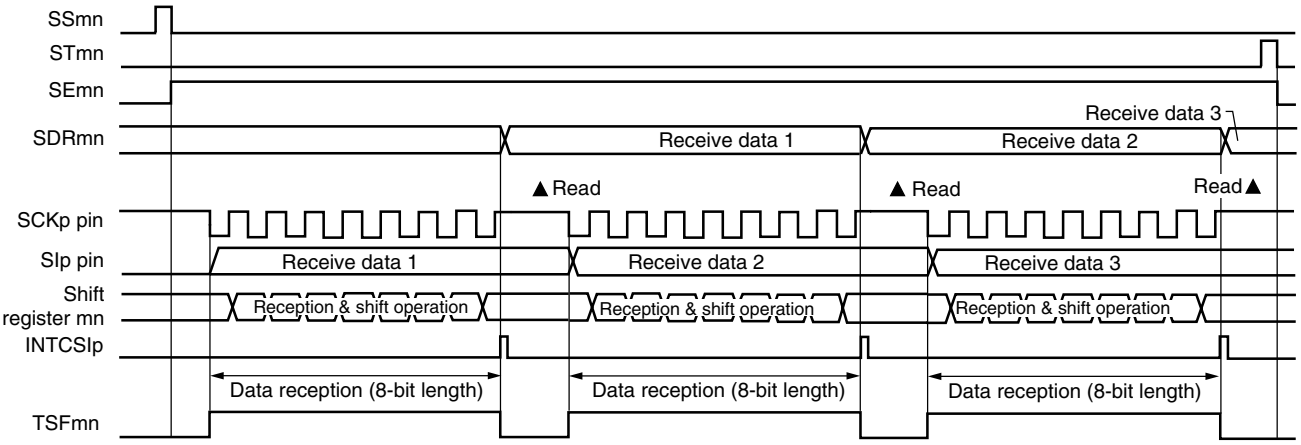


Figure 11-58. Procedure for Resuming Slave Reception



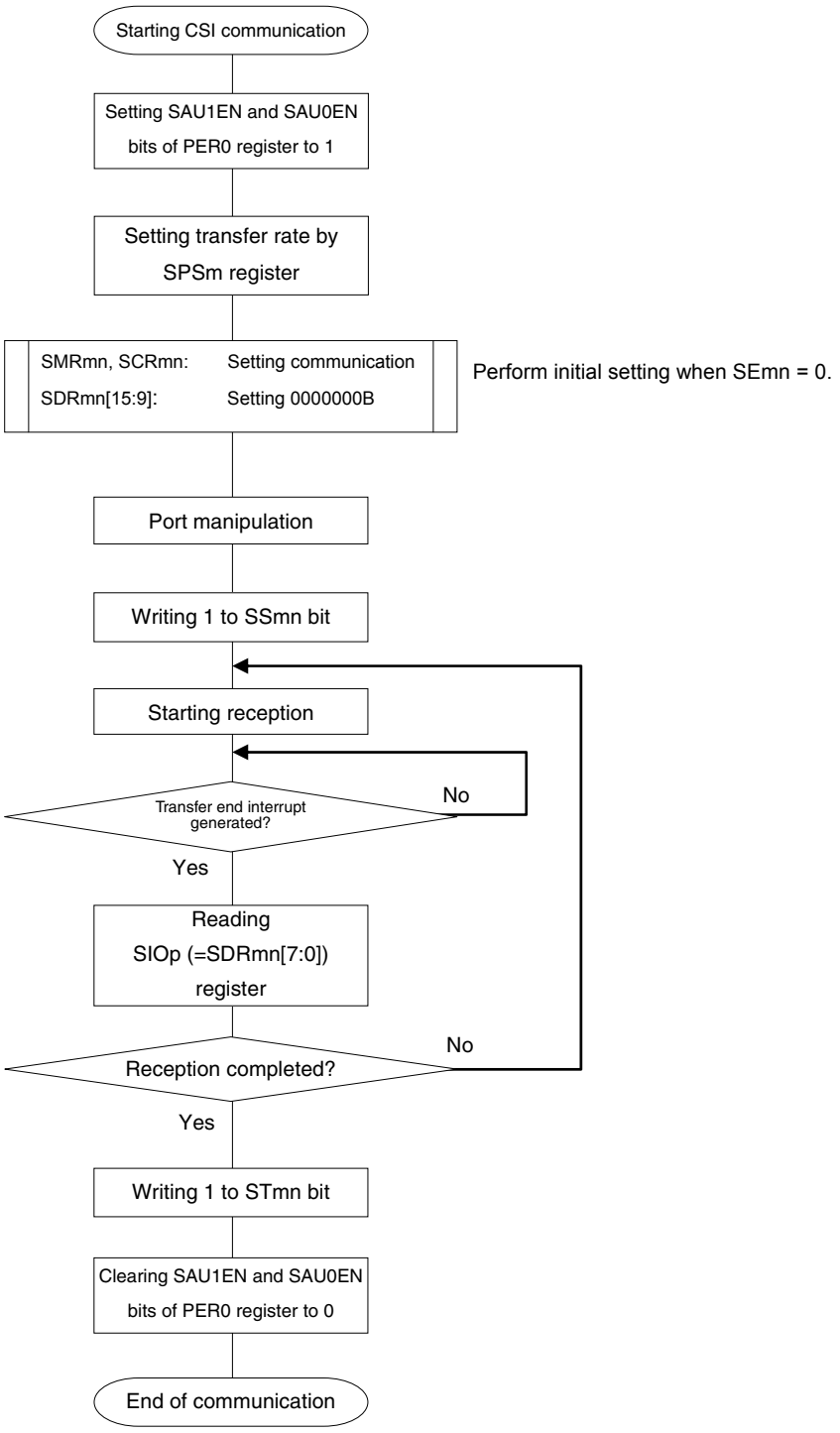
(3) Processing flow (in single-reception mode)

Figure 11-59. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

Figure 11-60. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

11.5.6 Slave transmission/reception

Slave transmission/reception is that the 78K0R/KC3-L, KE3-L transmits/receives 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 [Hz]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 $\overline{SCK00}$, $\overline{SCK01}$, and $\overline{SCK20}$ is sampled internally and used, the maximum 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 28 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), mn = 00, 02, 10

(1) Register setting

Figure 11-61. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10, CSI20) (1/2)

(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	SOEm1 0/1	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							

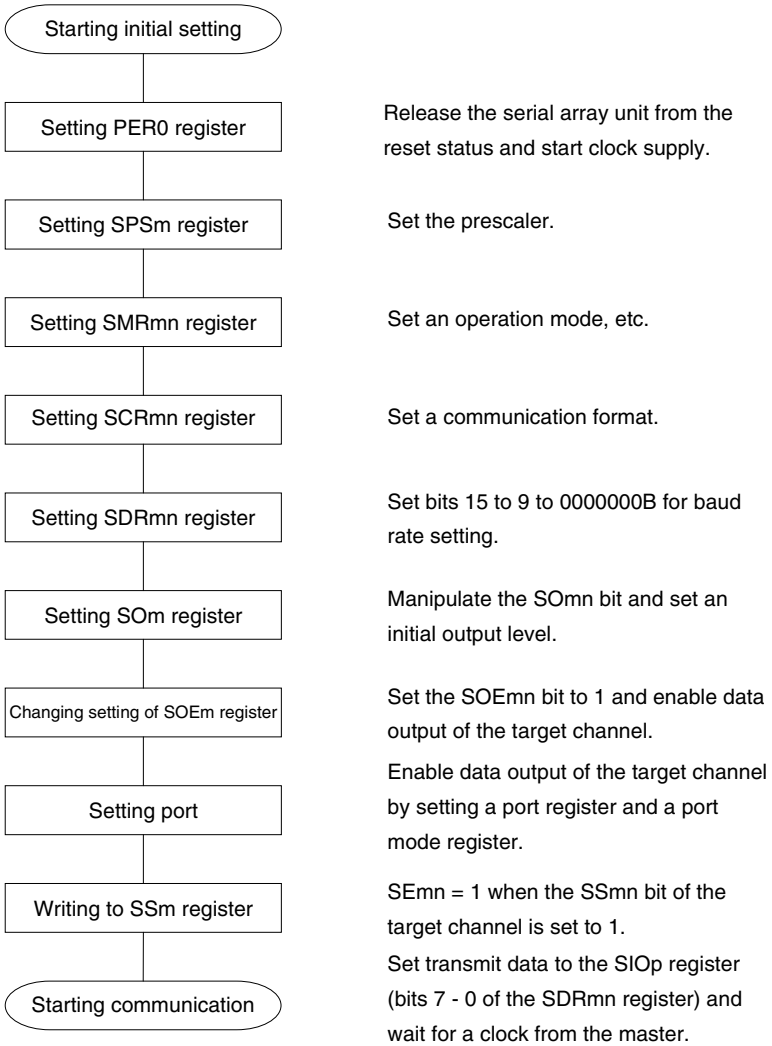
SIOp

Remark 1. **m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10**
p: CSI number (p = 00, 10, 20)

2. **□: 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

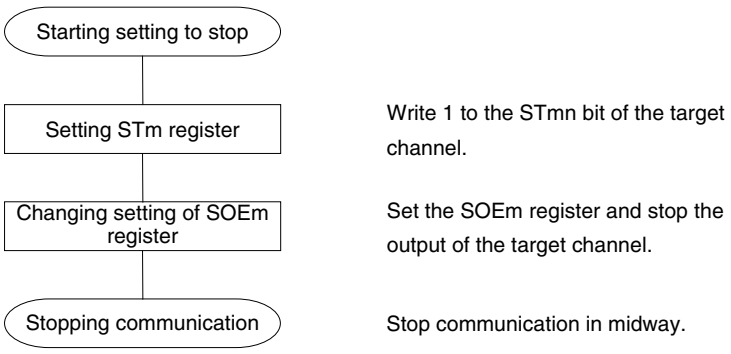
(2) Operation procedure

Figure 11-62. Initial Setting Procedure for Slave Transmission/Reception



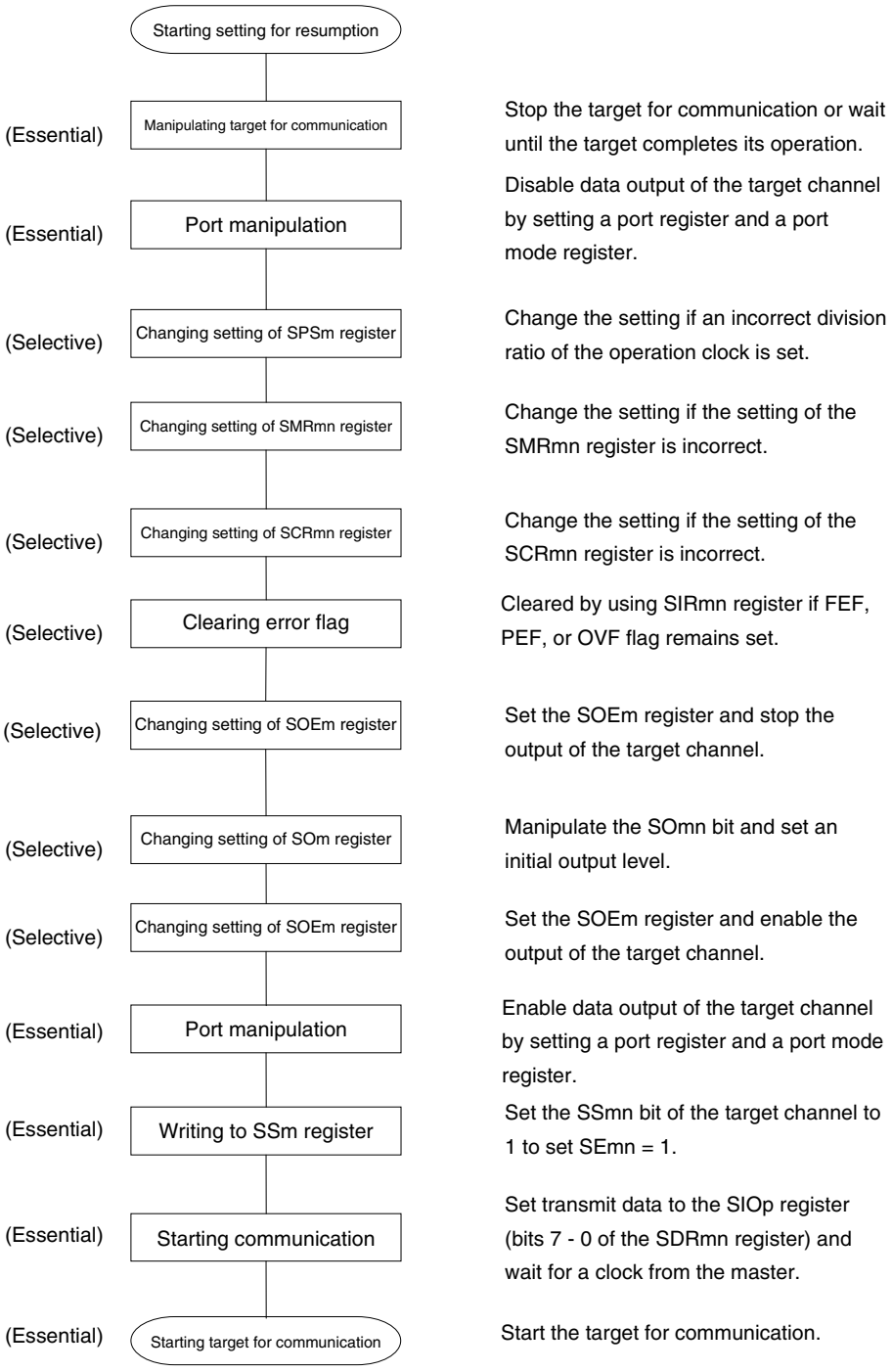
Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-63. Procedure for Stopping Slave Transmission/Reception



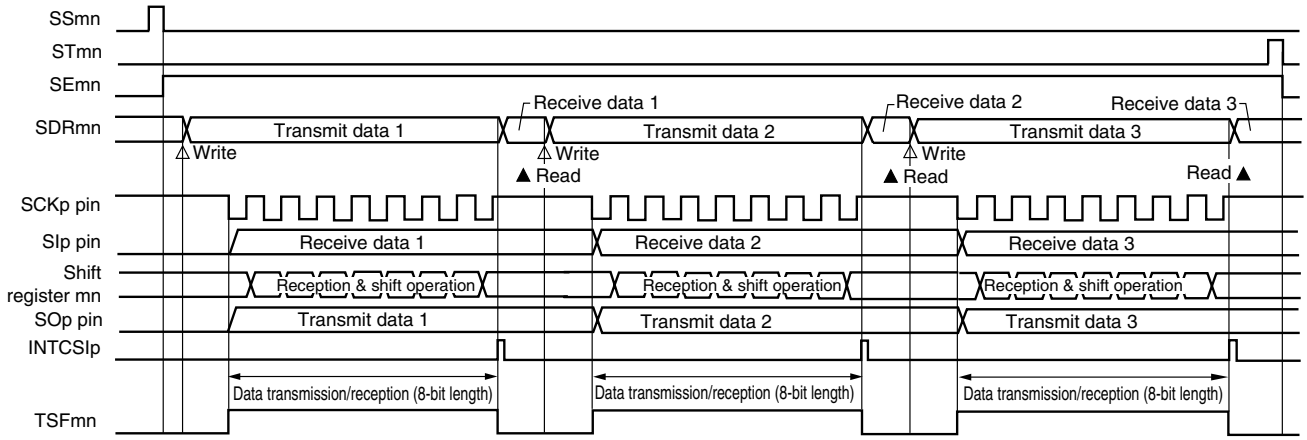
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see Figure 11-64 Procedure for Resuming Slave Transmission/Reception).

Figure 11-64. Procedure for Resuming Slave Transmission/Reception



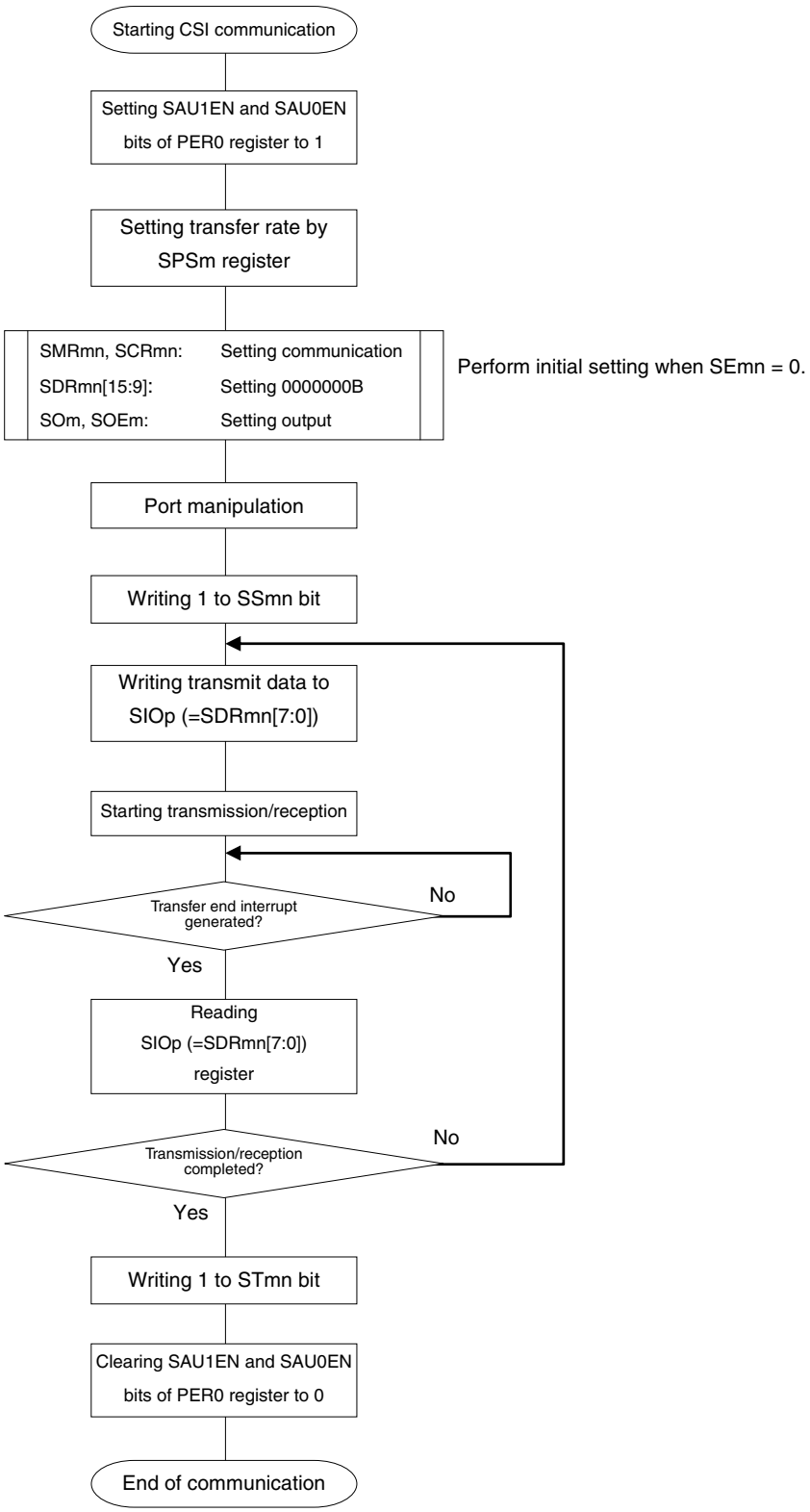
(3) Processing flow (in single-transmission/reception mode)

Figure 11-65. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

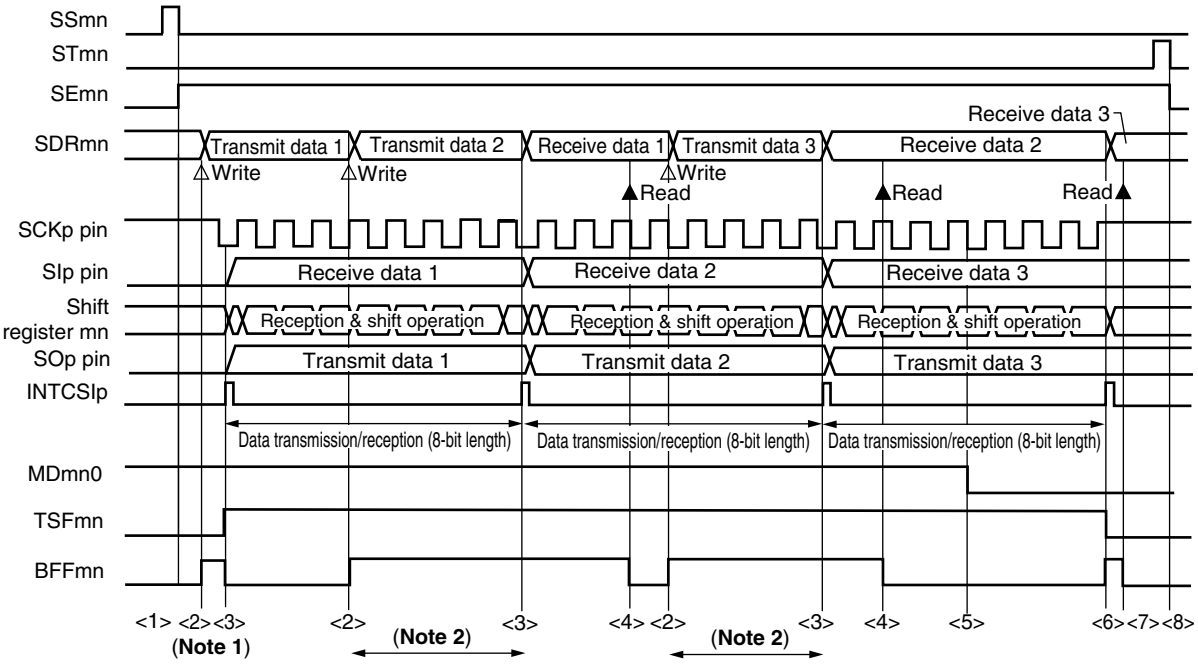
Figure 11-66. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-67. 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.

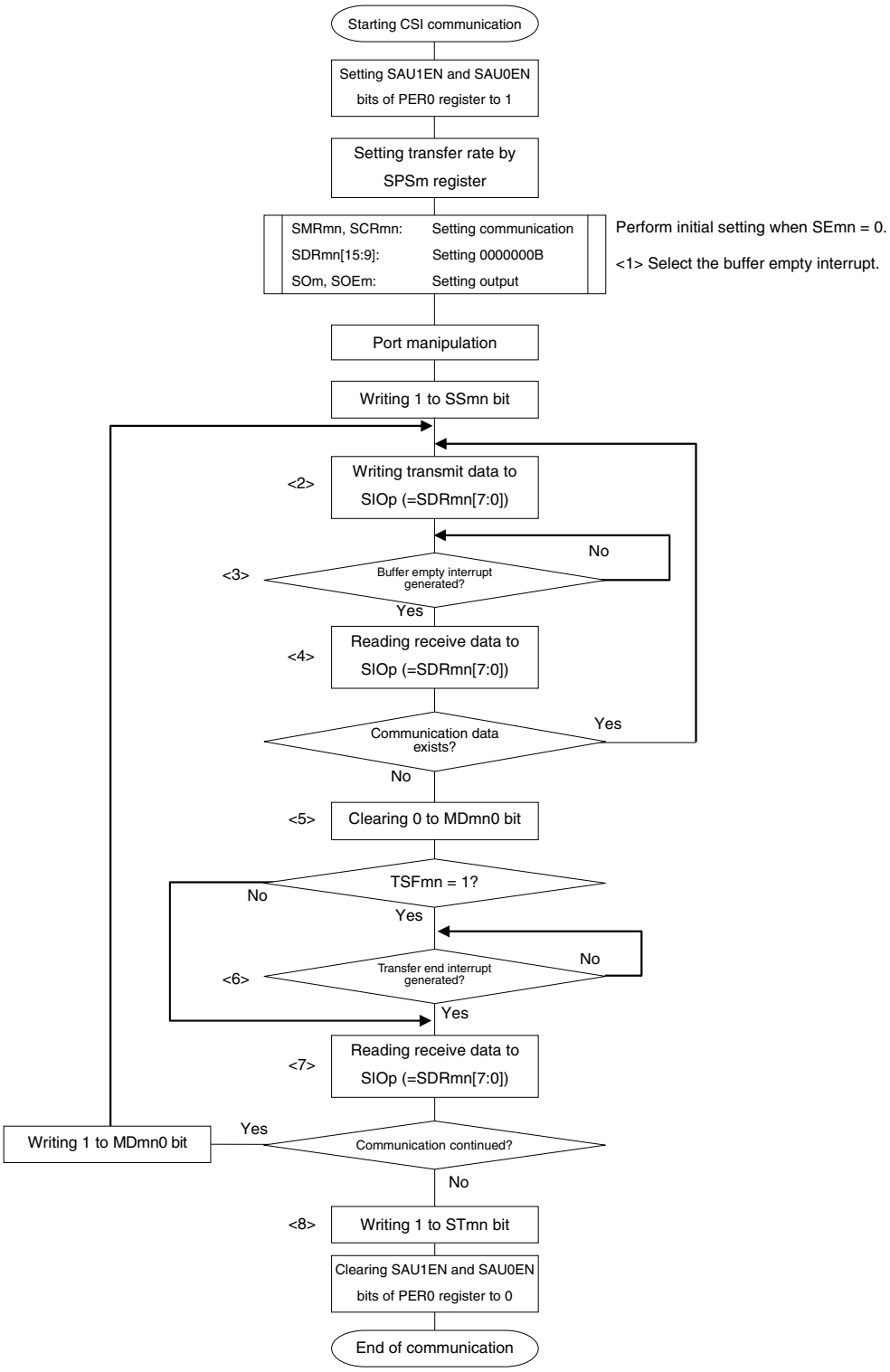
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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> - <8> in the figure correspond to <1> - <8> in Figure 11-68 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

Figure 11-68. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> - <8> in the figure correspond to <1> - <8> in Figure 11-67 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

11.5.7 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 (SCK) supplied by master}[Note] [Hz]

Note The permissible maximum transfer clock frequency is f_{MCK}/6.

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), mn = 00, 02, 10

p: CSI number (p = 00, 10, 20)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-2. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK) Note 1			
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 12 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	12 MHz	16 MHz	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	6 MHz	8 MHz	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	3 MHz	4 MHz	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	1.5 MHz	2 MHz	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	0.75 MHz	1 MHz	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	375 kHz	500 kHz	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	187.5 kHz	250 kHz	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	93.75 kHz	125 kHz	156.25 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	46.88 kHz	62.5 kHz	78.13 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	23.44 kHz	31.25 kHz	39.06 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	11.72 kHz	15.63 kHz	19.53 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	5.86 kHz	7.81 kHz	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0 Note 2. In the case of m = 1, setting is prohibited.			
1	0	0	0	0	X	X	X	X	f_{CLK}	12 MHz	16 MHz	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	6 MHz	8 MHz	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	3 MHz	4 MHz	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	1.5 MHz	2 MHz	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	0.75 MHz	1 MHz	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	375 kHz	500 kHz	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	187.5 kHz	250 kHz	312.5 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	93.75 kHz	125 kHz	156.25 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	46.88 kHz	62.5 kHz	78.13 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	23.44 kHz	31.25 kHz	39.06 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	11.72 kHz	15.63 kHz	19.53 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	5.86 kHz	7.81 kHz	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0 Note 2. In the case of m = 1, setting is prohibited.			
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 (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, sub system clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting the TIS02 bit of timer input select register 0 (TIS0) to 1) and selecting INTTM02 by using the SPS0 register in channel 2 of TAU0. When changing f_{CLK}, however, SAU0 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), mn = 00, 02, 10
p: CSI number (p = 00, 10, 20)

11.6 Operation of UART (UART0 - UART3) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit 0 with an external interrupt (INTP0).

[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 (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.

< In case of 78K0R/KC3-L >

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	–	–	–
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

< In case of 78K0R/KC3-L >

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	IIC10
	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 **11.6.1**.)
- UART reception (See **11.6.2**.)
- LIN transmission (UART3 only) (See **11.6.3**.)
- LIN reception (UART 3 only) (See **11.6.4**.)

11.6.1 In this chapter related to control register, 11.2 Configuration of Serial Array Unit of 78K0R/KE3-L has been taken as an example and explained. Refer it for the difference between configuration of Serial Array Unit of KC3-L and KE3-L .

11.6.1 UART transmission

UART transmission is an operation to transmit data from the 78K0R/KC3-L, KE3-L 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 first or LSB first			

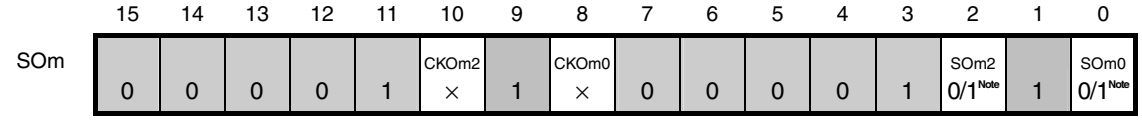
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see CHAPTER 28 ELECTRICAL SPECIFICATIONS).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency
2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

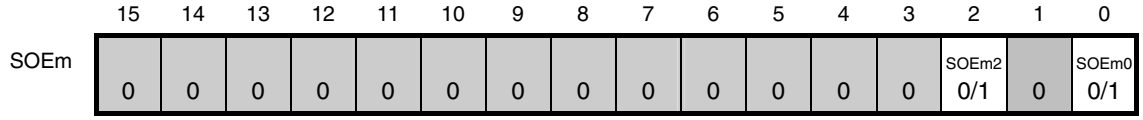
(1) Register setting

Figure 11-69. Example of Contents of Registers for UART Transmission of UART (UART0 to UART3) (1/2)

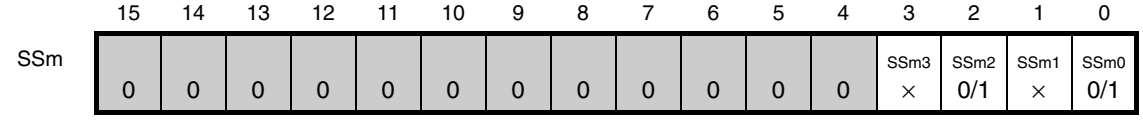
(a) Serial output register m (SOm) … Sets only the bits of the target channel to 1.



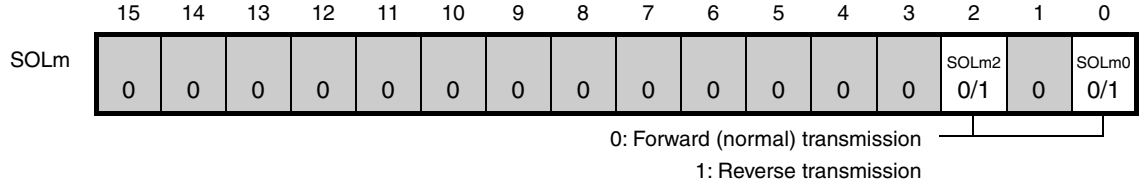
(b) Serial output enable register m (SOEm)



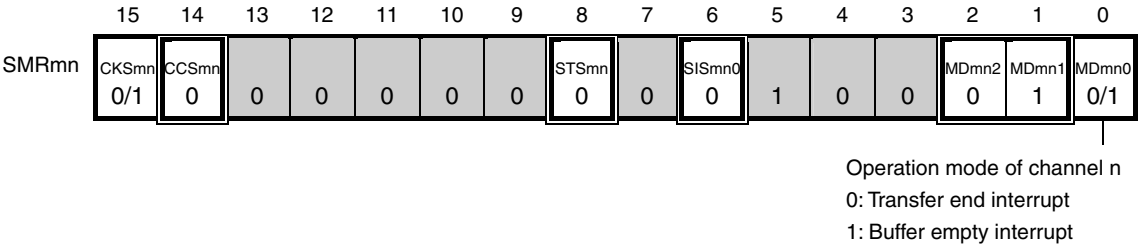
(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.



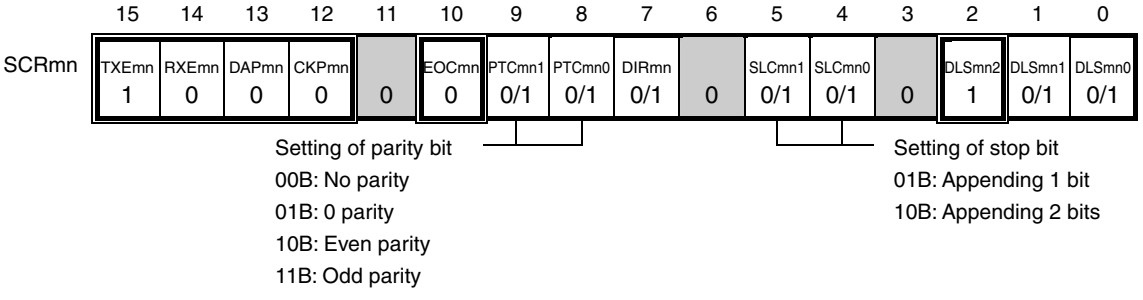
(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.



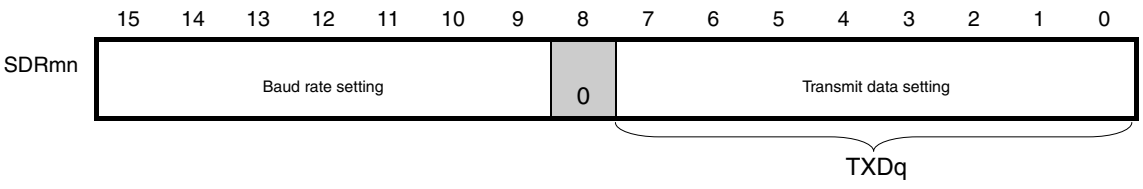
(e) Serial mode register mn (SMRmn)



(f) Serial communication operation setting register mn (SCRmn)



(g) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



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.

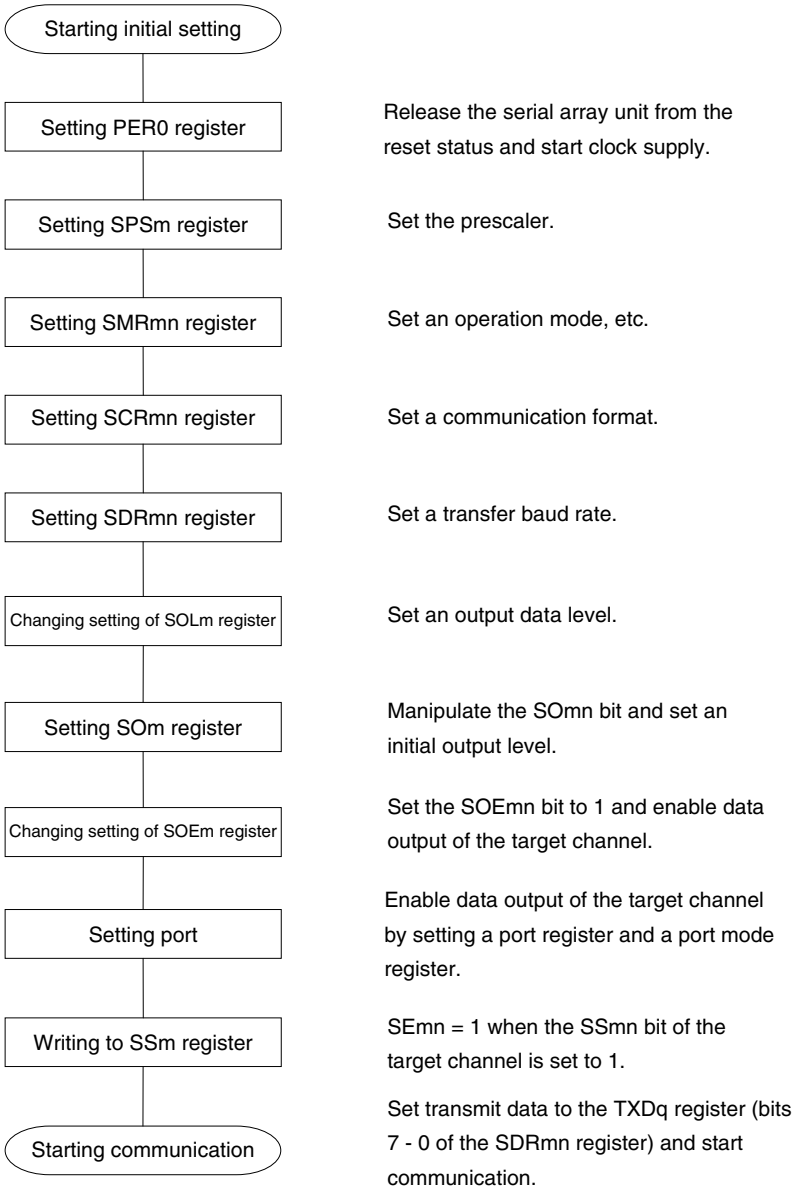
Figure 11-69. Example of Contents of Registers for UART Transmission of UART (UART0 to UART3) (2/2)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12
q: UART number (q = 0 to 3)

2. □ : Setting is fixed in the UART transmission mode, ■ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

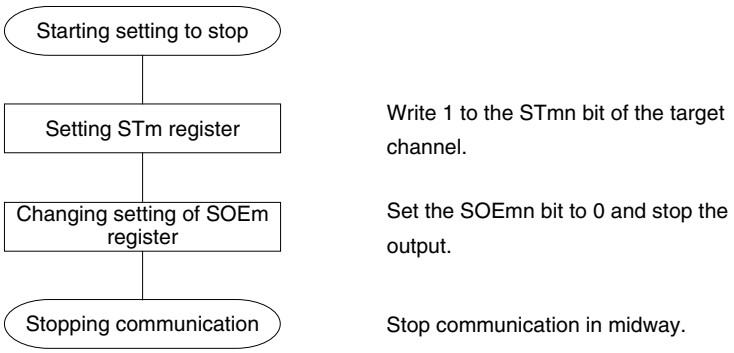
(2) Operation procedure

Figure 11-70. Initial Setting Procedure for UART Transmission



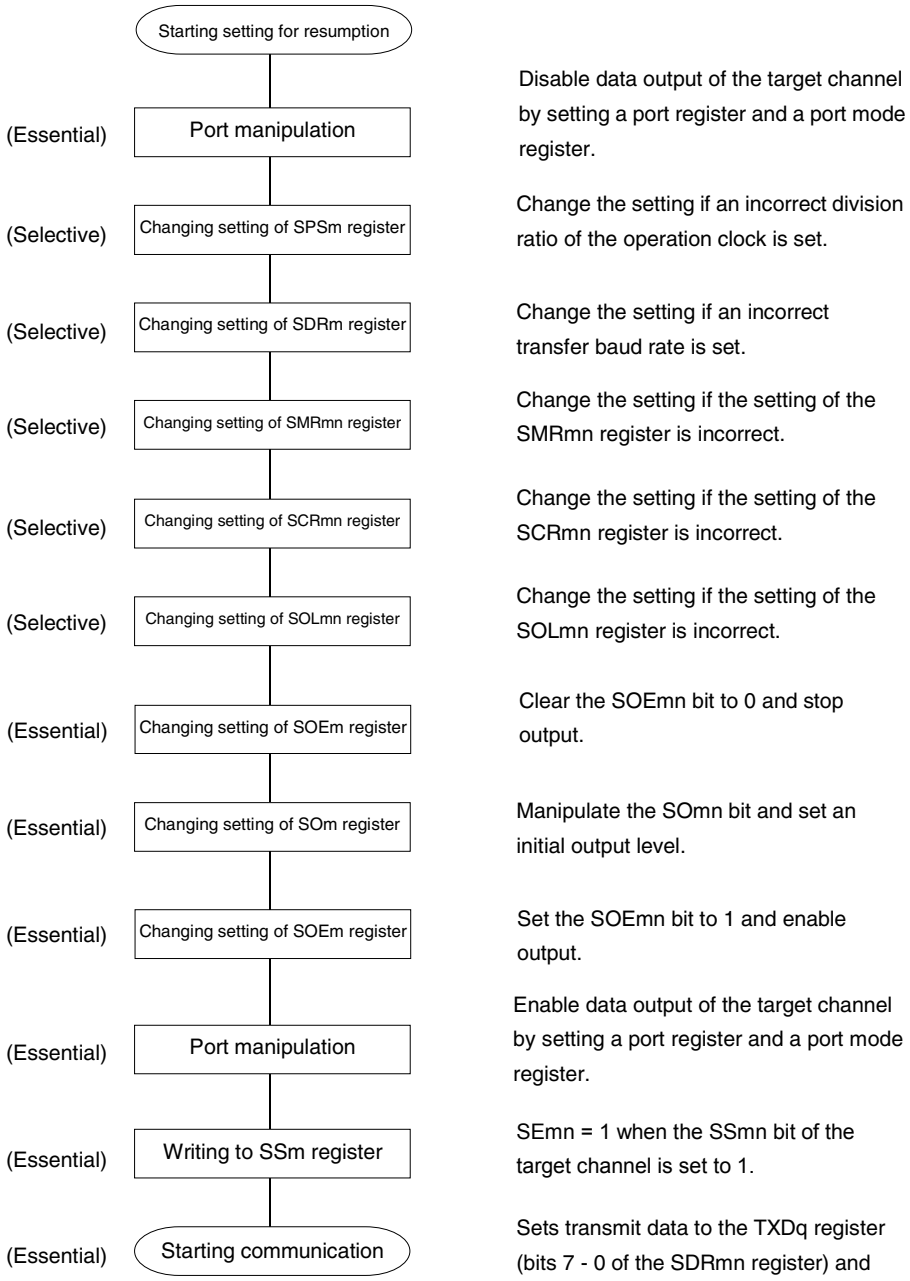
Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-71. Procedure for Stopping UART Transmission



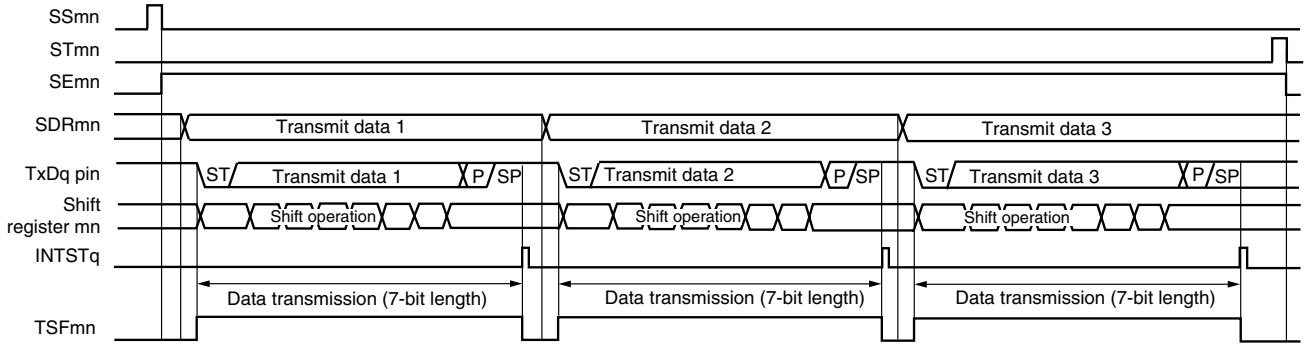
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see Figure 11-72 Procedure for Resuming UART Transmission).

Figure 11-72. Procedure for Resuming UART Transmission



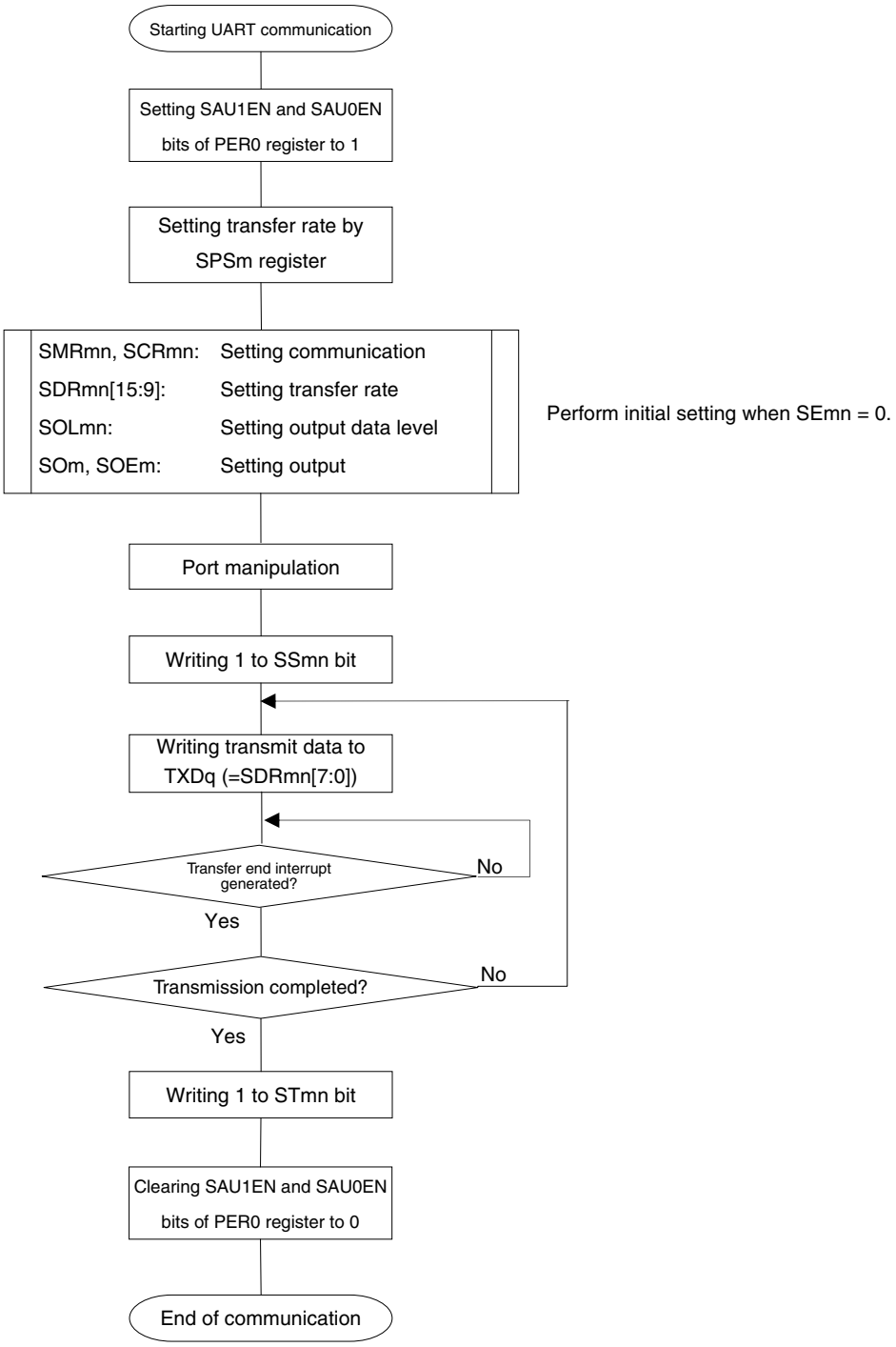
(3) Processing flow (in single-transmission mode)

Figure 11-73. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 - 2), mn = 00, 02, 10, 12

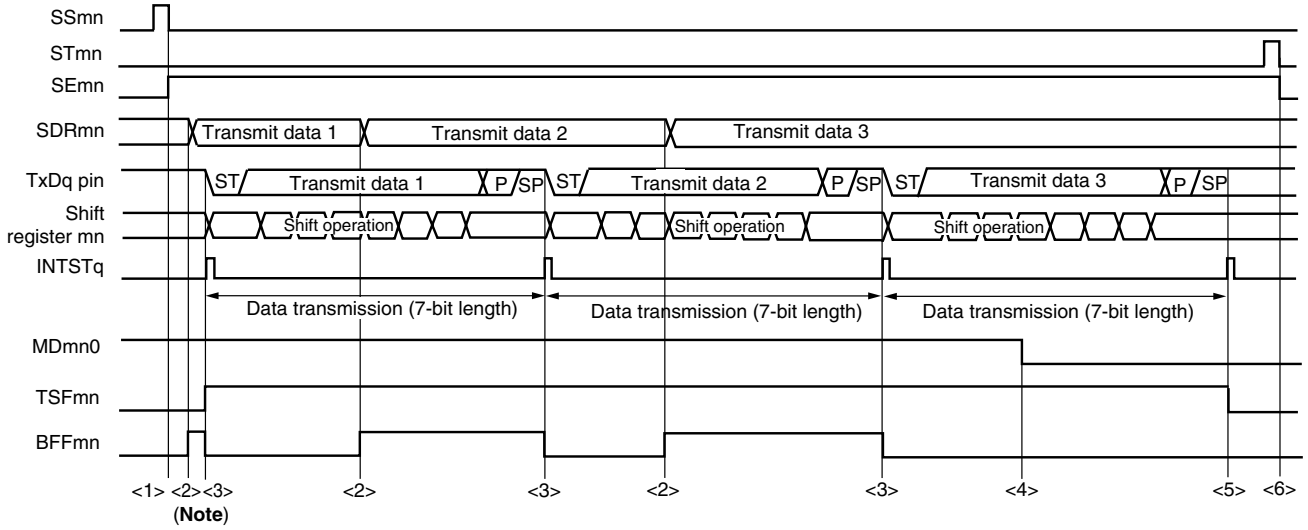
Figure 11-74. Flowchart of UART Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-75. Timing Chart of UART Transmission (in Continuous Transmission Mode)

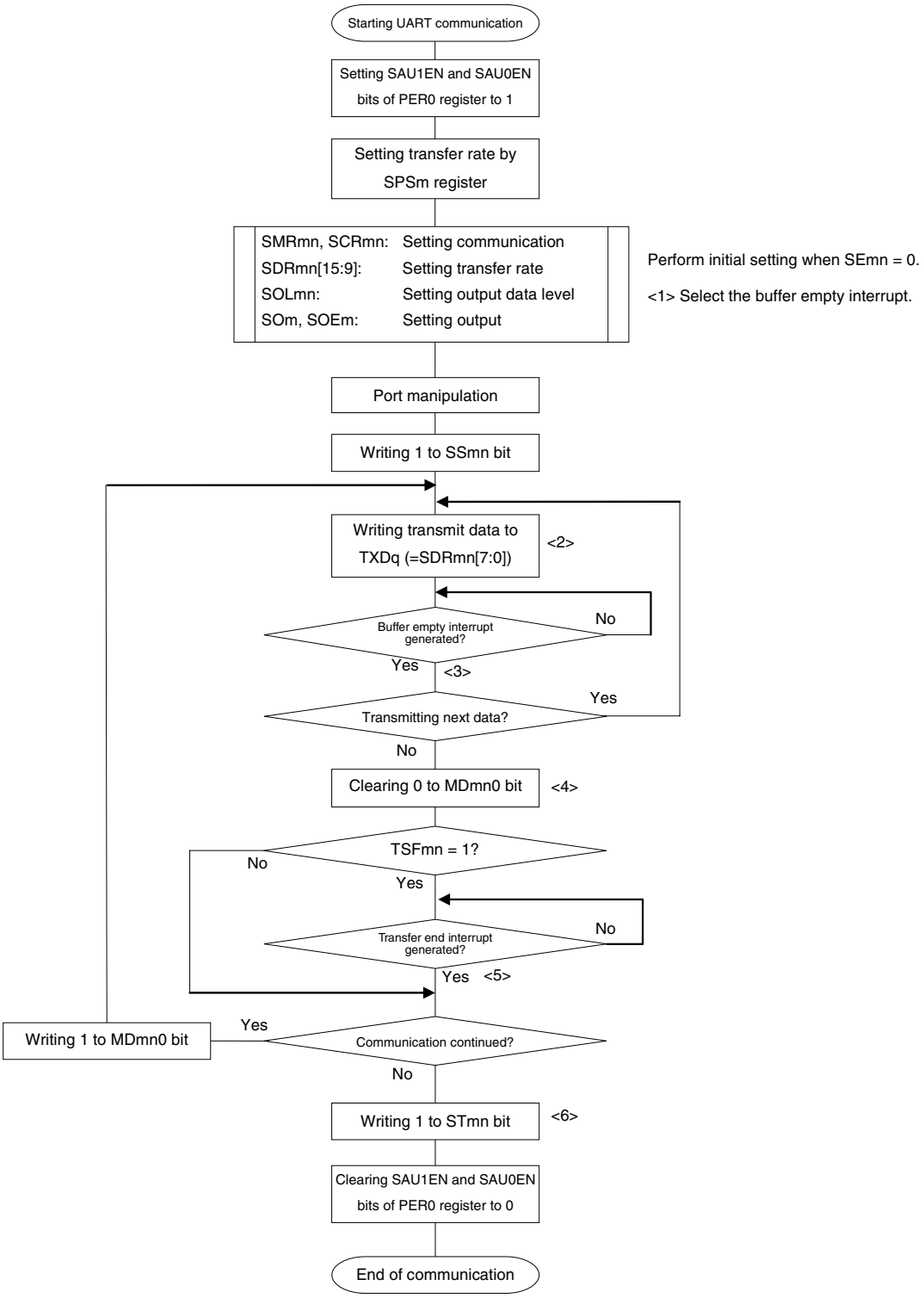


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

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), mn = 00, 02, 10, 12

Figure 11-76. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> - <6> in the figure correspond to <1> - <6> in Figure 11-75 Timing Chart of UART Transmission (in Continuous Transmission Mode).

11.6.2 UART reception

UART reception is an operation wherein the 78K0R/KC3-L, KE3-L asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of 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 first 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 28 ELECTRICAL SPECIFICATIONS).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency
2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

(1) Register setting

Figure 11-77. Example of Contents of Registers for UART Reception of UART (UART0 -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 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	1	SOm2 ×	SOm1 ×	SOm0 ×

(b) Serial output enable register m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Some	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	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 ×	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

(g) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Receive data register							

RXDq

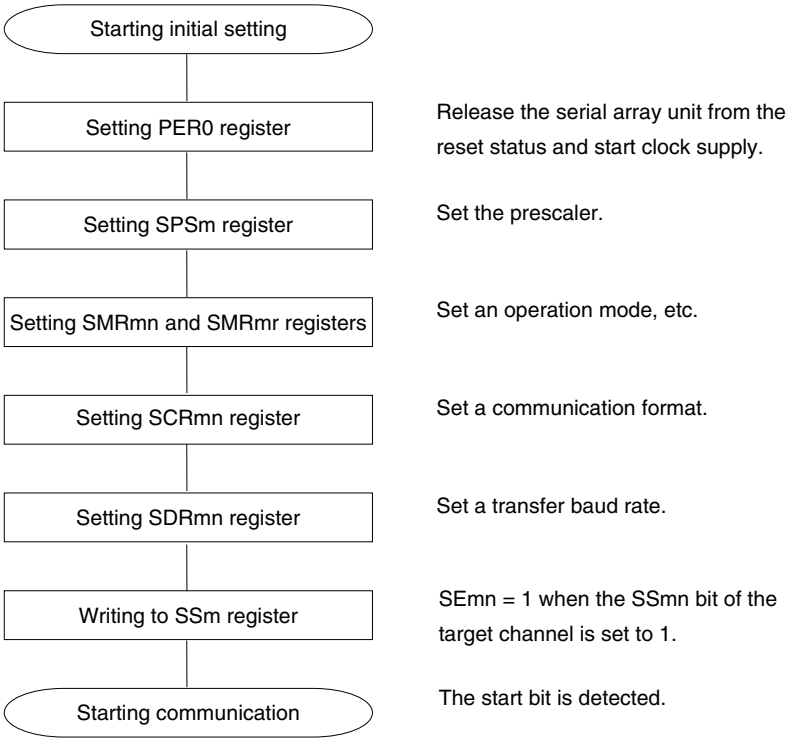
Caution For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.

Figure 11-77. Example of Contents of Registers for UART Reception of UART (UART0 to UART3) (2/2)

Remark 1 m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13
r: Channel number (r = n – 1), q: UART number (q = 0 to 3)
2. ☐ : Setting is fixed in the UART reception mode, ▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-78. Initial Setting Procedure for UART Reception



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-79. Procedure for Stopping UART Reception

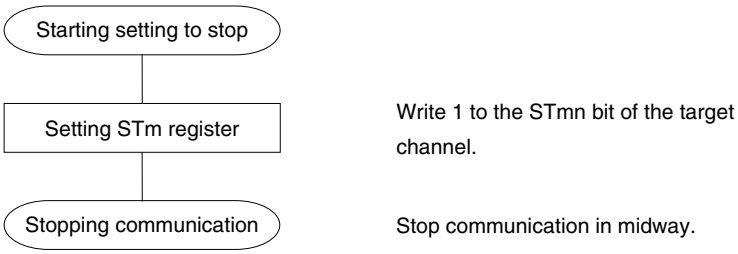
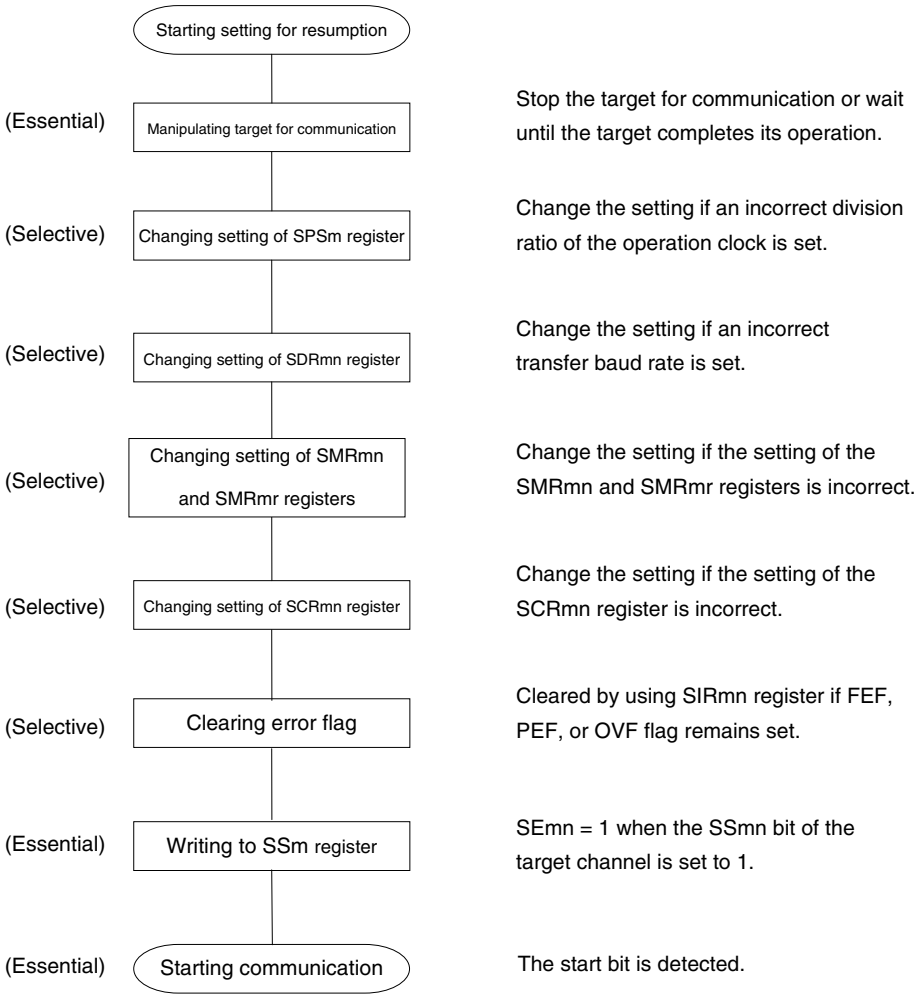
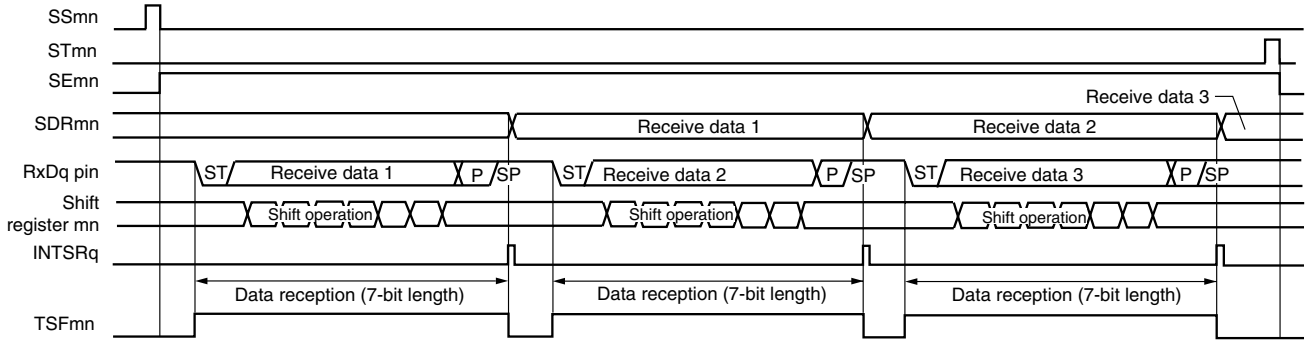


Figure 11-80. Procedure for Resuming UART Reception



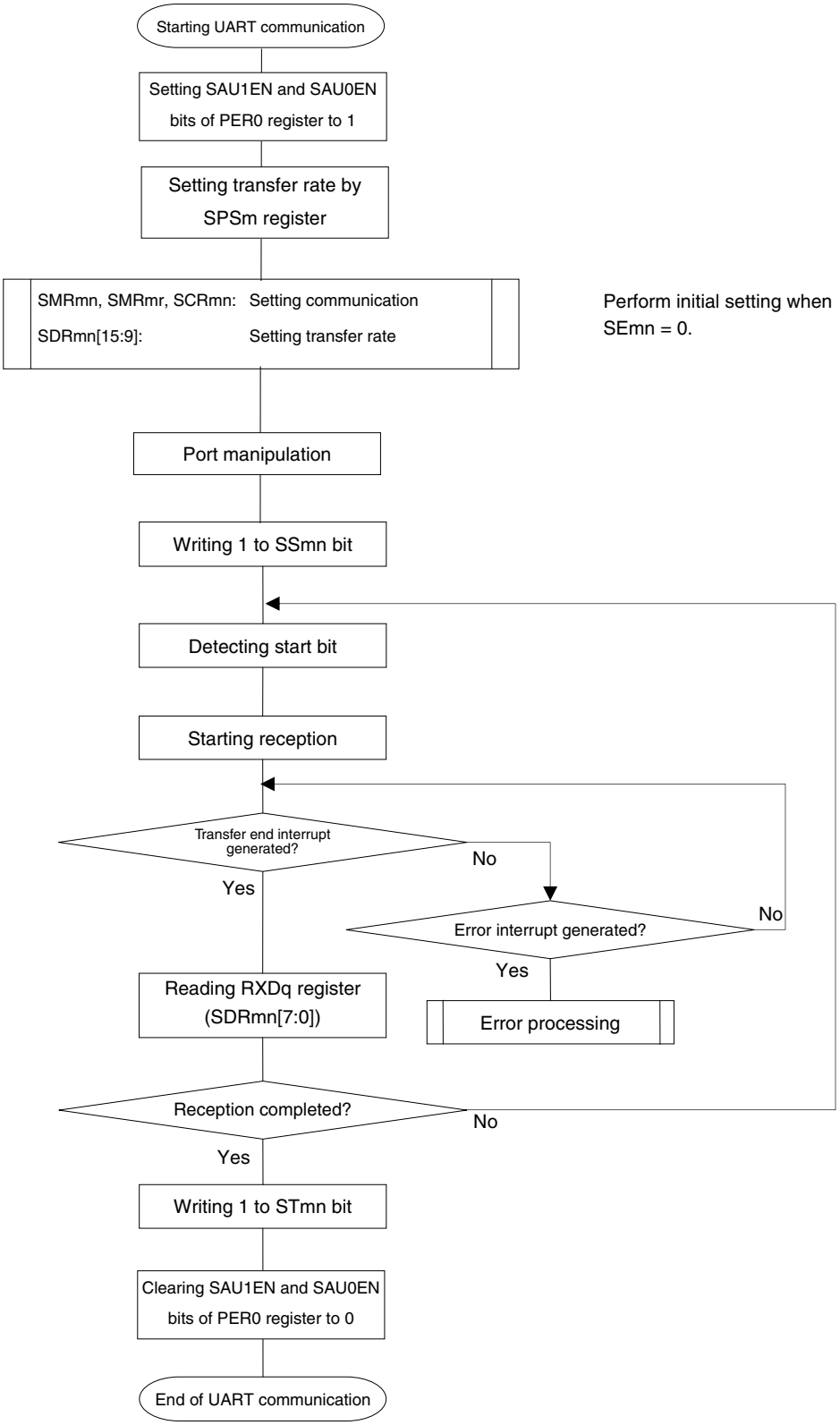
(3) Processing flow

Figure 11-81. Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

Figure 11-82. Flowchart of UART Reception



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

11.6.3 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] (SDR12 [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 first 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 28 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.

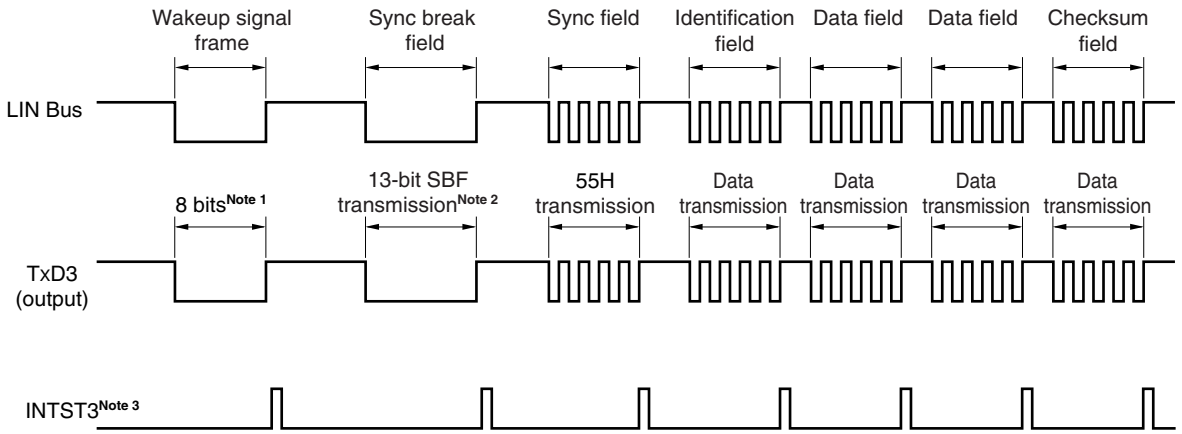
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 11-82 outlines a transmission operation of LIN.

Figure 11-83. 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.

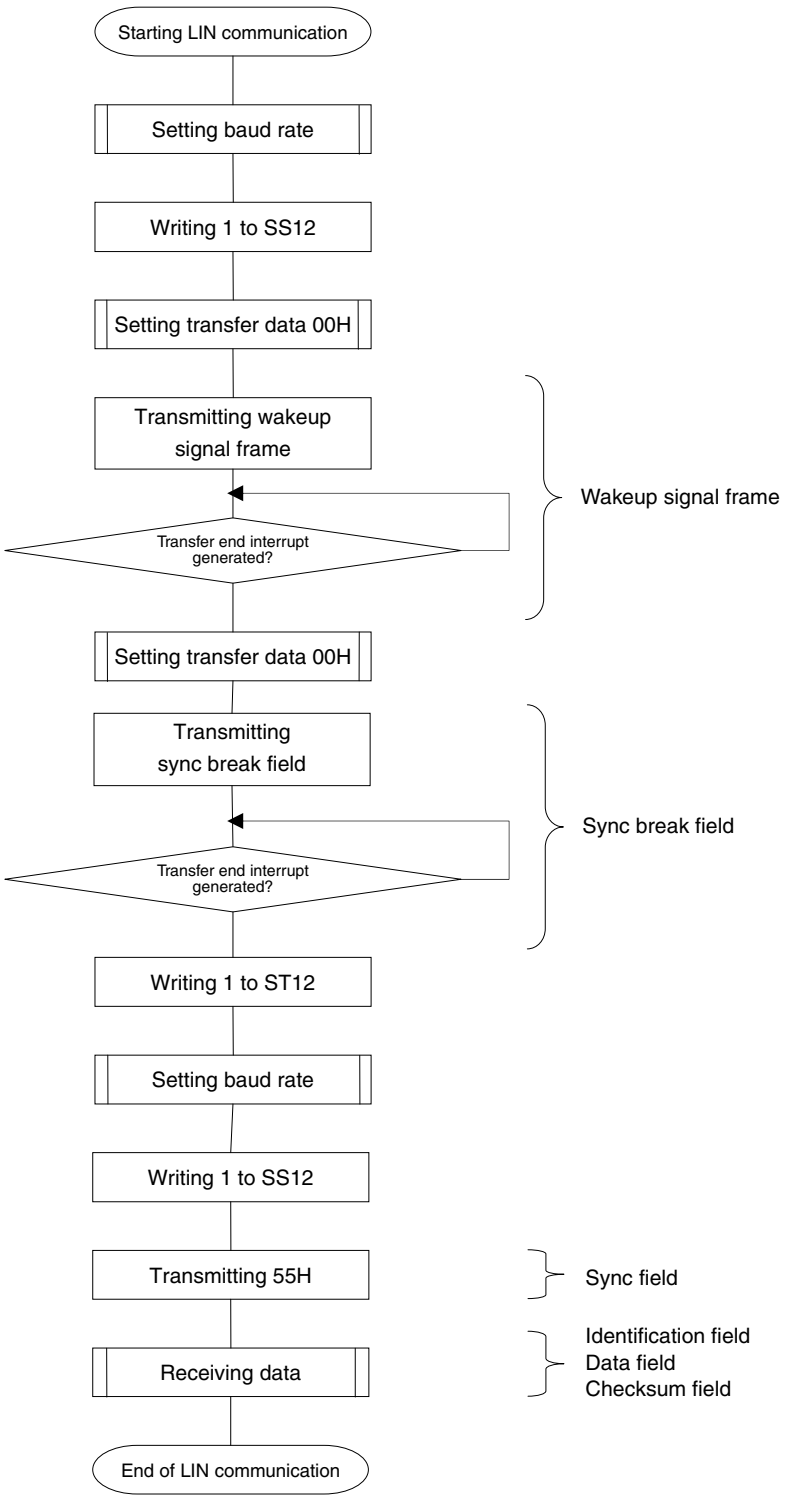
(Baud rate of sync break field) = 9/13 × N

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 11-84. Flowchart for LIN Transmission



11.6.4 LIN reception

Of UART reception, UART3 supports LIN communication.

For LIN reception, channel 3 of unit 1 (SAU1) is used.

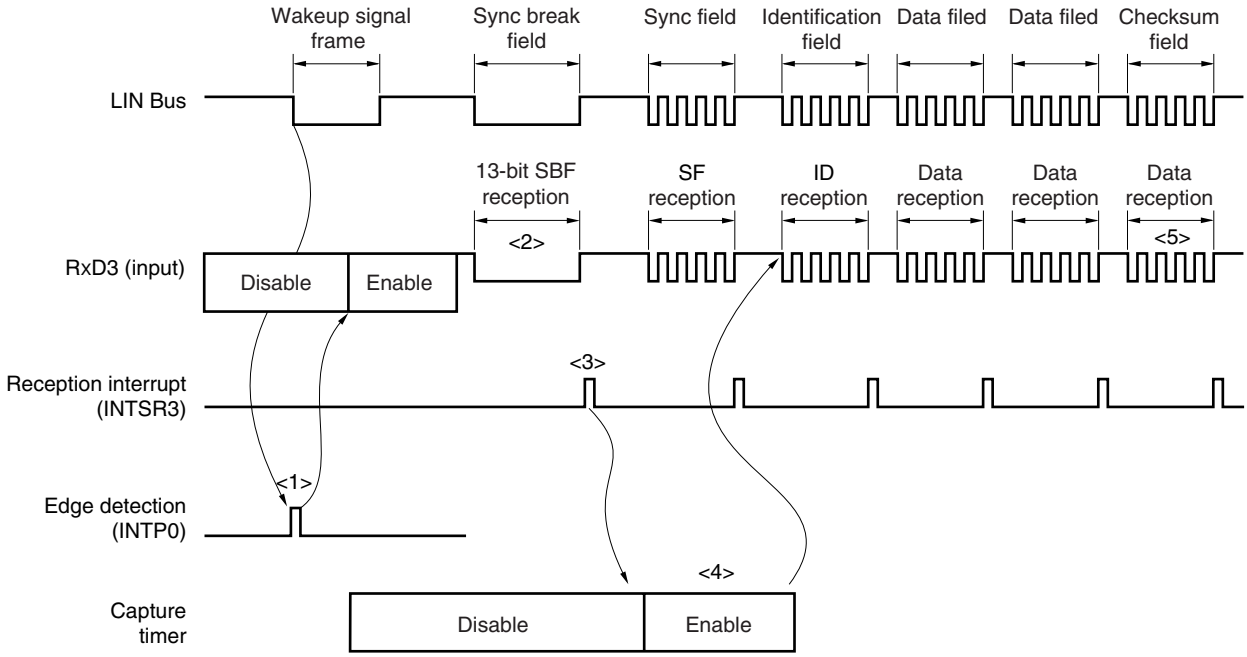
UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Not supported	Supported
Target channel	–	–	–	Channel 3 of SAU1
Pins used	–	–	–	RxD3
Interrupt	–	–	–	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	–	–	–	INTSRE3
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] (SDR13 [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 28 ELECTRICAL SPECIFICATIONS).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

Figure 11-85 outlines a reception operation of LIN.

Figure 11-85. 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 TAU0 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.

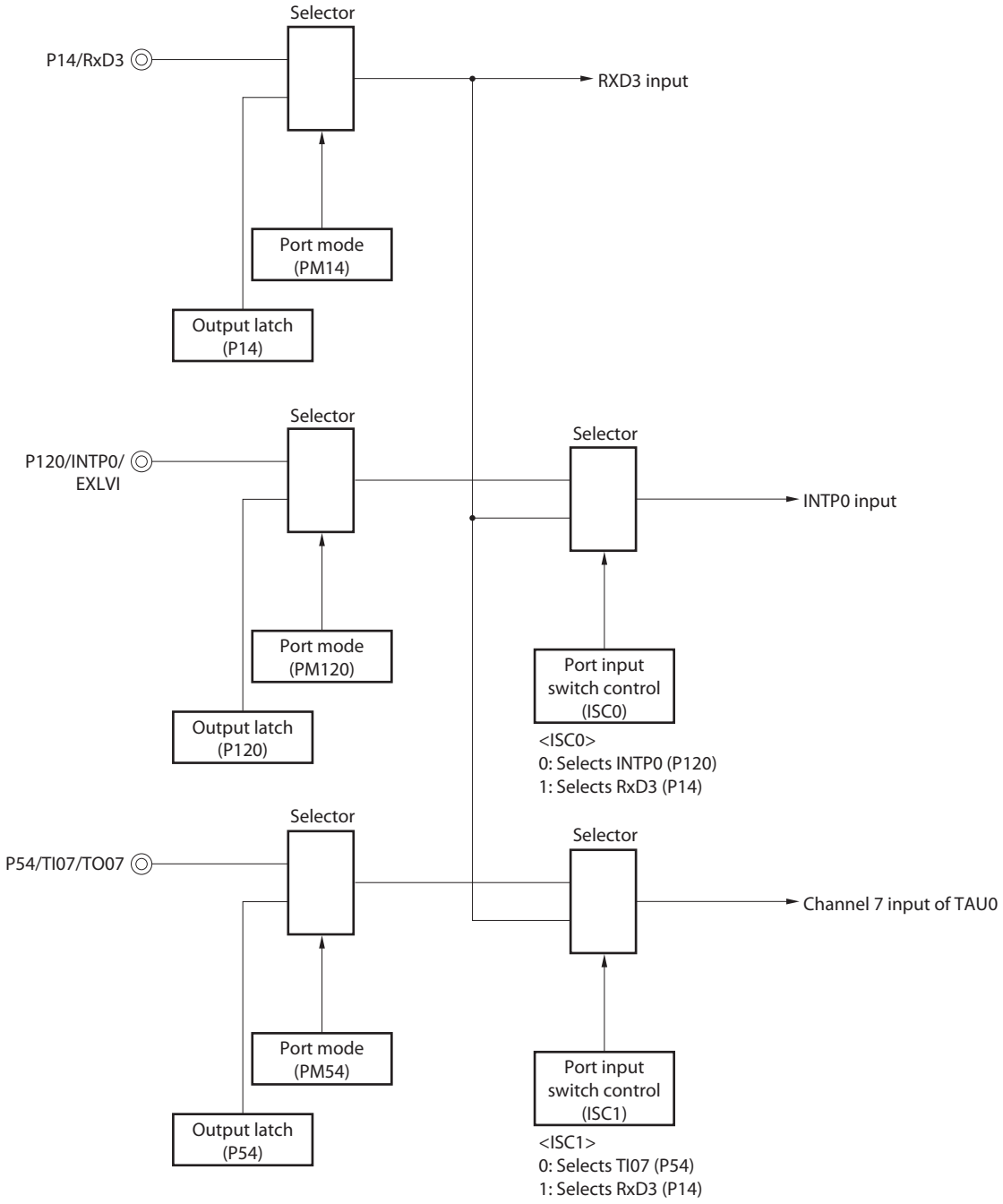
<5> The checksum field should be distinguished by software. In addition, processing to initialize UART3 after the checksum field is received and to wait for reception of SBF should also be performed by software.

Figure 11-85 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 (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 (TAU0).

Figure 11-86. Port Configuration for Manipulating Reception of LIN



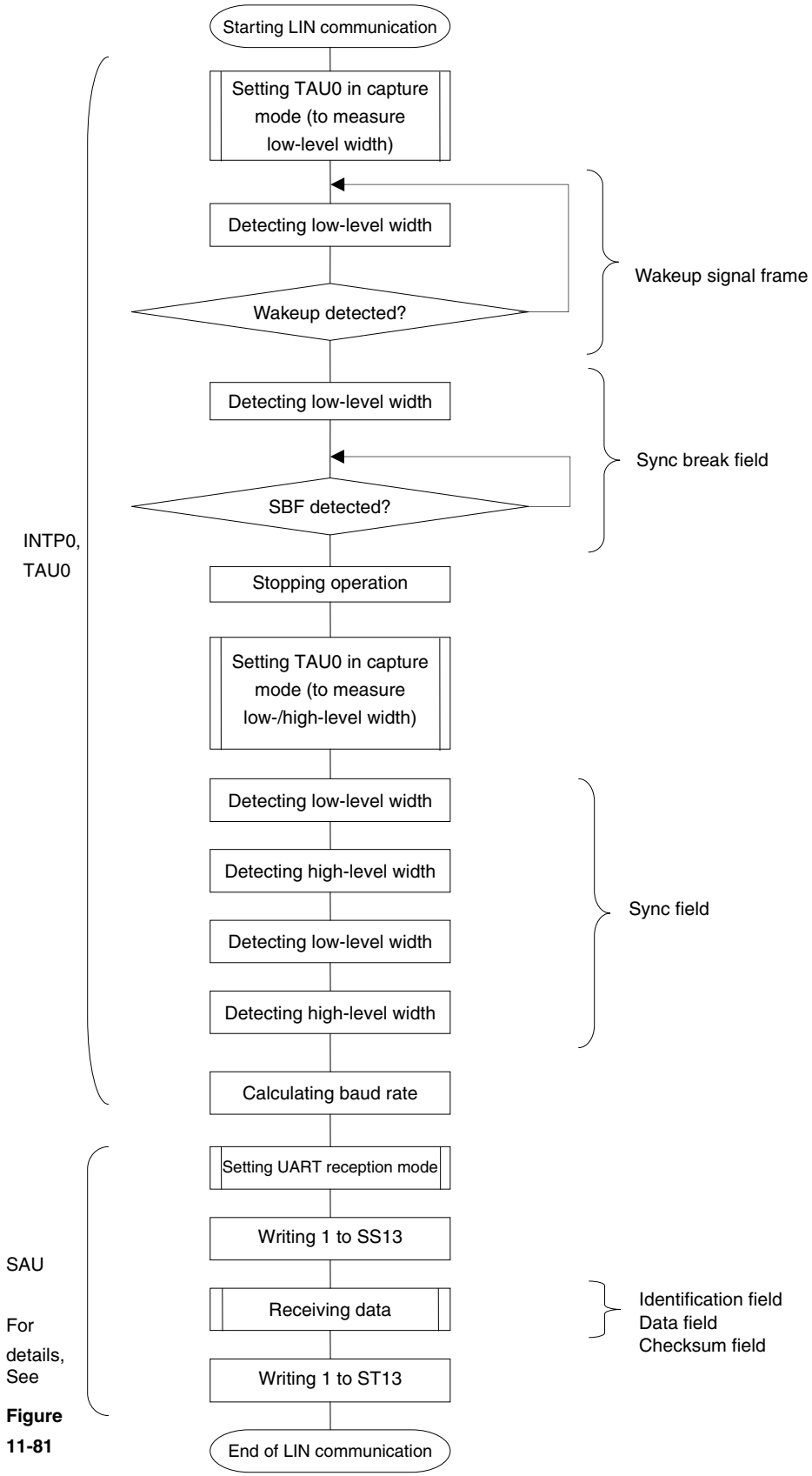
Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See Figure 11-17.)

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 (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 (SAU1)

Figure 11-87. Flowchart of LIN Reception



11.6.5 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (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 - 9 of the SDRmn register (0000010B - 1111111B) and therefore is 2 - 127.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 - 03, 10 - 13

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-3. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK) Note 1			
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 10 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	12 MHz	16 MHz	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	6 MHz	8 MHz	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	3 MHz	4 MHz	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	1.5 MHz	2 MHz	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	0.75 MHz	1 MHz	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	375 kHz	500 kHz	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	187.5 kHz	250 kHz	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	93.75 kHz	125 kHz	156.25 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	46.88 kHz	62.5 kHz	78.13 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	23.44 kHz	31.25 kHz	39.06 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	11.72 kHz	15.63 kHz	19.53 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	5.86 kHz	7.81 kHz	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0 Note 2. In the case of m = 1, setting is prohibited.			
1	0	0	0	0	X	X	X	X	f_{CLK}	12 MHz	16 MHz	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	6 MHz	8 MHz	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	3 MHz	4 MHz	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	1.5 MHz	2 MHz	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	0.75 MHz	1 MHz	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	375 kHz	500 kHz	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	187.5 kHz	250 kHz	312.5 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	93.75 kHz	125 kHz	156.25 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	46.88 kHz	62.5 kHz	78.13 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	23.44 kHz	31.25 kHz	39.06 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	11.72 kHz	15.63 kHz	19.53 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	5.86 kHz	7.81 kHz	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0 Note 2. In the case of m = 1, setting is prohibited.			
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 (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, sub system clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting the TIS02 bit of timer input select register 0 (TIS0) to 1) and selecting INTTM02 by using the SPS0 register in channel 2 of TAU0. When changing f_{CLK}, however, SAU0 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 - 3), mn = 00 - 03, 10 - 13

(2) Baud rate error during transmission

The baud rate error of UART (UART0 - 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} = 12, 16, 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 12, 16, 20 MHz				
	Operation Clock (MCK)	SDRmn[15:9	Error margin with target baud rate		
			12 MHz	16 MHz	20 MHz
300 bps	$f_{CLK}/2^9$	64	0.16%	0.16%	0.16%
600 bps	$f_{CLK}/2^8$	64	0.16%	0.16%	0.16%
1200 bps	$f_{CLK}/2^7$	64	0.16%	0.16%	0.16%
2400 bps	$f_{CLK}/2^6$	64	0.16%	0.16%	0.16%
4800 bps	$f_{CLK}/2^5$	64	0.16%	0.16%	0.16%
9600 bps	$f_{CLK}/2^4$	64	0.16%	0.16%	0.16%
19200 bps	$f_{CLK}/2^3$	64	0.16%	0.16%	0.16%
31250 bps	$f_{CLK}/2^3$	39	±0.0%	±0.0%	±0.0%
38400 bps	$f_{CLK}/2^2$	64	0.16%	0.16%	0.16%
76800 bps	$f_{CLK}/2$	64	0.16%	0.16%	0.16%
153600 bps	f_{CLK}	64	0.16%	0.16%	0.16%
312500 bps	f_{CLK}	31	1.05%	-1.54%	±0.0%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 - 2), mn = 00, 02, 10, 12

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

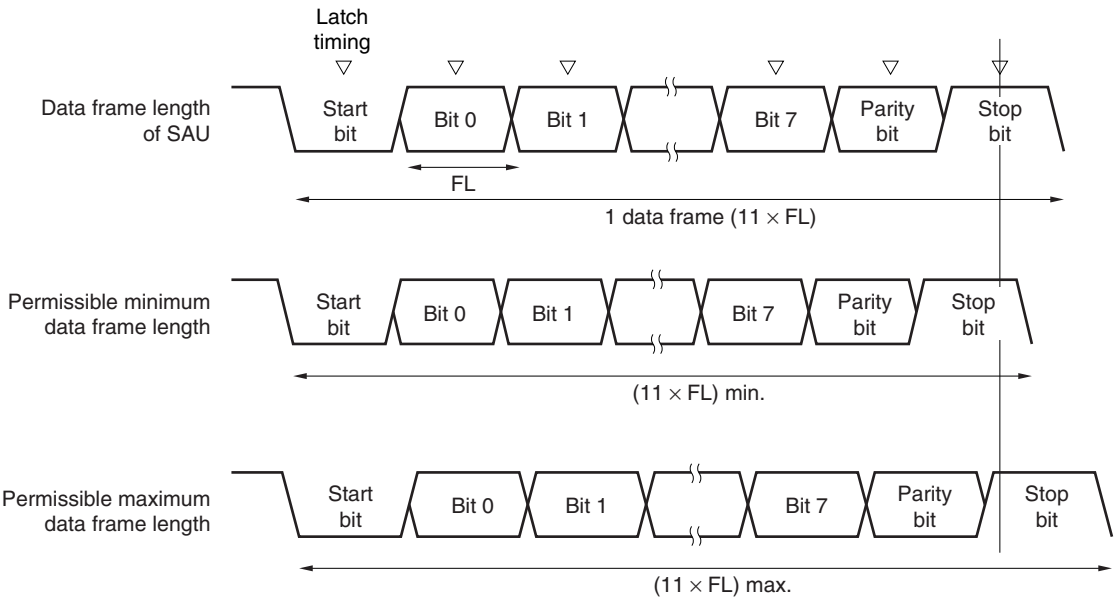
Brate: Calculated baud rate value at the reception side (See **11.6.5 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

Figure 11-88. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 11-88, 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.

11.7 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 11.7.3 (2) Processing flow for details.

Remarks 1. To use the full-function I²C bus, see CHAPTER 12 SERIAL INTERFACE IICA.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10

The channels supporting simplified I²C (IIC10, IIC20) changes as below.

<In case of 78K0R/KC3-L>

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	–	–
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

<In case of 78K0R/KE3-L>

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	–
	1	CSI01		–
	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 **11.7.1**.)
- Data transmission (See **11.7.2**.)
- Data reception (See **11.7.3**.)
- Stop condition generation (See **11.7.4**.)

In this chapter 11.7.1 related to control register, 11.2 Configuration of Serial Array Unit, of 78K0R/KE3-L has been taken as example and explained. Refer it for difference between configuration of Serial Array Unit of KC3-L and KE3-L.

11.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^2C	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 f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first 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 N-ch open-drain output (V_{DD} tolerance) mode (POM03, POM143 = 1) for the port output mode registers (POM0, POM14) (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 (POM04, POM142 = 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), mn = 02, 10

(1) Register setting

Figure 11-89. 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	CKOm1 ×	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	SOm1 ×	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	SOEm1 ×	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)							

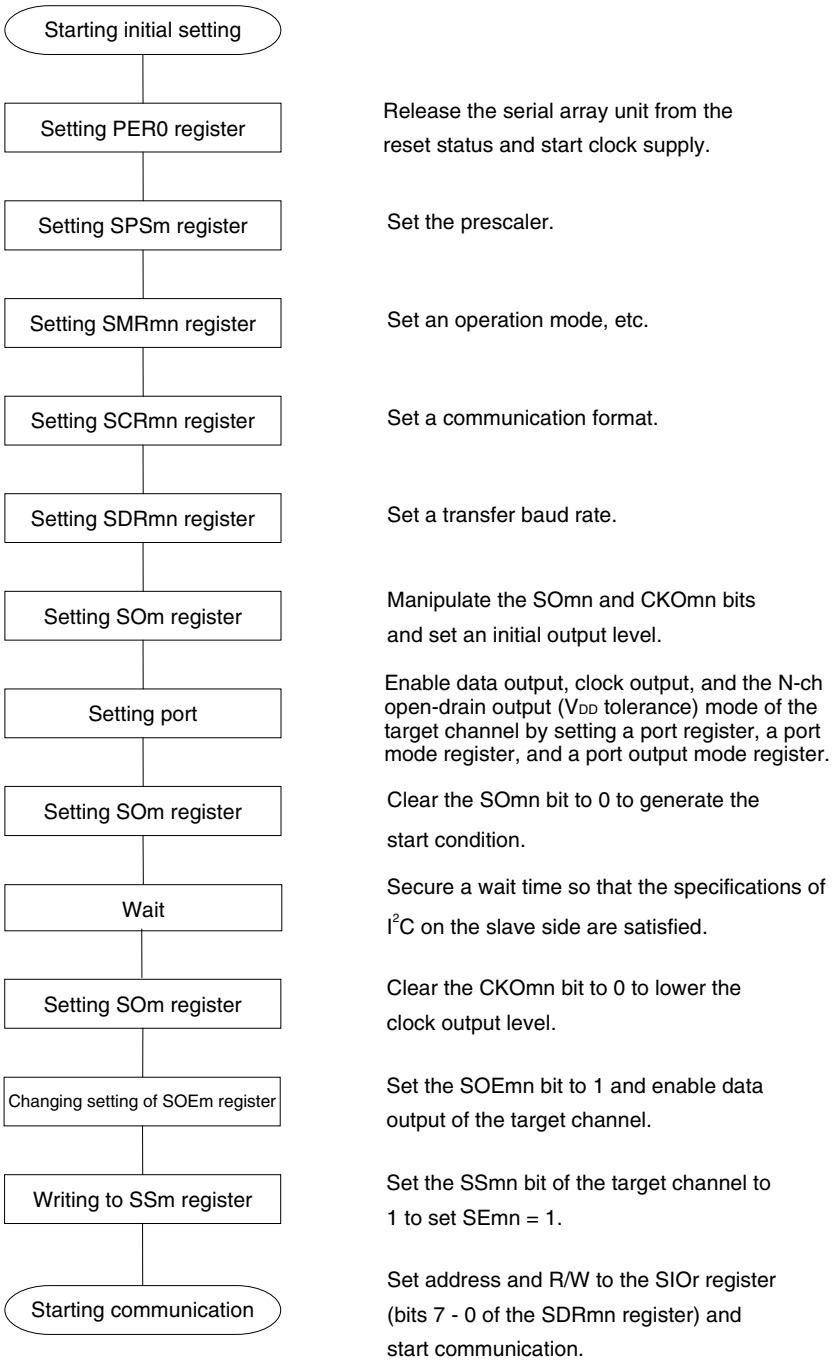
SIOr

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

2. □: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in IIC mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

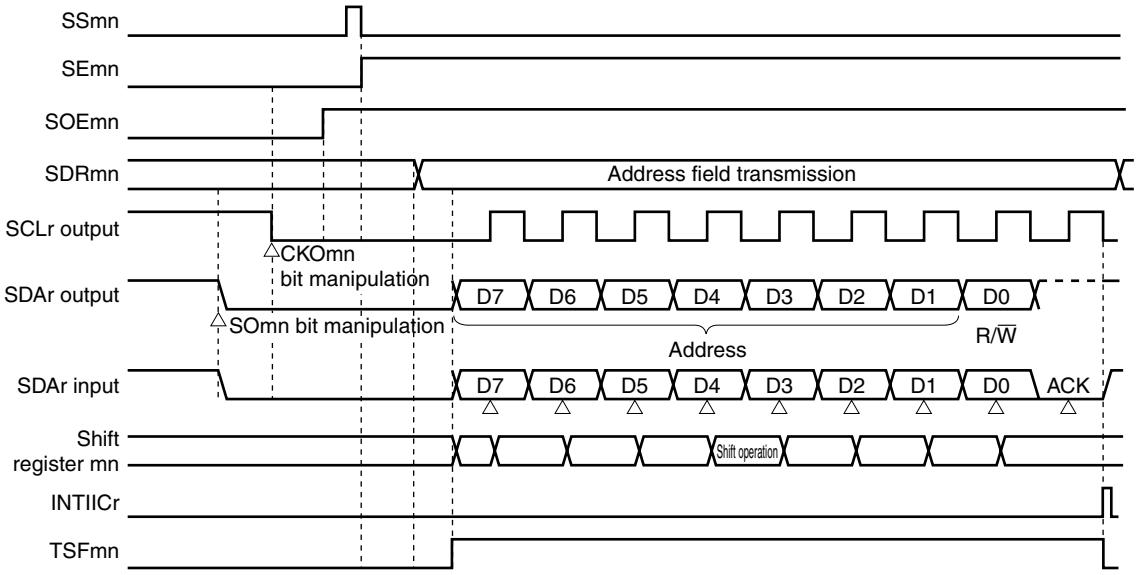
Figure 11-90. Initial Setting Procedure for Address Field Transmission



Caution After setting the SAUmEN bit of PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

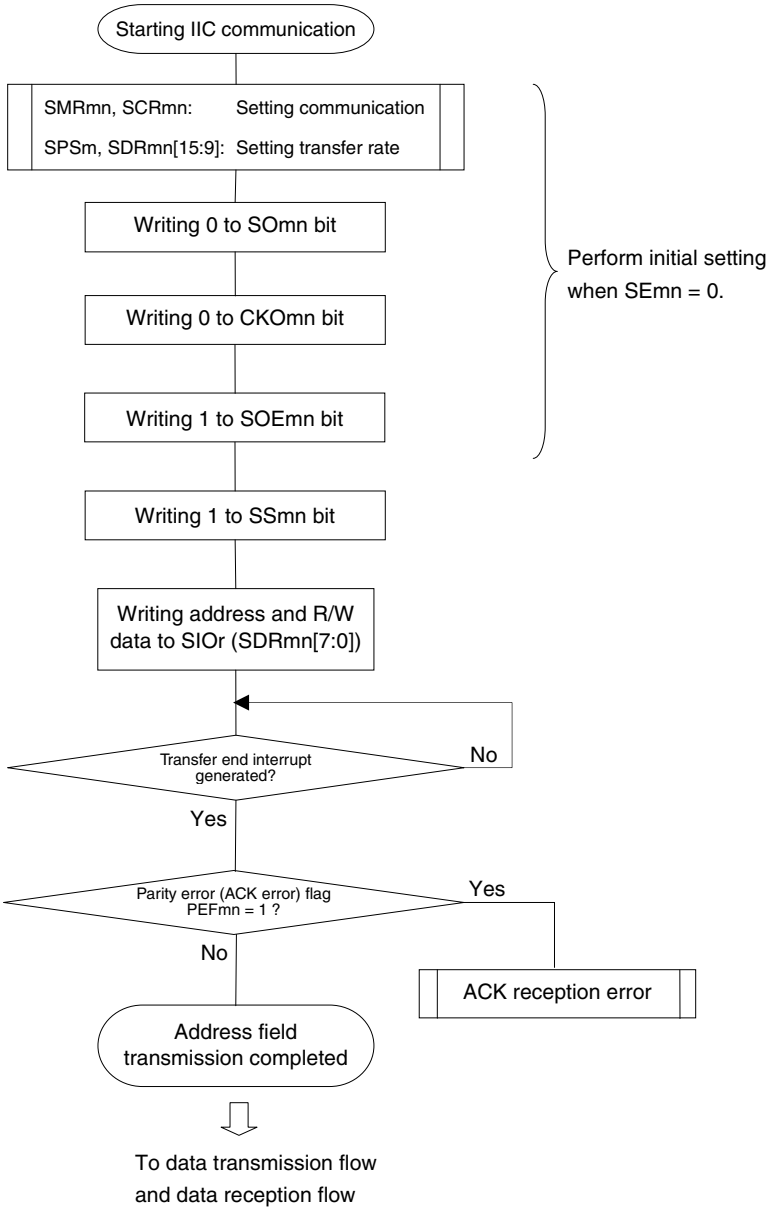
(3) Processing flow

Figure 11-91. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-92. Flowchart of Address Field Transmission



11.7.2 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 f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first 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 N-ch open-drain output (V_{DD} tolerance) mode (POM03, POM143 = 1) for the port output mode registers (POM0, POM14) (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 (POM04, POM142 = 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), mn = 02, 10

(1) Register setting

Figure 11-93. 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	CKOm1 ×	CKOm0 0/1Note	0	0	0	0	1	SOm2 0/1Note	SOm1 ×	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	SOEm1 ×	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

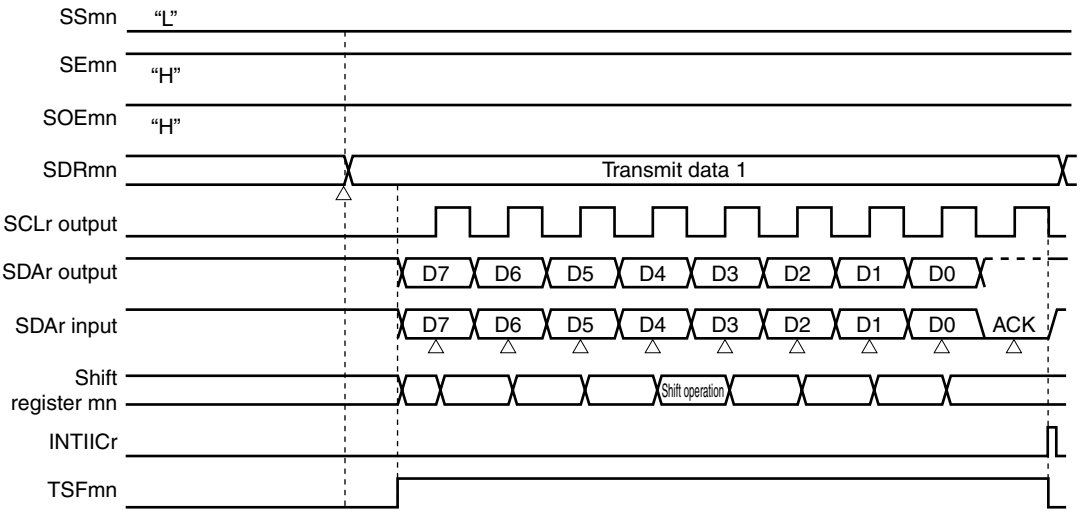
Note The value varies depending on the communication data during communication operation.

Remarks 1 r: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

2. ☐ : Setting is fixed in the IIC mode, ▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

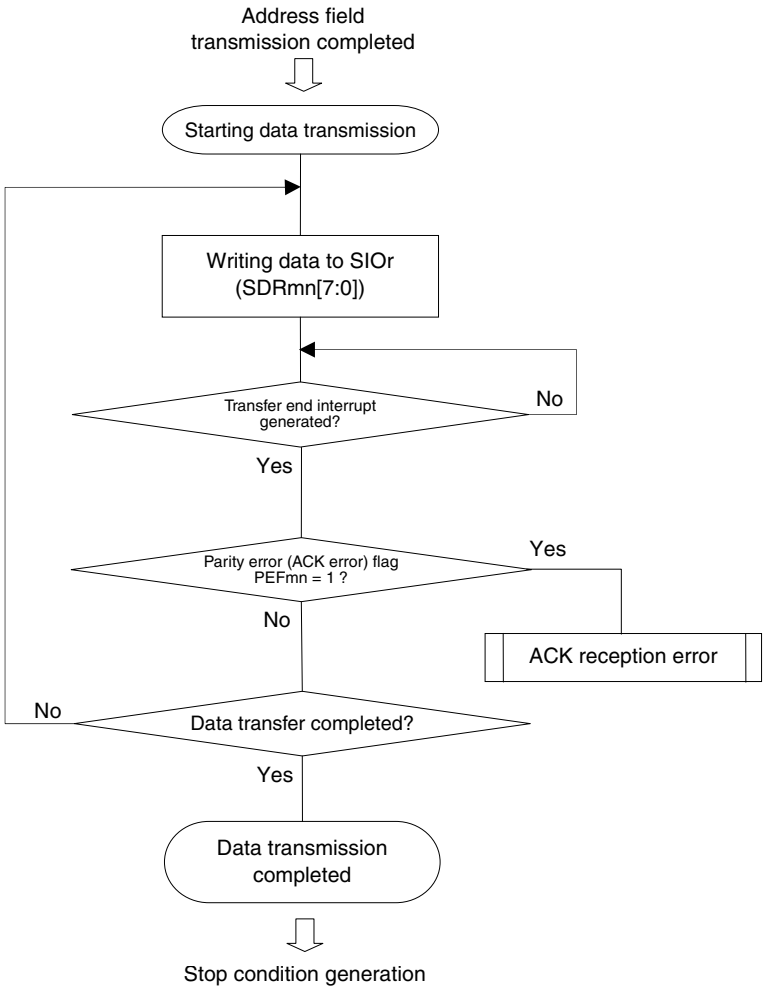
(2) Processing flow

Figure 11-94. Timing Chart of Data Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-95. Flowchart of Data Transmission



11.7.3 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 f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first 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 N-ch open-drain output (V_{DD} tolerance) mode (POM03, POM143 = 1) for the port output mode registers (POM0, POM14) (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 (POM04, POM142 = 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), mn = 02, 10

(1) Register setting

Figure 11-96. Example of Contents of Registers for Data Reception 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	CKOm1 ×	CKOm0 0/1Note	0	0	0	0	1	SOm2 0/1Note	SOm1 ×	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	SOEm1 ×	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.

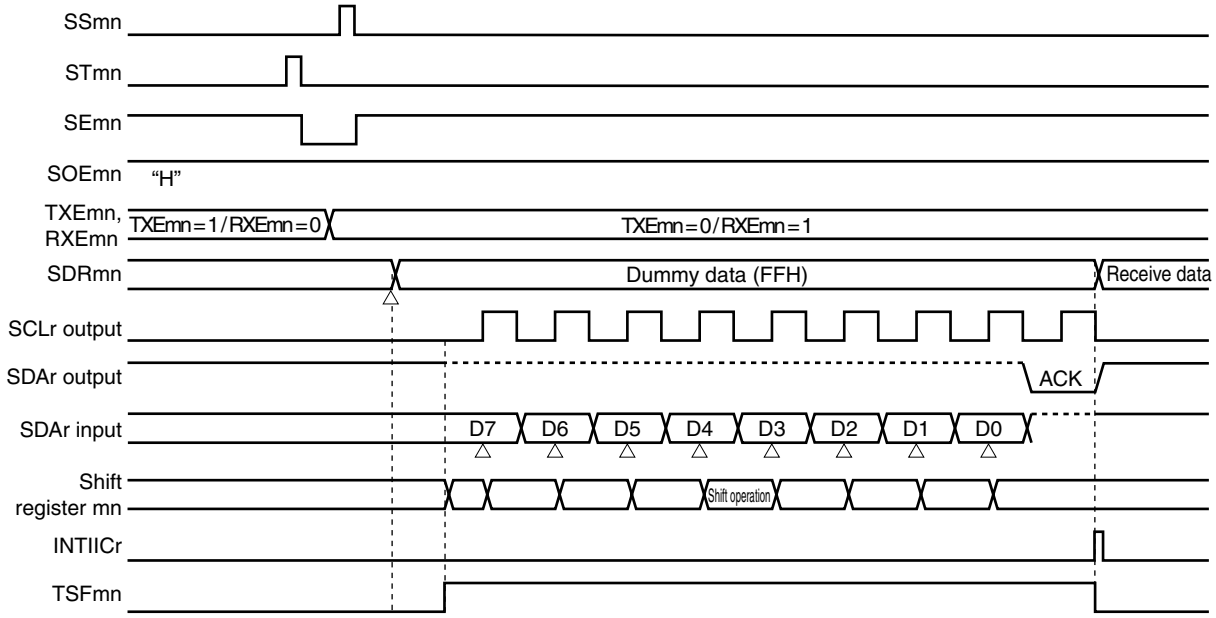
Remarks 1 r: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

2. □: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

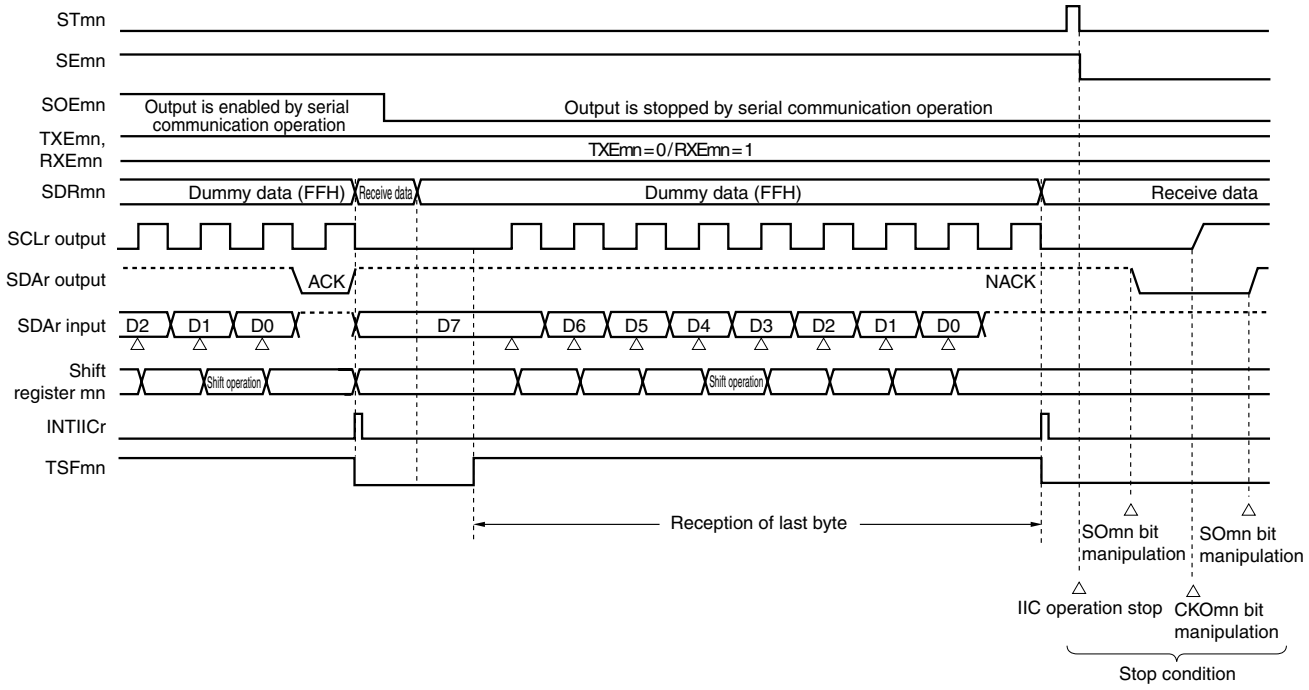
(2) Processing flow

Figure 11-97. Timing Chart of Data Reception

(a) When starting data reception

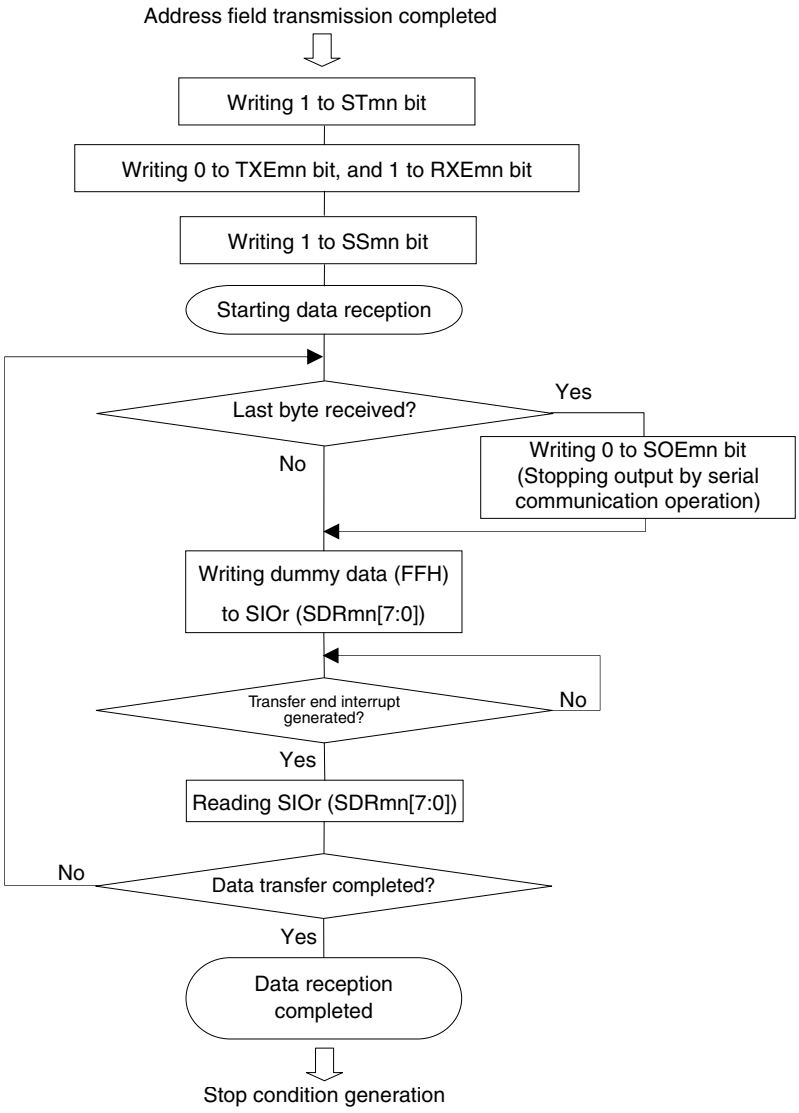


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-98. Flowchart of Data Reception



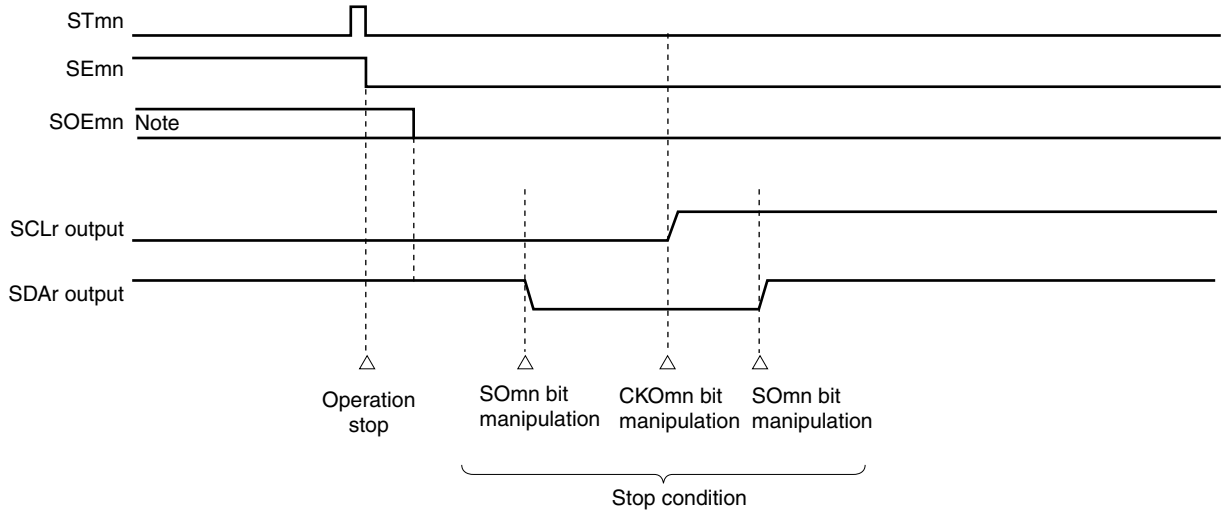
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STmn bit to stop operation and generating a stop condition.

11.7.4 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

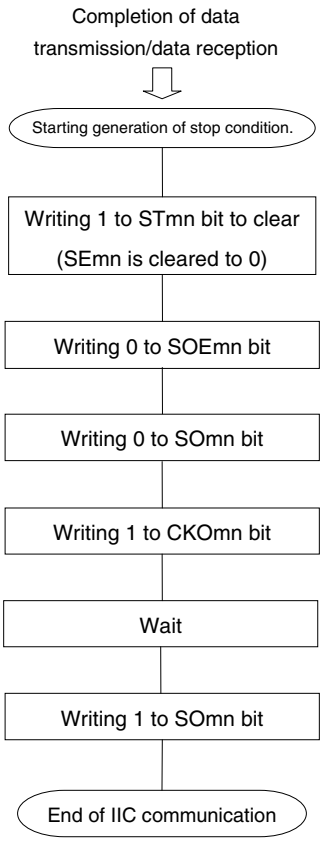
Figure 11-99. Timing Chart of Stop Condition Generation



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), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-100. Flowchart of Stop Condition Generation



Secure a wait time so that the specifications of I^2C on the slave side are satisfied.

11.7.5 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

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), mn = 02, 10

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-4. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK) Note 1			
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 12 MHz	f_{CLK} = 16 MHz	f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	12 MHz	16 MHz	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	6 MHz	8 MHz	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	3 MHz	4 MHz	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	1.5 MHz	2 MHz	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	0.75 MHz	1 MHz	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	375 kHz	500 kHz	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	187.5 kHz	250 kHz	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	93.75 kHz	125 kHz	156.25 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	46.88 kHz	62.5 kHz	78.13 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	23.44 kHz	31.25 kHz	39.06 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	11.72 kHz	15.63 kHz	19.53 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	5.86 kHz	7.81 kHz	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0 Note 2. In the case of m = 1, setting is prohibited.			
1	0	0	0	0	X	X	X	X	f_{CLK}	12 MHz	16 MHz	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	6 MHz	8 MHz	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	3 MHz	4 MHz	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	1.5 MHz	2 MHz	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	0.75 MHz	1 MHz	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	375 kHz	500 kHz	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	187.5 kHz	250 kHz	312.5 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	93.75 kHz	125 kHz	156.25 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	46.88 kHz	62.5 kHz	78.13 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	23.44 kHz	31.25 kHz	39.06 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	11.72 kHz	15.63 kHz	19.53 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	5.86 kHz	7.81 kHz	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0 Note 2. In the case of m = 1, setting is prohibited.			
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 (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, sub system clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting the TIS02 bit of timer input select register 0 (TIS0) to 1) and selecting INTTM02 by using the SPS0 register in channel 2 of TAU0. When changing f_{CLK}, however, SAU0 and TAU0 must be stopped as described in Note 1 above.

<Remark is explained on next page.>

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10

Here is an example of setting an IIC transfer rate where MCK = f_{CLK} = 12, 16, 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 12, 16, 20 MHz							
	Operation Clock (MCK)	SDRmn[15:9]			Calculated Transfer Rate	Error from Desired Transfer Rate		
		12 MHz	16 MHz	20 MHz		12 MHz	16 MHz	20 MHz
100 kHz	f_{CLK}	99	79	59	100 kHz	0.0%	0.0%	0.0%
400 kHz	f_{CLK}	24	19	14	400 kHz	0.0%	0.0%	0.0%

11.8 Processing Procedure in Case of Error

The processing procedure to be followed if an error of each type occurs is described in Figures 11-101 to 11-103.

Figure 11-101. 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.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00 to 03, 10, 13

Figure 11-102. 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.	Only error generated at the point of reading can be cleared, 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 - 3) , mn = 00 - 03, 10 - 13

Figure 11-103. 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.	Only error generated at the point of reading can be cleared, 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 retry 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

11.9 Relationship Between Register Settings and Pins

Tables 11-5 to 11-12 show the relationship between register settings and pins for each channel of serial array units 0 and 1.

Table 11-5. Relationship between register settings and pins (Channel 0 of unit 0: CSI00, UART0 transmission)

SE00 Note 1	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM10	P10	PM11 Note 2	P11 Note 2	PM12	P12	Operation mode	Pin Function		
															$\overline{\text{SCK00}}$/P10	SI00/RxD0/P11 Note 2	SO00/TxD0/P12
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P10	P11	P12
	0	1														P11/RxD0	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI00 reception	$\overline{\text{SCK00}}$ (input)	SI00	P12
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI00 transmission	$\overline{\text{SCK00}}$ (input)	P11	SO00
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (input)	SI00	SO00
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI00 reception	$\overline{\text{SCK00}}$ (output)	SI00	P12
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI00 transmission	$\overline{\text{SCK00}}$ (output)	P11	SO00
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (output)	SI00	SO00
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART0 transmission Note 5	P10	P11/RxD0	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 11-6). 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 11.3 (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 11-6).

Remark X: Don't care

Table 11-6. Relationship between register settings and pins (Channel 1 of unit 0: UART0 reception)

SE01 Note 1	MD012	MD011	TXE01	RXE01	PM11 Note 2	P11 Note 2	Operation mode	Pin Function
								SI00/RxD0/P11 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI00/P11 Note 2
1	0	1	0	1	1	×	UART0 reception Notes 4, 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 11-5).
When channel 0 of unit 0 is set to CSI10, 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 11-5).

5. The SMR00 register of channel 0 of unit 0 must also be set during UART0 reception. For details, refer to 11.6.2 (1) Register setting.

Remark X: Don't care

Table 11-7. Relationship between register settings and pins
(Channel 2 of unit 0: CSI10, UART1 transmission, IIC10)

SE 02 Note 1	MD 022	MD 021	SOE 02	SO 02	CKO 02	TXE 02	RXE 02	PM 04	P04	PM03 Note 2	P03 Note 2	PM02	P02	Operation mode	Pin Function		
															$\overline{\text{SCK10}}$/ SCL10/P04	SI10/SDA10/ RxD1/P03 Note 2	SO10/ TxD1/P02
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P04	P03	P02
	0	1														P03/RxD1	
	1	0														P03	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI10 reception	$\overline{\text{SCK10}}$ (input)	SI10	P02
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI10 transmission	$\overline{\text{SCK10}}$ (input)	P03	SO10
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission /reception	$\overline{\text{SCK10}}$ (input)	SI10	SO10
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI10 reception	$\overline{\text{SCK10}}$ (output)	SI10	P02
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI10 transmission	$\overline{\text{SCK10}}$ (output)	P03	SO10
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI10 transmission /reception	$\overline{\text{SCK10}}$ (output)	SI10	SO10
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART1 transmission Note 5	P04	P03/RxD1	TxD1
0	1	0	0	0/1 Note 6	0/1 Note 6	0	0	0	1	0	1	× Note 3	× Note 3	IIC10 start condition	SCL10	SDA10	P02
						1	0										
						0	1										
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC10 address field transmission	SCL10	SDA10	P02
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC10 data transmission	SCL10	SDA10	P02
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC10 data reception	SCL10	SDA10	P02
0			0	0/1 Note 7	0/1 Note 7	0	0	0	1	0	1	× Note 3	× Note 3	IIC10 stop condition	SCL10	SDA10	P02
						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 11-8). 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 11.3 (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 11-8).

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 11-8. Relationship between register settings and pins (Channel 3 of unit 0: UART1 reception)

SE03 Note 1	MD032	MD031	TXE03	RXE03	PM03 Note 2	P03 Note 2	Operation mode	Pin Function SI10/SDA10/RxD1/P03 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI10/SDA10/P03 Note 2
1	0	1	0	1	1	×	UART1 reception Notes 4, 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 11-7).
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 11-7).

5. The SMR02 register of channel 2 of unit 0 must also be set during UART1 reception. For details, refer to 11.6.2 (1) Register setting.

Remark X: Don't care

Table 11-9. Relationship between register settings and pins
(Channel 0 of unit 1: CSI20, UART2 transmission, IIC20)

SE 10 Note 1	MD 102	MD 101	SOE 10	SO 10	CKO 10	TXE 10	RXE 10	PM 142	P142	PM 143 Note 2	P143 Note 2	PM 144	P144	Operation mode	Pin Function		
															$\overline{\text{SCK20}}$/ SCL20/P142	SI20/SDA20/ RxD2/P143 Note 2	SO20/ TxD2/P144
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P142	P143	P144
	0	1														P143/RxD2	
	1	0														P143	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI20 reception	$\overline{\text{SCK20}}$ (input)	SI20	P144
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI20 transmission	$\overline{\text{SCK20}}$ (input)	P143	SO20
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI20 transmission/reception	$\overline{\text{SCK20}}$ (input)	SI20	SO20
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI20 reception	$\overline{\text{SCK20}}$ (output)	SI20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI20 transmission	$\overline{\text{SCK20}}$ (output)	P143	SO20
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI20 transmission/reception	$\overline{\text{SCK20}}$ (output)	SI20	SO20
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART2 transmission Note 5	P142	P143/RxD2	TxD2
0	1	0	0	0/1 Note 6	0/1 Note 6	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 start condition	SCL20	SDA20	P144
						1	0										
						0	1										
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 address field transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 data transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC20 data reception	SCL20	SDA20	P144
0			0	0/1 Note 7	0/1 Note 7	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 stop condition	SCL20	SDA20	P144
						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 11-10). 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 11.3 (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 11-10).

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 11-10. Relationship between register settings and pins (Channel 1 of unit 1: UART2 reception)

SE11 Note 1	MD112	MD111	TXE11	RXE11	PM143 Note 2	P143 Note 2	Operation mode	Pin Function
								SI20/SDA20/RxD2/P143 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI20/SDA20/P143
1	0	1	0	1	1	×	UART2 reception Notes 4, 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 11-9).
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 11-9).

5. The SMR10 register of channel 0 of unit 1 must also be set during UART2 reception. For details, refer to 11.6.2 (1) Register setting.

Remark X: Don't care

Table 11-11. Relationship between register settings and pins (Channel 2 of unit 1: UART3 transmission)

SE12 Note1	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM13 Note 2	P13 Note 2	Operation mode	Pin Function
										TxD3/P13
0	0	1	0	1	0	0	× Note 3	× Note 3	Operation stop mode	P13
1	0	1	1	0/1 Note 4	1	0	0	1	UART3 transmission Note 5	TxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 3 of unit 1 is set to UART1 reception, this pin becomes an RxD3 function pin (refer to Table 11-12). In this case, set channel 2 of unit 1 to operation stop mode or UART2 transmission.

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 11.3 (12) Serial output register m (SOm).

5. When using UART3 transmission and reception in a pair, set channel 3 of unit 1 to UART3 reception (refer to Table 11-12).

Remark X: Don't care

Table 11-12. Relationship between register settings and pins (Channel 3 of unit 1: UART3 reception)

SE13 Note 1	MD132	MD131	TXE13	RXE13	PM14 Note 2	P14 Note 2	Operation mode	Pin Function
								RxD3/P14 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	P14
1	0	1	0	1	1	×	UART3 reception Notes 4, 5	RxD3

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 UART3 reception, this pin becomes an RxD3 function pin. In this case, set channel 2 of unit 1 to operation stop mode or UART3 transmission (refer to Table 11-11).

3. This pin can be set as a port function pin.

4. When using UART3 transmission and reception in a pair, set channel 2 of unit 1 to UART3 transmission (refer to Table 11-11).

5. The SMR12 register of channel 2 of unit 1 must also be set during UART3 reception. For details, refer to 11.6.2 (1) Register setting.

Remark X: Don't care

CHAPTER 12 SERIAL INTERFACE IICA

12.1 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 12-1 shows a block diagram of serial interface IICA.

Figure 12-1. Block Diagram of Serial Interface IICA

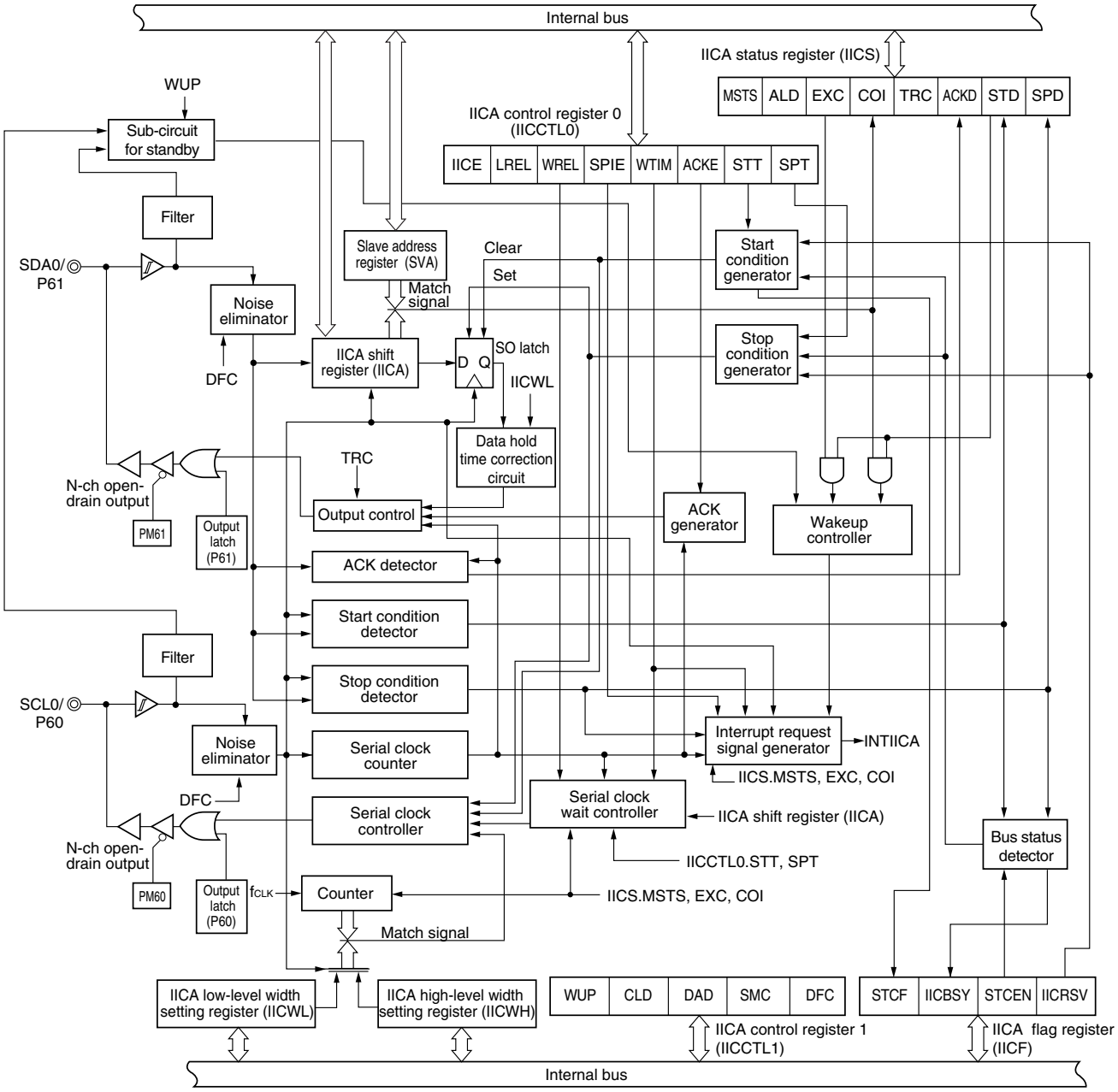
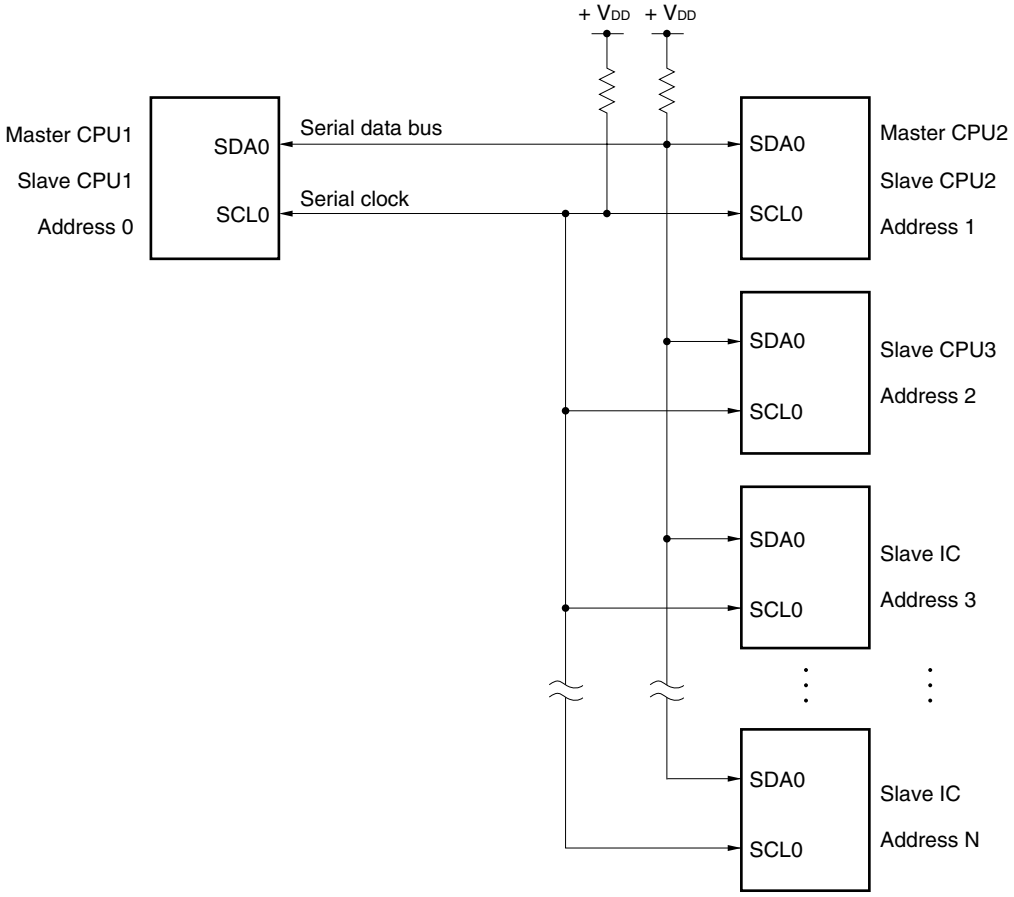


Figure 12-2 shows a serial bus configuration example.

Figure 12-2. Serial Bus Configuration Example Using I²C Bus



12.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 12-1. 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 12-3. 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 12-4. 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	0Note

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 from the master device 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)

12.3 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 12-5. 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	0	ADCEN	IICAEN	SAU1EN	SAU0EN	0	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.

Cautions 1. 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. Be sure to clear bit 6 of the PER0 register to 0.

(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 = 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 12-6. 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 state. 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 state 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 high level, the SDA0 line is low level, and the digital filter is turned on (DFC bit of IICCTL1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE = 1).

Figure 12-6. 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 IICE 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.

<R> **Figure 12-6. 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 IICE 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 12-6. 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 (transmission status), WREL 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.

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.

<R> **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 12-7. 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 12-7. 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).

<R>

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.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 12-7. 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 line'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 12-8. 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 12-9. Format of IICA Control Register 1 (IICCTL1) (1/2)

Address: F0231H After reset: 00H R/W[Note]

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 12-22 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 MSTS, EXC, and COI are “0”, and STD also “0” (communication not entered)) Note 2

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.

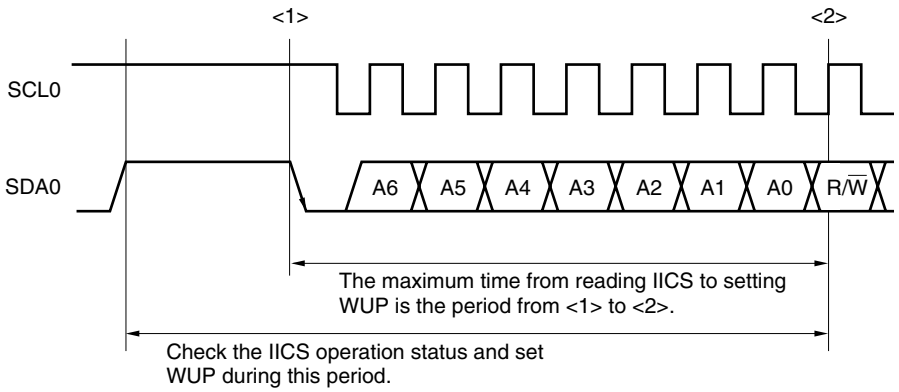


Figure 12-9. Format of IICA Control Register 1 (IICCTL1) (2/2)

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.	

Note Bits 4 and 5 are read-only.

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 12-10. 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 12-11. 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 **12.4.2 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 12-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bit 5-7 of port 6 to 0. When it resets to default value, be sure to set it to 0.

12.4 I^2C Bus Mode Functions

12.4.1 Pin configuration

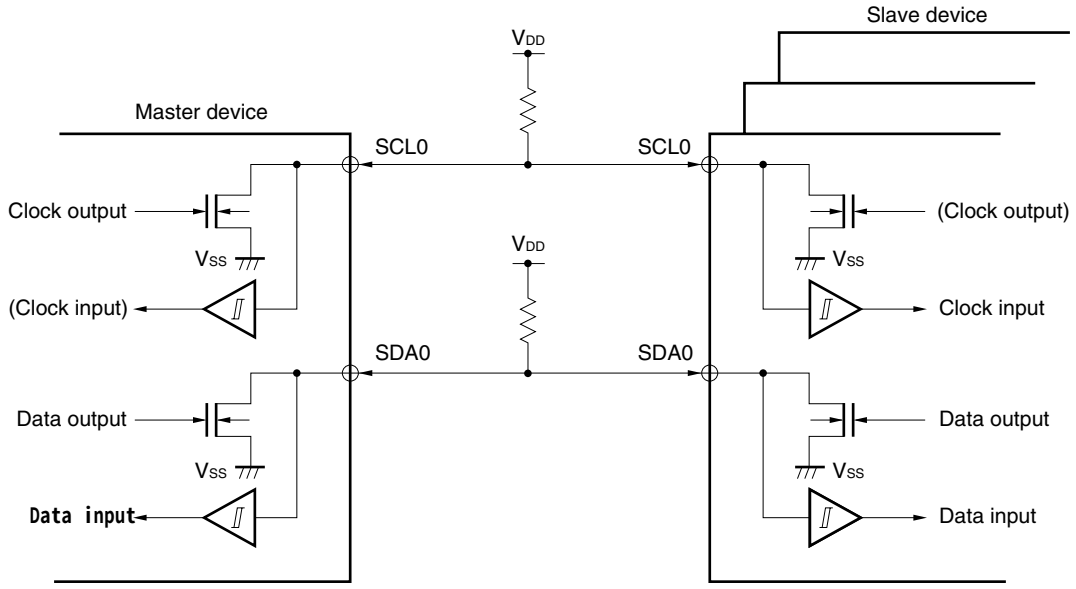
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(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 12-13. Pin Configuration Diagram



<R>12.4.2 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

$$IICWL = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = (\frac{0.53}{\text{Transfer clock}} - t_R - t_F) \times f_{CLK}$$

(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

$$IICWL = 4.7\ \mu s \times f_{CLK}$$
$$IICWH = (5.3\ \mu s - t_R - t_F) \times f_{CLK}$$

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.)

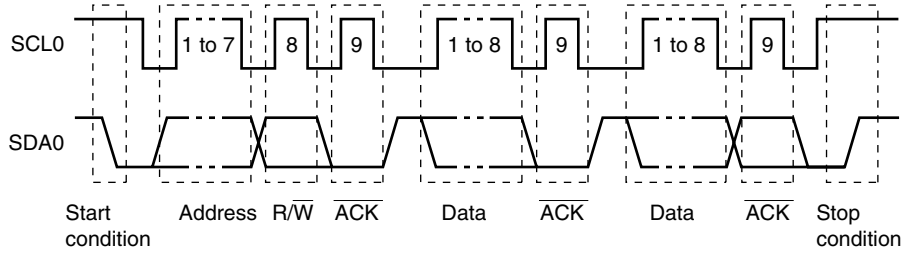
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDA0 and SCL0 signals separately, because they differ depending on the pull-up resistance and wire load.

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

12.5 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 12-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 12-14. I^2C Bus Serial Data Transfer Timing



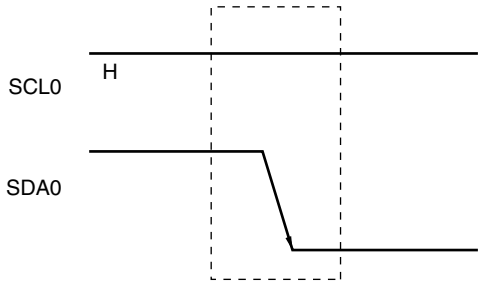
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

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.

12.5.1 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 12-15. 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).

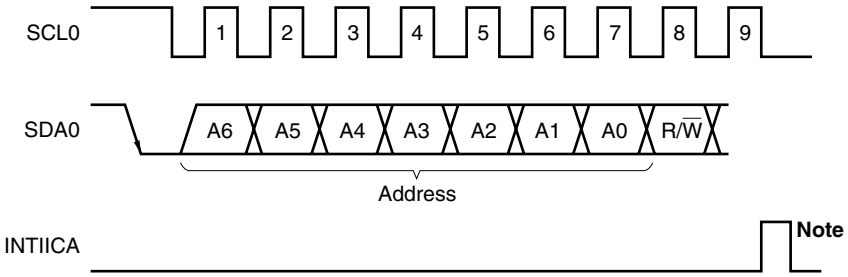
12.5.2 Addresses

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 12-16. 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 **12.5.3 Transfer direction specification** are written to the IICA shift register (IICA). The received addresses are written to IICA.

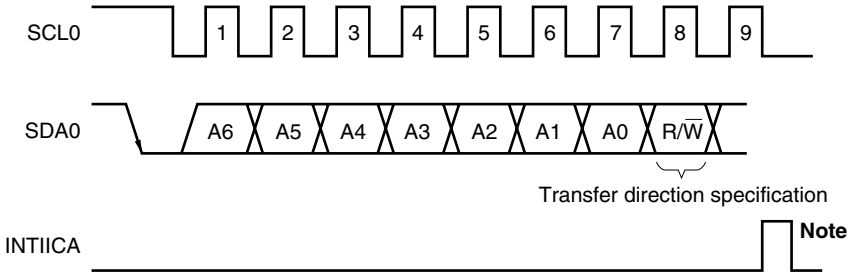
The slave address is assigned to the higher 7 bits of IICA.

12.5.3 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 12-17. Transfer Direction Specification



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

12.5.4 Acknowledge ($\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{\text{ACK}}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{\text{ACK}}$ after transmitting 8-bit data. When $\overline{\text{ACK}}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ and instead generates a stop condition. If a slave does not return $\overline{\text{ACK}}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{\text{ACK}}$ 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.

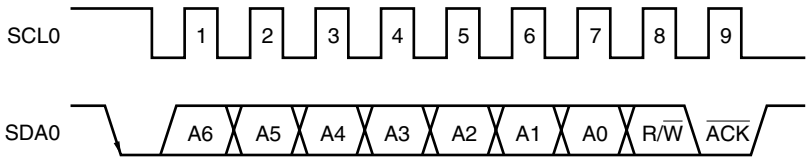
To generate $\overline{\text{ACK}}$, the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ 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 12-18. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated, regardless of the value of ACKE. When an address other than that of the local address is received, $\overline{\text{ACK}}$ is not generated (NACK).

When an extension code is received, $\overline{\text{ACK}}$ is generated if ACKE is set to 1 in advance.

How $\overline{\text{ACK}}$ is generated when data is received differs as follows depending on the setting of the wait timing.

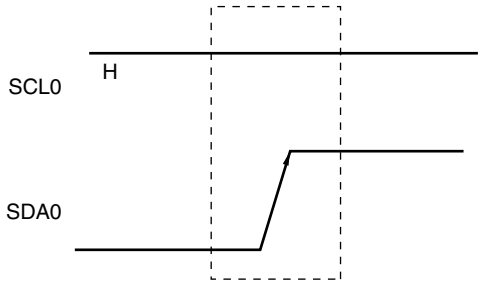
- When 8-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 0):
  By setting ACKE to 1 before releasing the wait state, $\overline{\text{ACK}}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 1):
  $\overline{\text{ACK}}$ is generated by setting ACKE to 1 in advance.

12.5.5 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 12-19. 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.

12.5.6 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 12-20. 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)

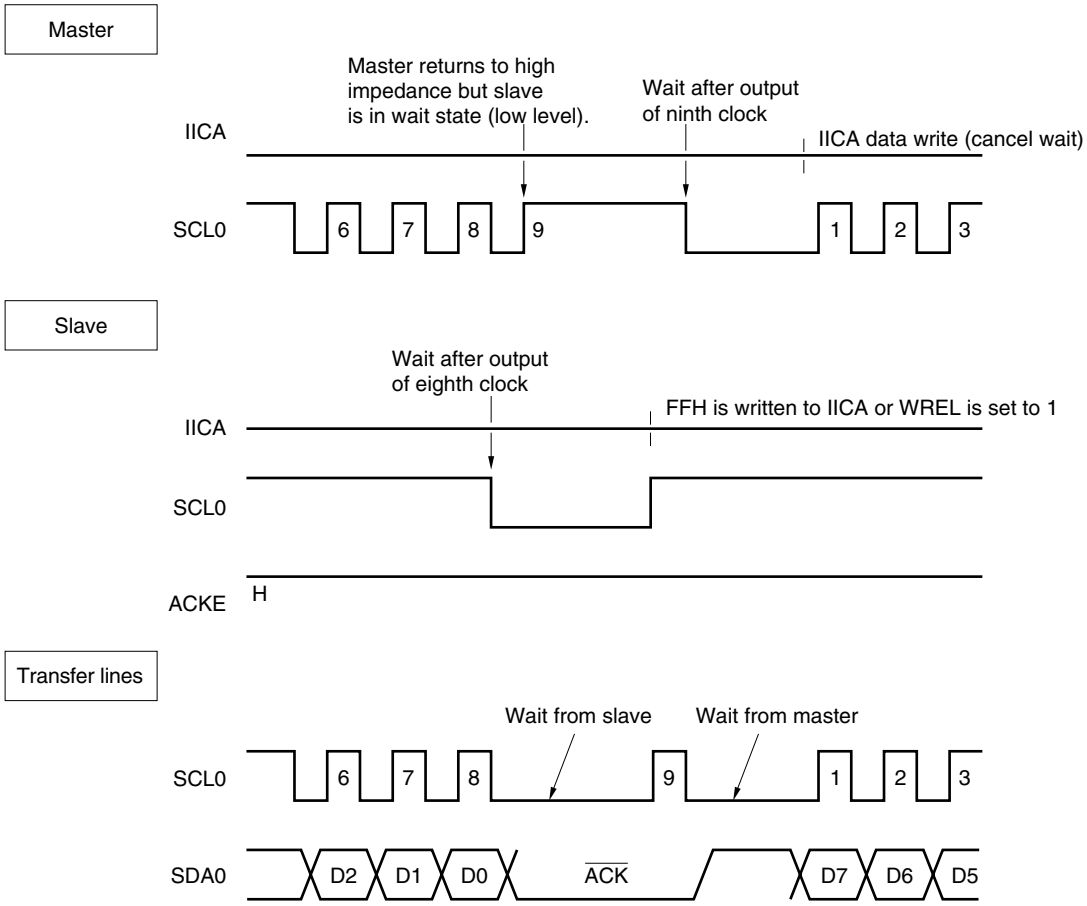
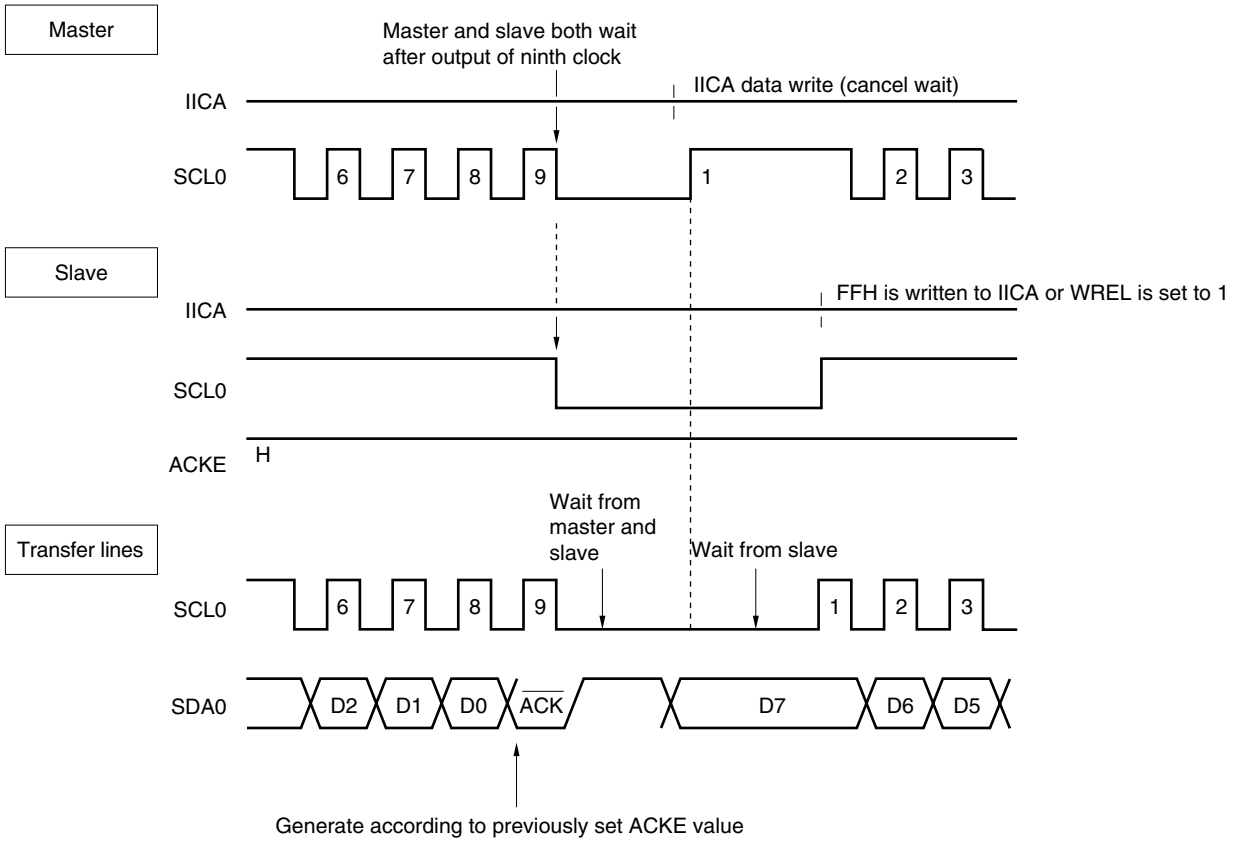


Figure 12-20. Wait (2/2)

(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.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT) of IICCTL0 to 1
- By setting bit 0 (SPT) of IICCTL0 to 1

12.5.7 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, the device exits from communication by setting bit 6 (LREL) of IICCTL0, so that the wait state can be canceled.

Caution If the above wait canceling processing is executed when bit 7 (WUP) of IICCTL1 register = 1, the wait is not canceled.

12.5.8 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 12-2.

Table 12-2. 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 $\overline{\text{ACK}}$ 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).

12.5.9 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.

12.5.10 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.

12.5.11 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) If “11110××0” is set to SVA and “11110××0” is transferred from the master device by a 10-bit address transfer, the results are as follows. 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 state for the next communication operation.

Table 12-3. Bit Definitions of Major Extension Codes

Slave Address	R/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 (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Remark See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

12.5.12 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 **12.5.8 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 12-21. Arbitration Timing Example

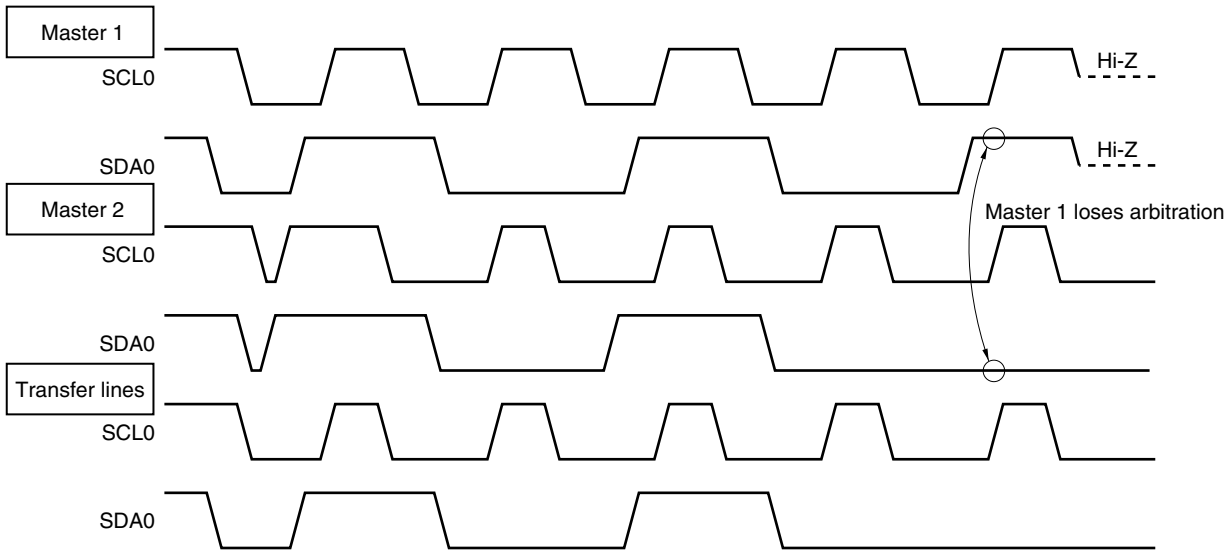


Table 12-4. Status During Arbitration and Interrupt Request Generation Timing

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)

12.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA) when a local address or 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, setting of bit 4 (SPIE) of IICA control register 0 (IICCTL0) , regardless of the wakeup function, 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 12-22 shows the flow for setting WUP = 1 and Figure 12-23 shows the flow for setting WUP = 0 upon an address match.

Figure 12-22. Flow When Setting WUP = 1

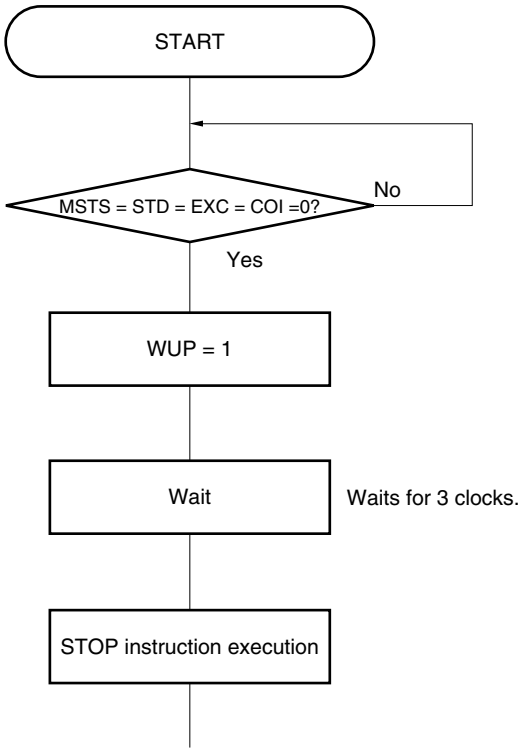
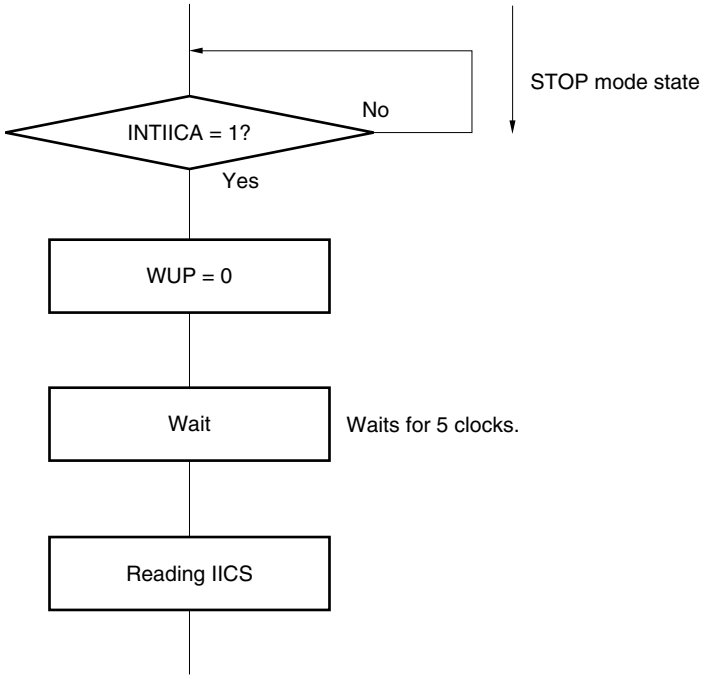


Figure 12-23. Flow When Setting WUP = 0 upon Address Match (Including Extension Code Reception)

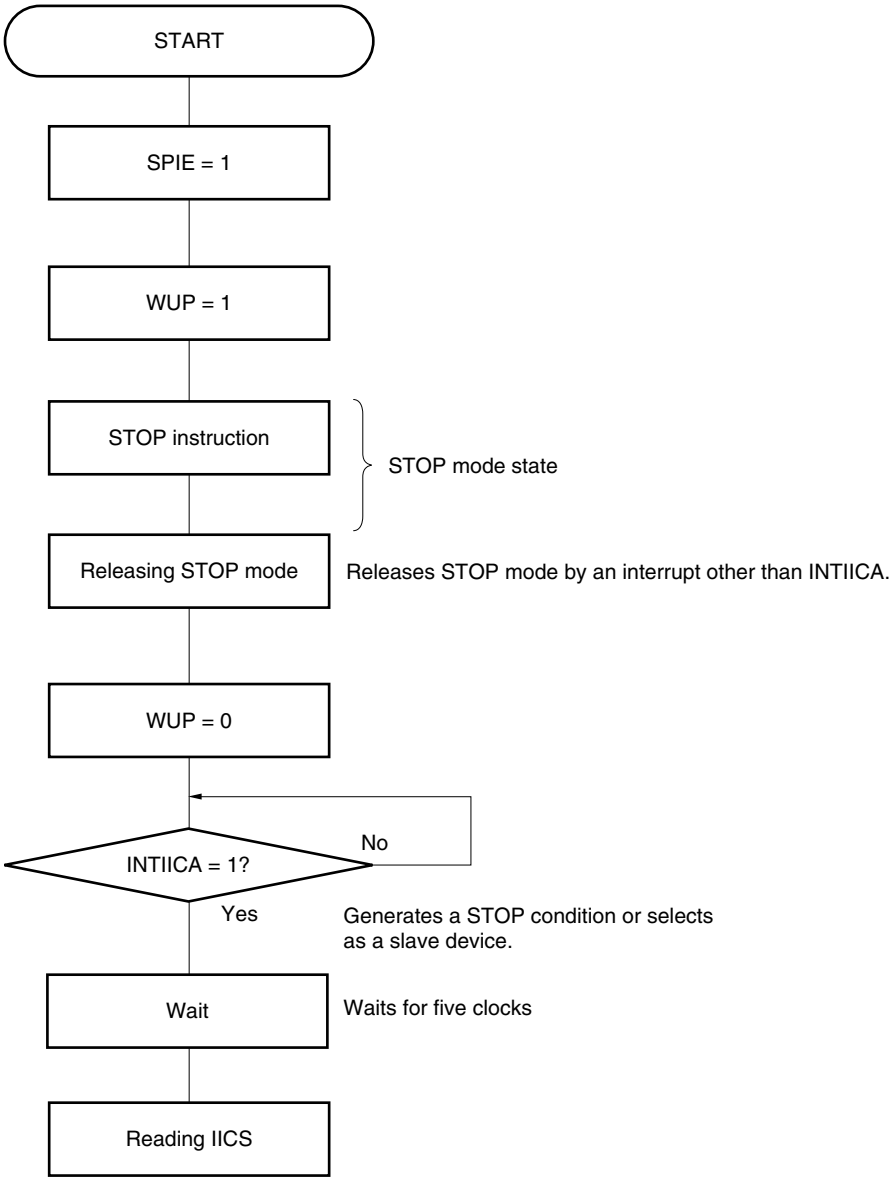


Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

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 12-24
- Slave device operation: Same as the flow shown in Figure 12-23

Figure 12-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA



Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

12.5.14 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 exiting communication).

If bit 1 (STT) of the IICCTL0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register (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 state) ..... communication reservation

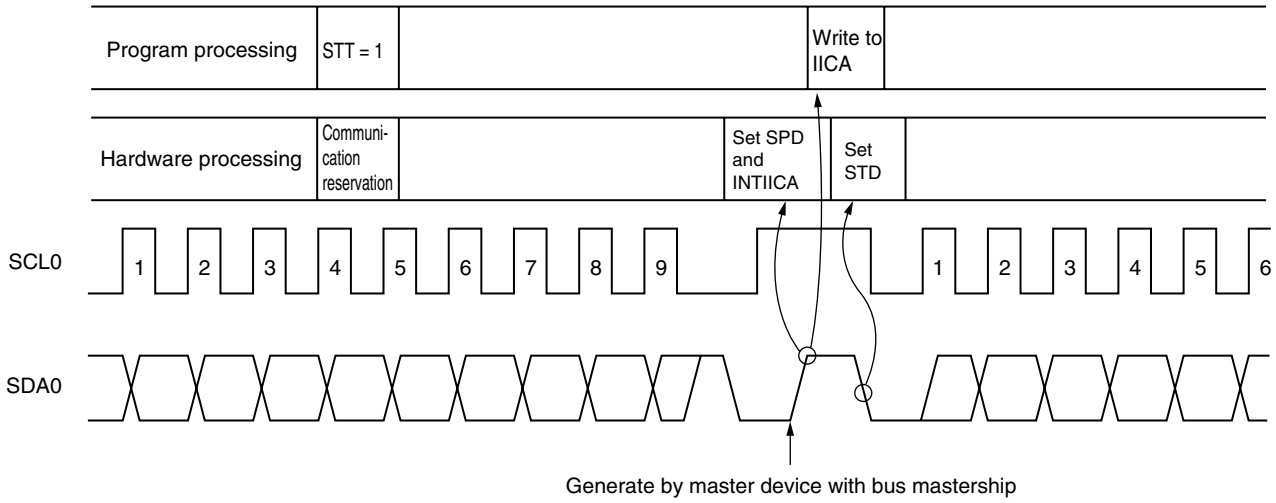
Check whether the communication reservation operates or not by using MSTS (bit 7 of the IICA status register (IICS)) after STT is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT = 1 to checking the MSTS flag:
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clock]

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 12-25 shows the communication reservation timing.

Figure 12-25. 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 12-26. 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 12-26. Timing for Accepting Communication Reservations

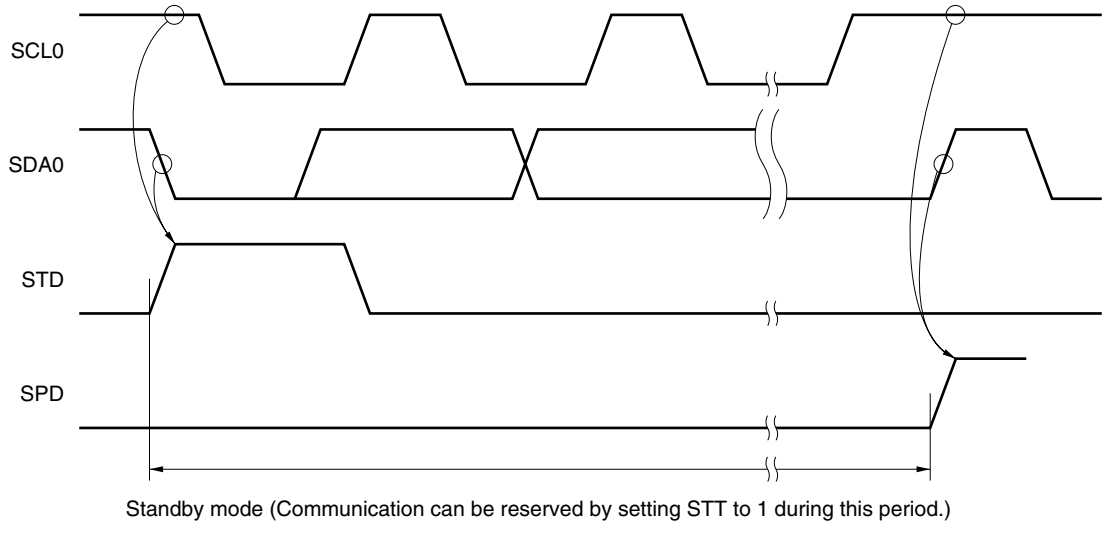
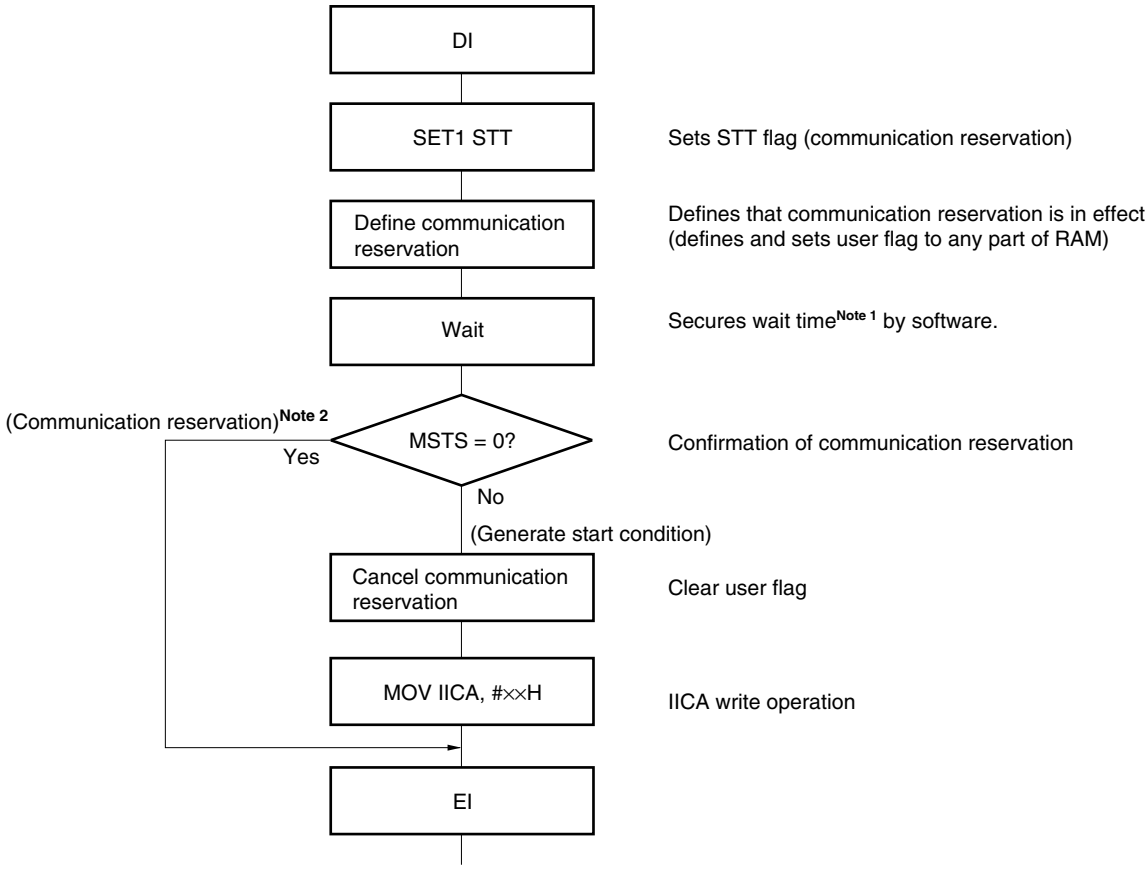


Figure 12-27 shows the communication reservation protocol.

Figure 12-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.

(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clock]

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)

If bit 1 (STT) of IICA control register 0 (IICCTL0) is set to 1 when the device does not participate in 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 the device does not participate in communication.

- 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 exiting 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.

12.5.15 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 IICCTL0) 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.

12.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0R/KF3-L 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/KF3-L 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/KF3-L 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

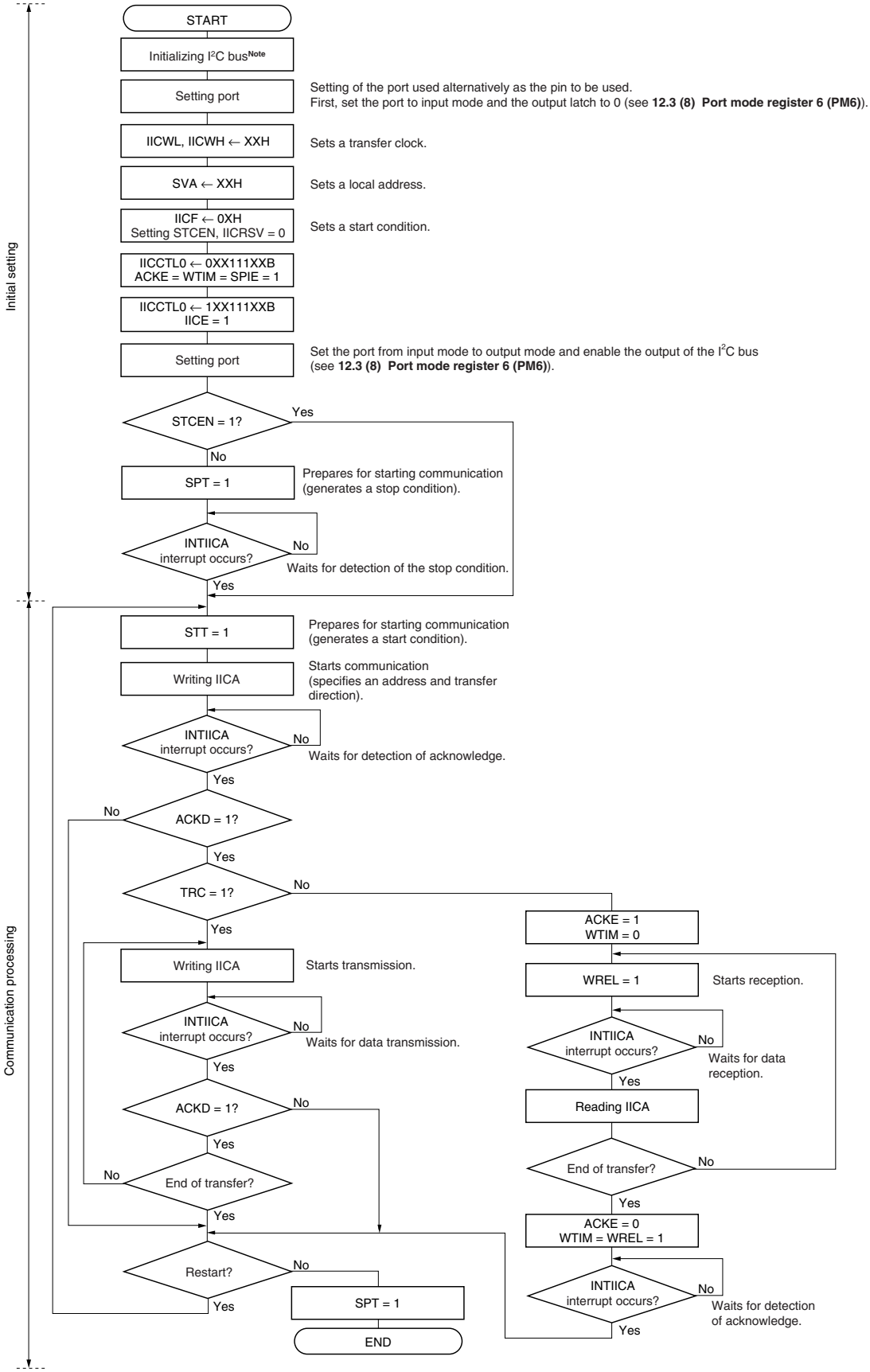
An example of when the 78K0R/KF3-L is used as the I^2C bus slave is shown below.

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 12-28. Master Operation in Single-Master System

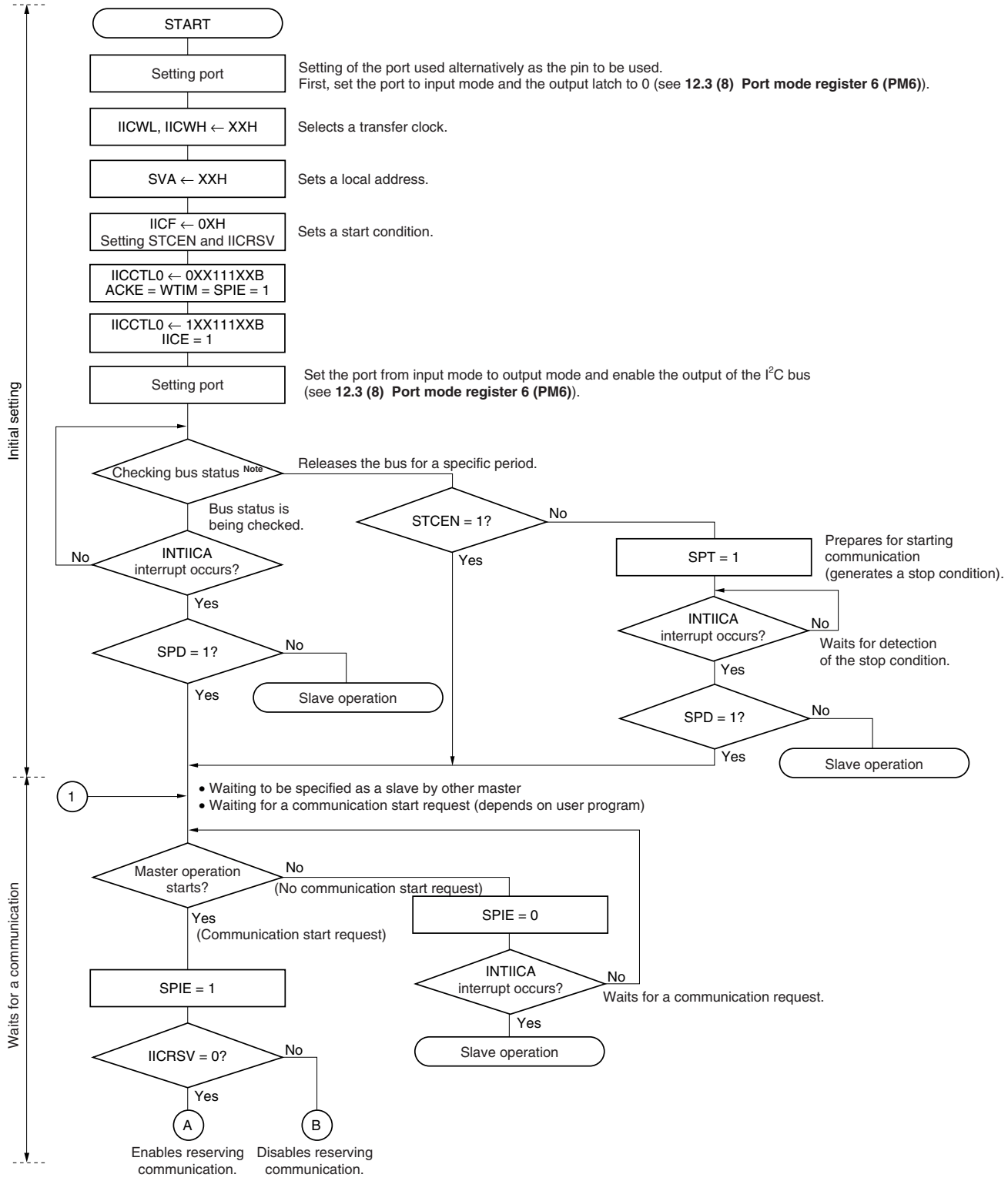


Note Release (SCL0 and SDA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the 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.

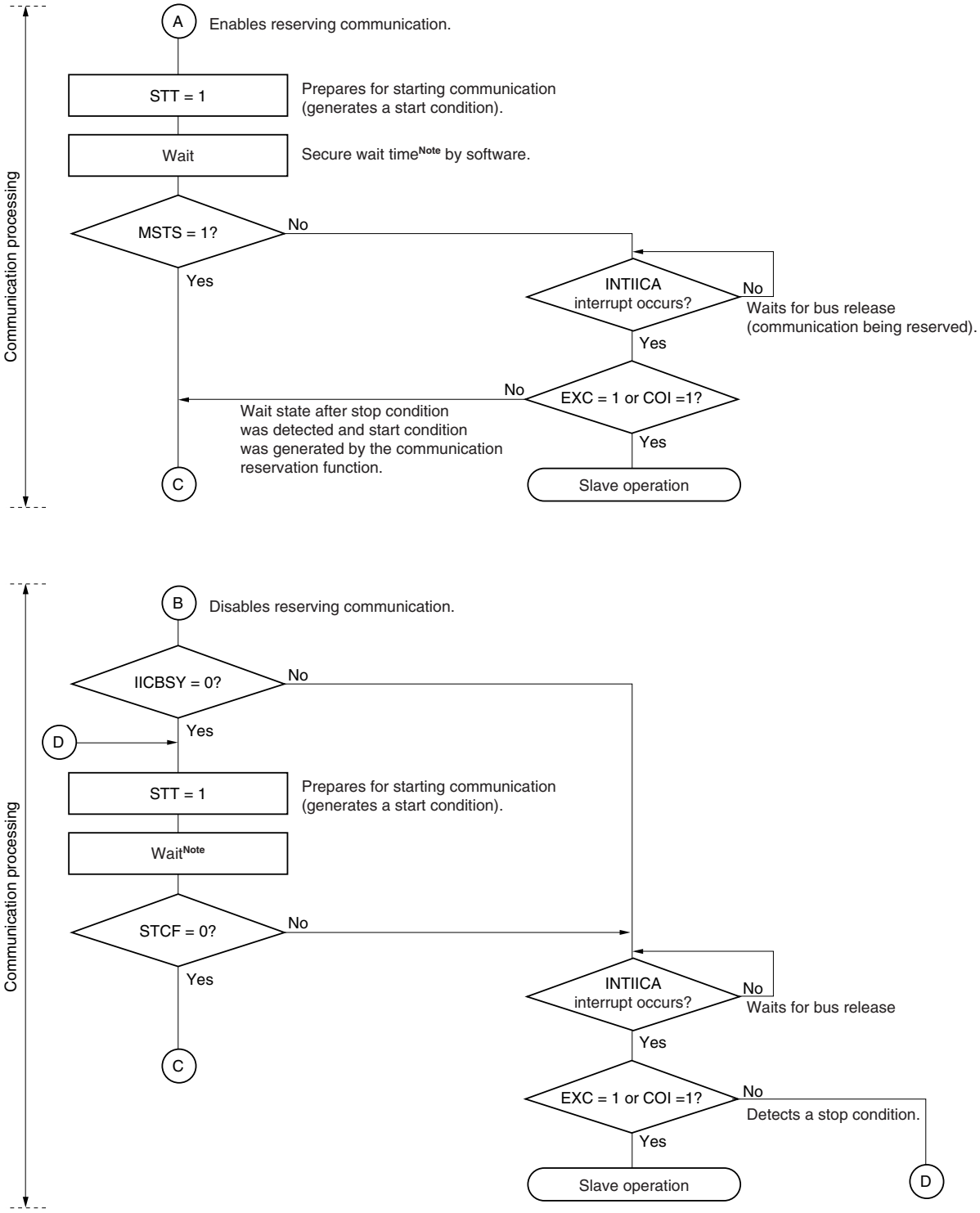
(2) Master operation in multi-master system

Figure 12-29. 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 12-29. 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} + $t_F \times 2$

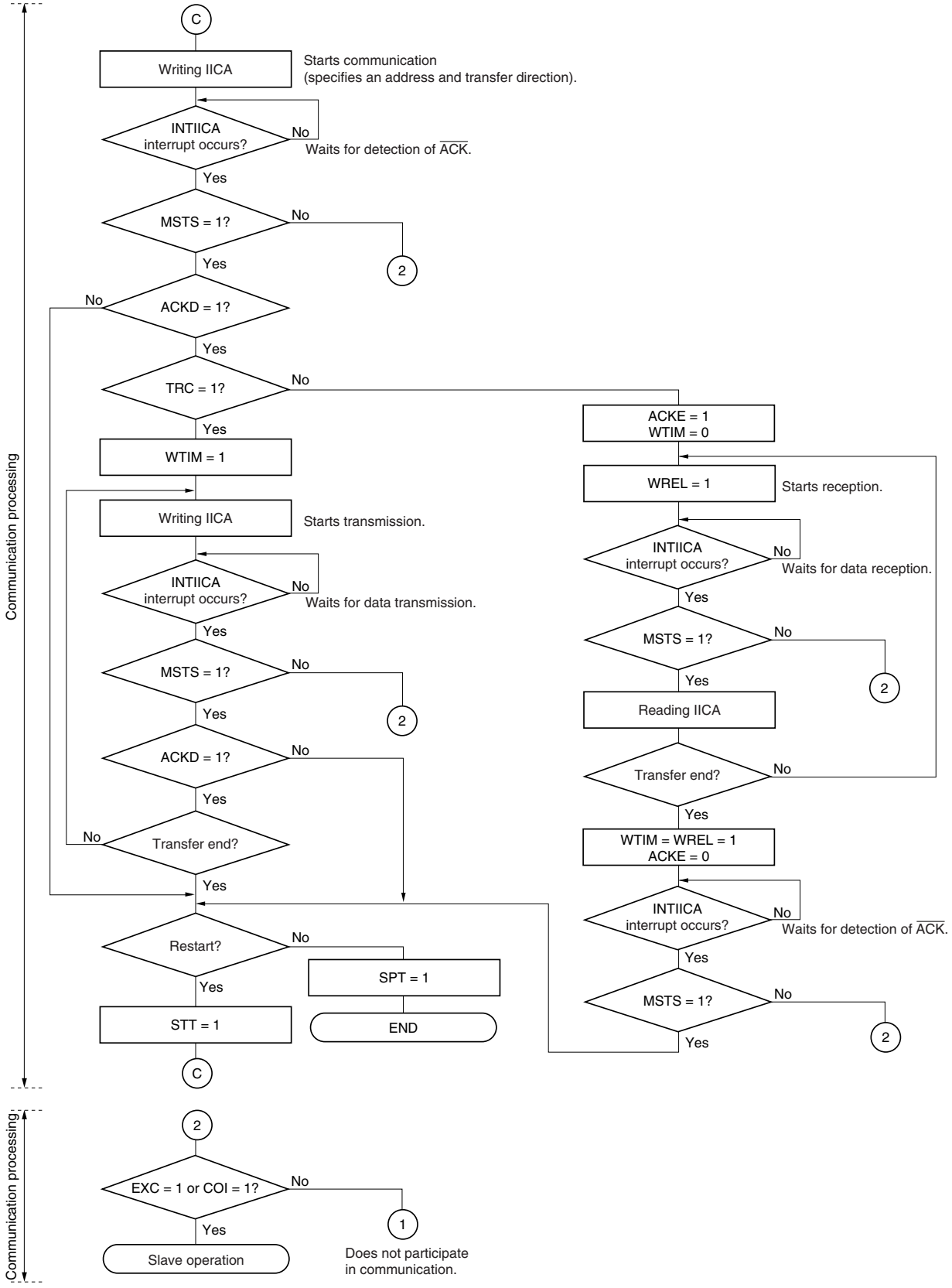
Remark IICWL: IICA low-level width setting register

IICWH: IICA high-level width setting register

t_F: SDA0 and SCL0 signal falling times

f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12-29. 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.

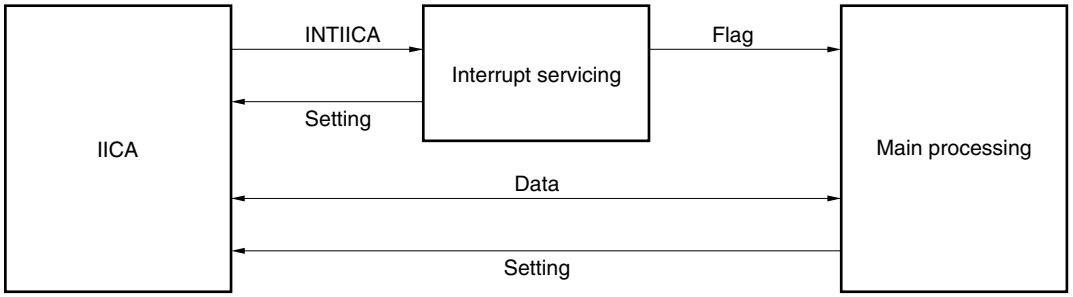
2. To use the device as a master in a multi-master system, read the MSTS bit each time interrupt INTIICA has occurred to check the arbitration result.

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.

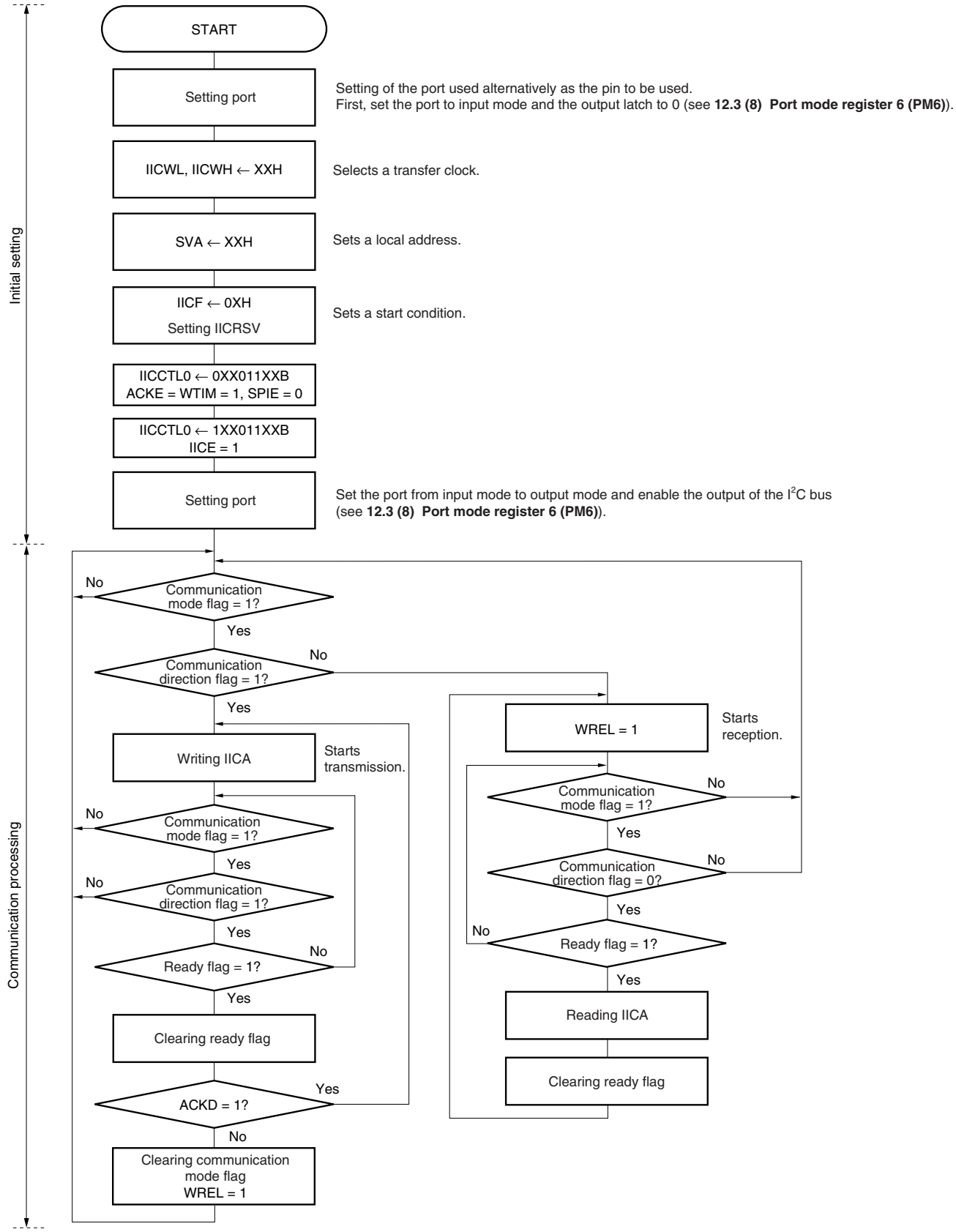
The main processing of the slave operation is explained next.

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 12-30. Slave Operation Flowchart (1)



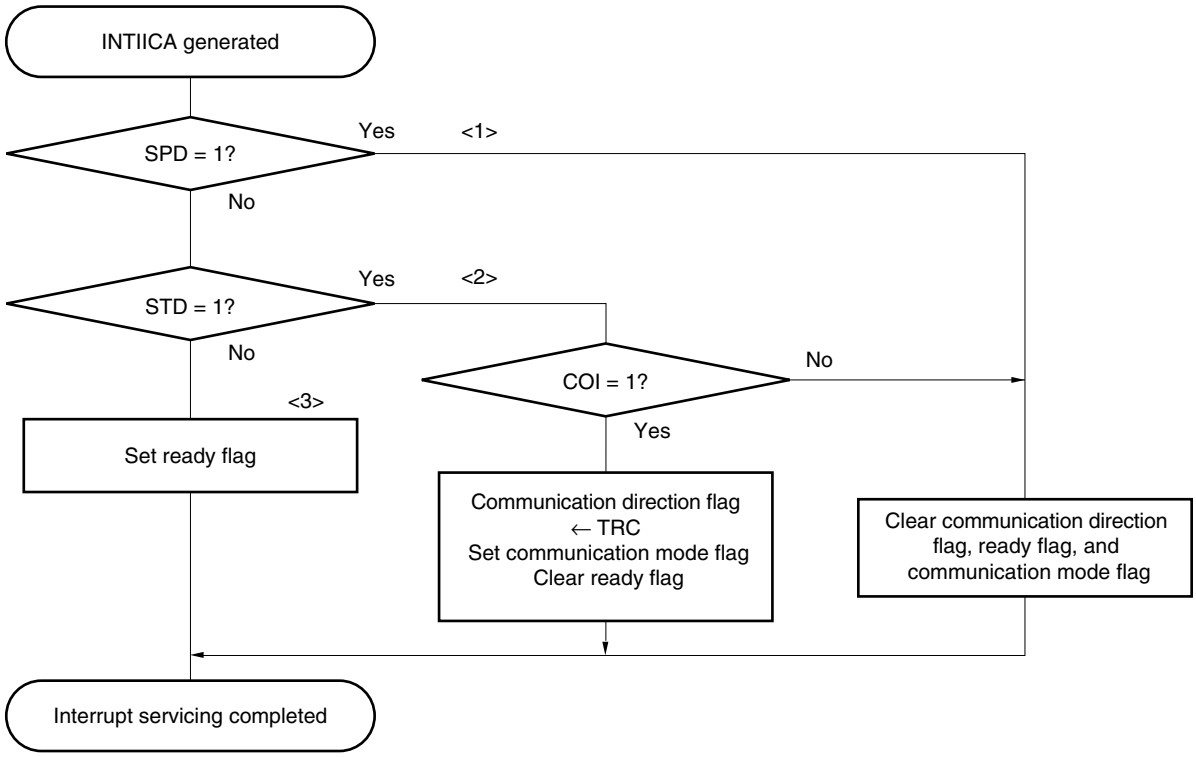
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

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 12-31 Slave Operation Flowchart (2).

Figure 12-31. Slave Operation Flowchart (2)



12.5.17 Timing of I²C interrupt request (INTIICA) occurrence

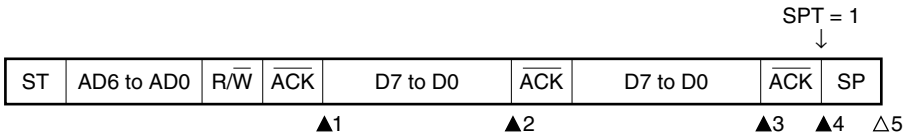
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA, and the value of the IICS register when the INTIICA signal is generated are shown below.

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

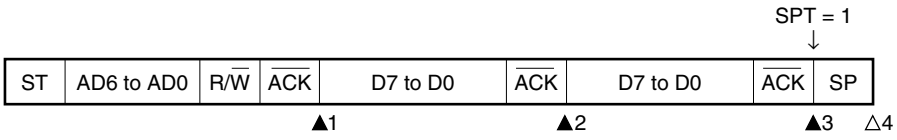


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B
▲3: IICS = 1000×000B (Sets WTIM to 1)[Note]
▲4: IICS = 1000××00B (Sets SPT to 1)
△5: IICS = 00000001B

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

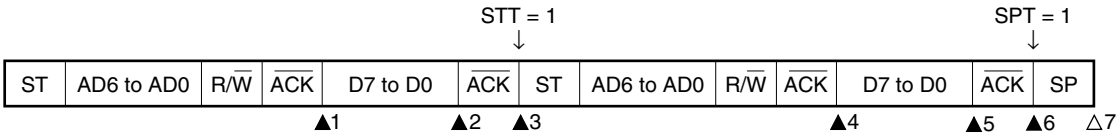


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000××00B (Sets SPT to 1)
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM = 0

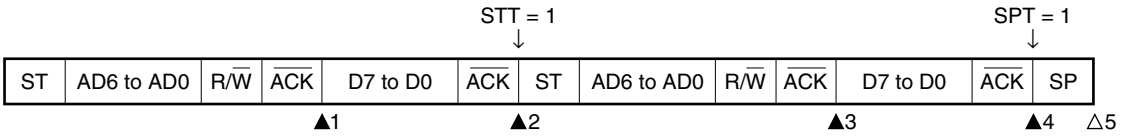


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)[Note 1]
▲3: IICS = 1000××00B (Clears WTIM to 0[Note 2], sets STT to 1)
▲4: IICS = 1000×110B
▲5: IICS = 1000×000B (Sets WTIM to 1)[Note 3]
▲6: IICS = 1000××00B (Sets SPT to 1)
△7: IICS = 00000001B

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

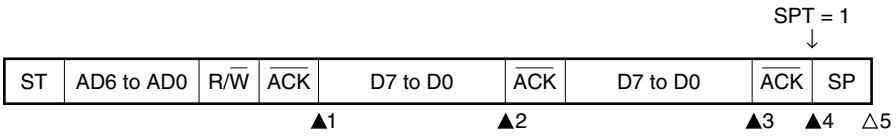


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT to 1)
▲3: IICS = 1000×110B
▲4: IICS = 1000××00B (Sets SPT to 1)
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM = 0

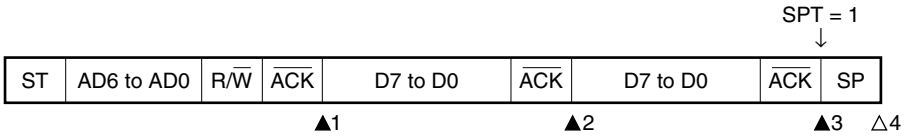


▲1: IICS = 1010×110B
▲2: IICS = 1010×000B
▲3: IICS = 1010×000B (Sets WTIM to 1)Note
▲4: IICS = 1010××00B (Sets SPT to 1)
△5: IICS = 00000001B

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



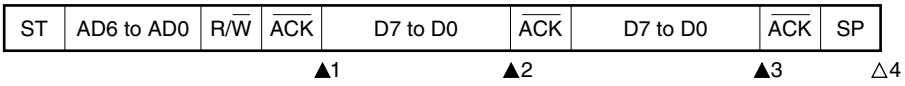
▲1: IICS = 1010×110B
▲2: IICS = 1010×100B
▲3: IICS = 1010××00B (Sets SPT to 1)
△4: IICS = 00001001B

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

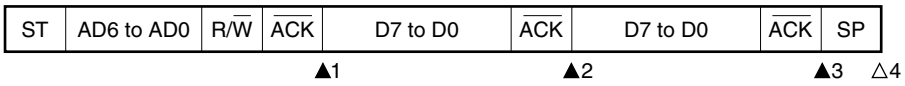
(i) When WTIM = 0



▲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

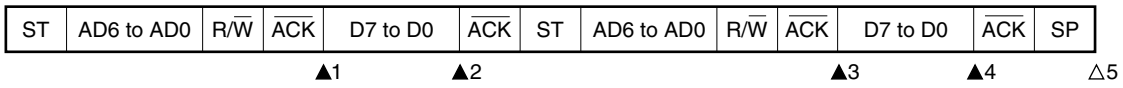


▲1: IICS = 0001×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

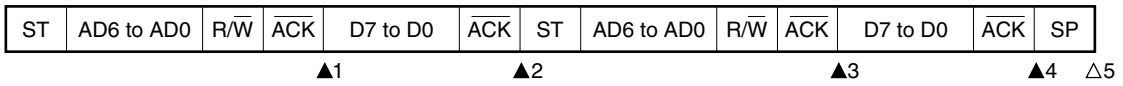
(i) When WTIM = 0 (after restart, matches with SVA)



▲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)

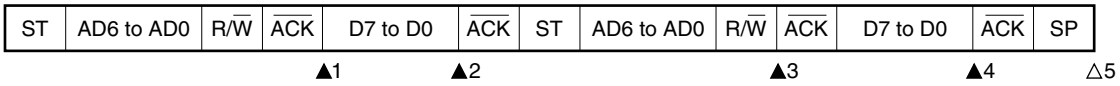


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0001×110B
▲4: IICS = 0001××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

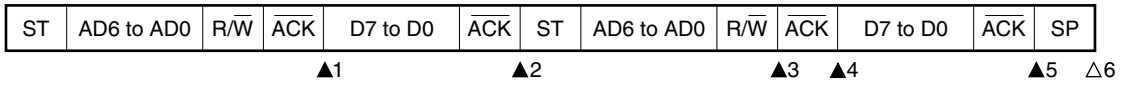
(i) When WTIM = 0 (after restart, does not match address (= extension code))



▲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))

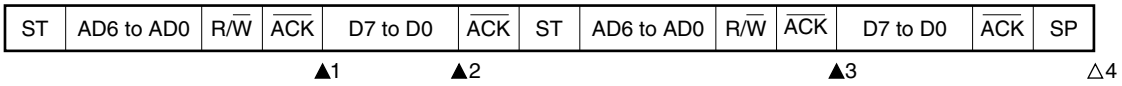


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0010×010B
▲4: IICS = 0010×110B
▲5: IICS = 0010××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

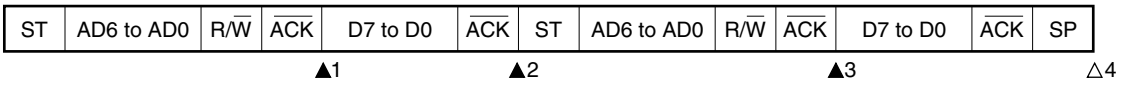
(i) When WTIM = 0 (after restart, does not match address (= not extension code))



▲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

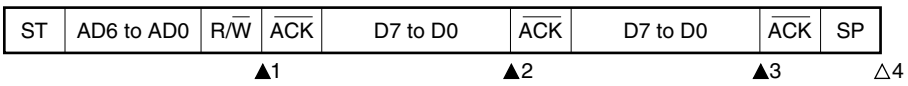
Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(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

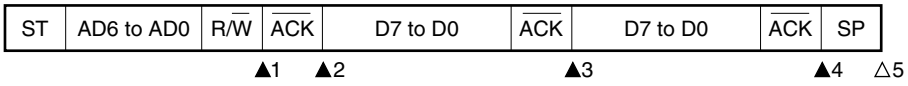
(i) When WTIM = 0



▲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

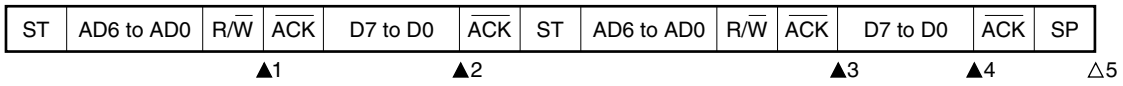


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

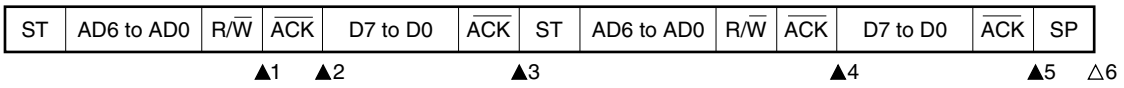
(i) When WTIM = 0 (after restart, matches SVA)



▲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)

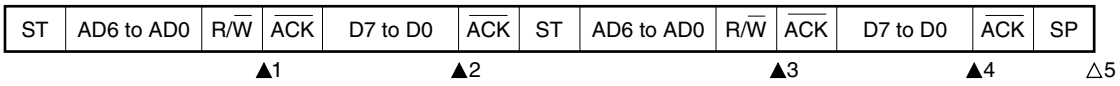


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0001×110B
▲5: IICS = 0001××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

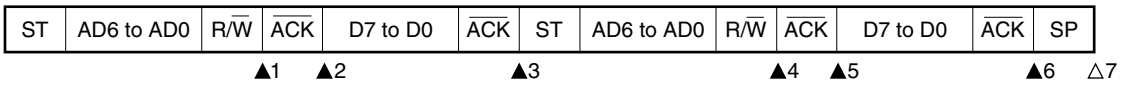
(i) When WTIM = 0 (after restart, extension code reception)



▲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

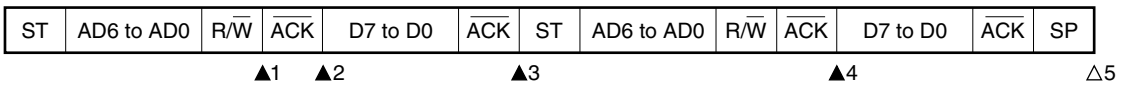
(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(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))

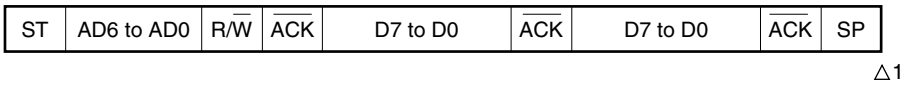


▲1: IICS = 00100010B
▲2: IICS = 00100110B
▲3: IICS = 00100×00B
▲4: IICS = 00000110B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS = 00000001B

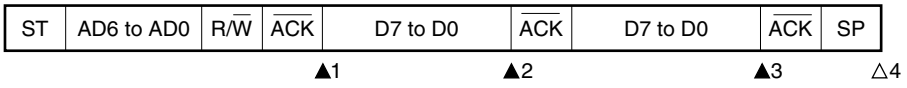
Remark △: Generated only when SPIE = 1

(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

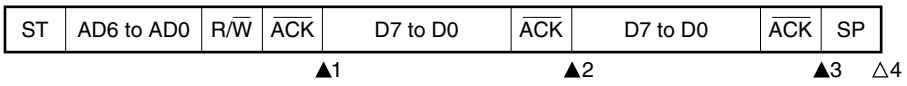
(i) When WTIM = 0



▲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

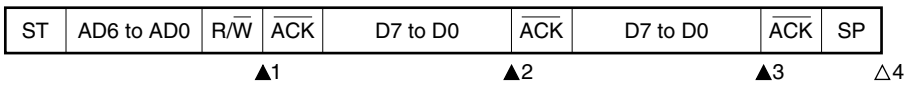


▲1: IICS = 0101×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

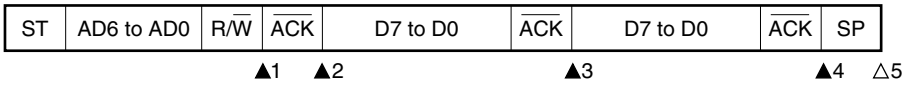
(i) When WTIM = 0



▲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



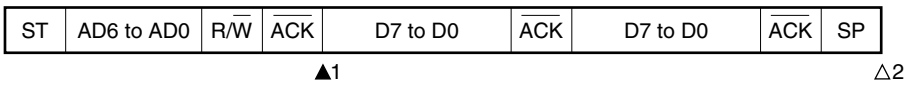
▲1: IICS = 0110×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

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.

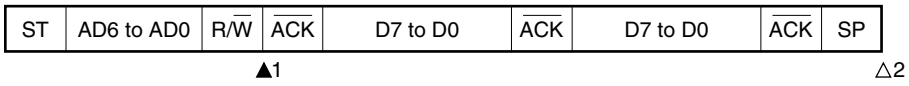
(a) When arbitration loss occurs during transmission of slave address data (when WTIM = 1)



▲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

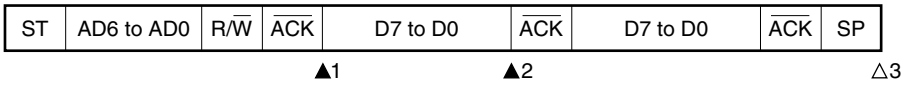
Remark ▲: Always generated

△: Generated only when SPIE = 1

×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM = 0



▲1: IICS = 10001110B

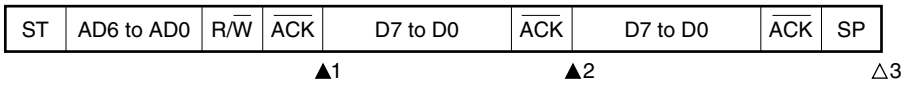
▲2: IICS = 01000000B

△3: IICS = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE = 1

(ii) When WTIM = 1

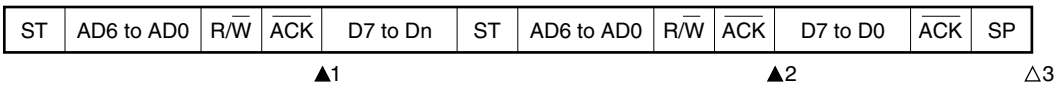


▲1: IICS = 10001110B
▲2: IICS = 01000100B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(d) When loss occurs due to restart condition during data transfer

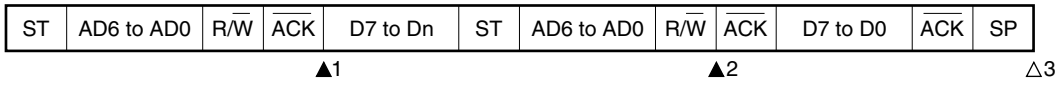
(i) Not extension code (Example: unmatches with SVA)



▲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

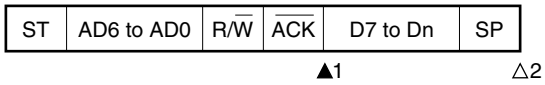
(ii) Extension code



▲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

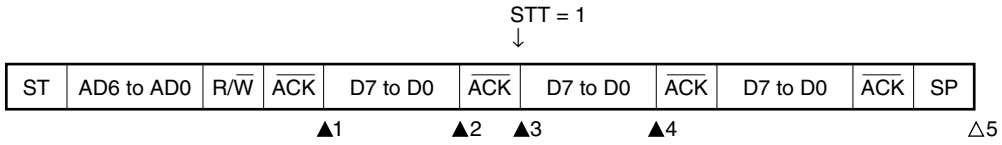


▲1: IICS = 10000110B
△2: IICS = 01000001B

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

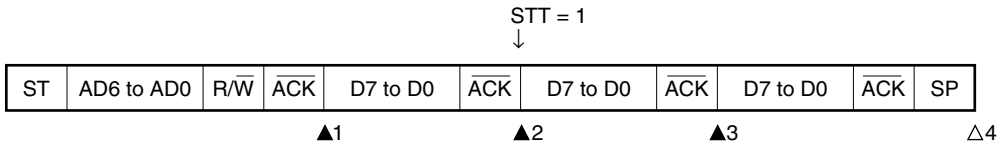
(i) When WTIM = 0



▲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

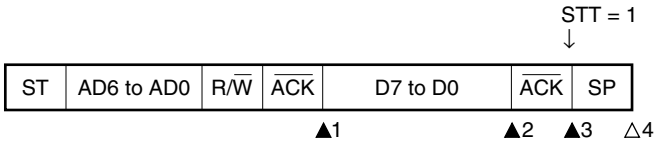


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets STT to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

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

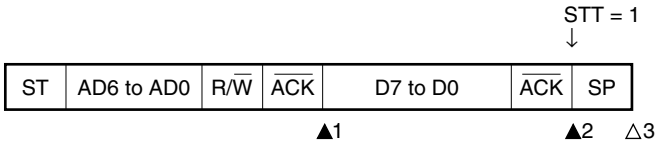
(i) When WTIM = 0



▲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

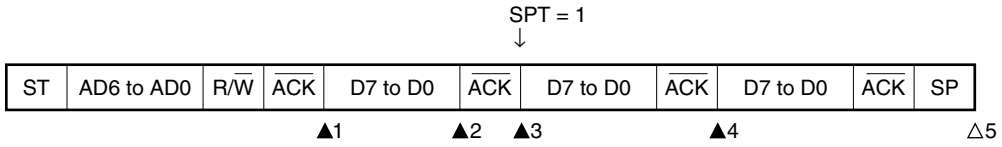


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT to 1)
△3: IICS = 01000001B

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

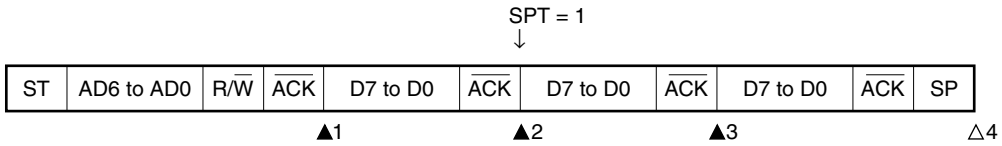
(i) When WTIM = 0



▲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

12.6 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.

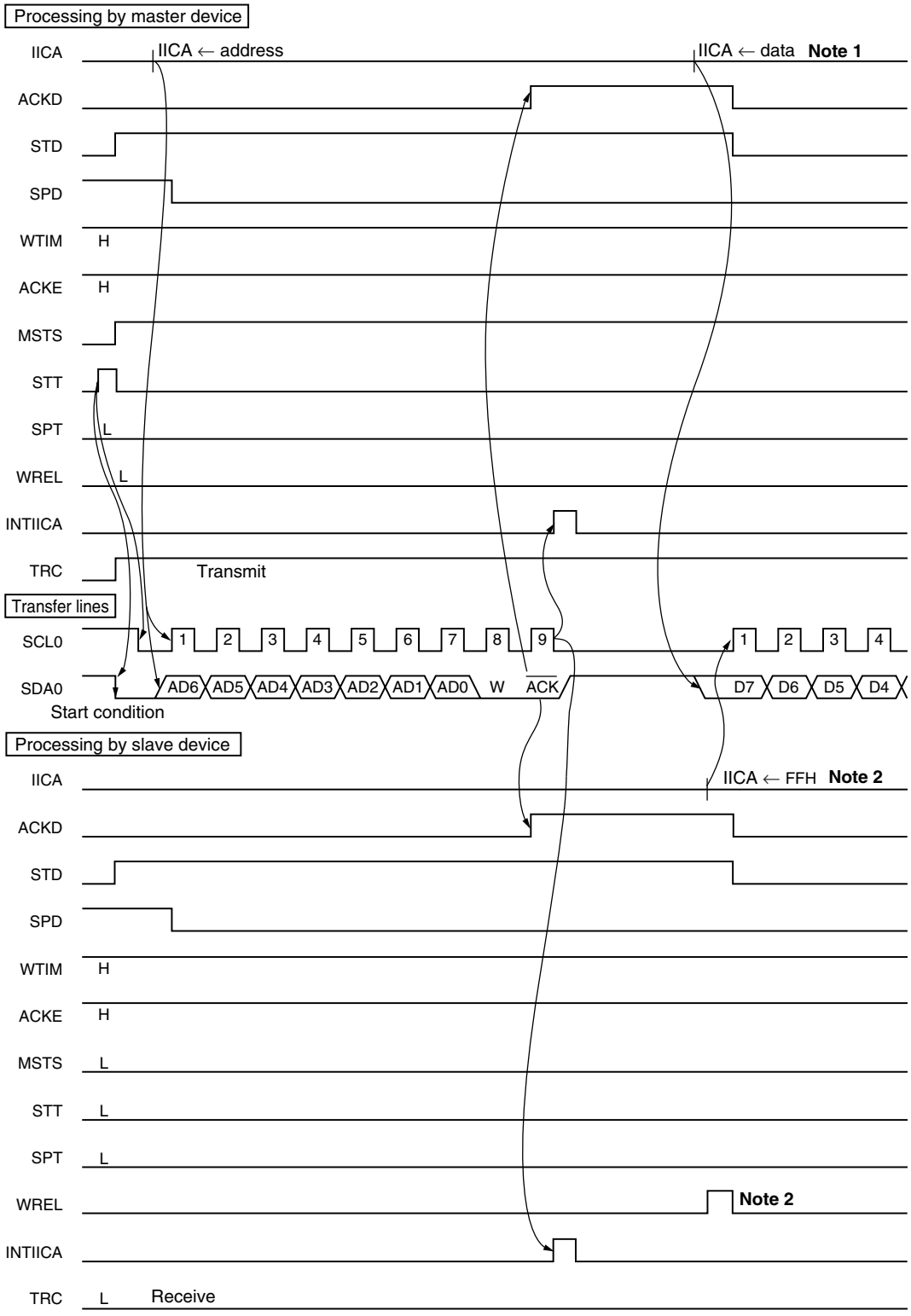
Figures 12-32 and 12-33 show timing charts of the data communication.

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 12-32. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(1) Start condition ~ address

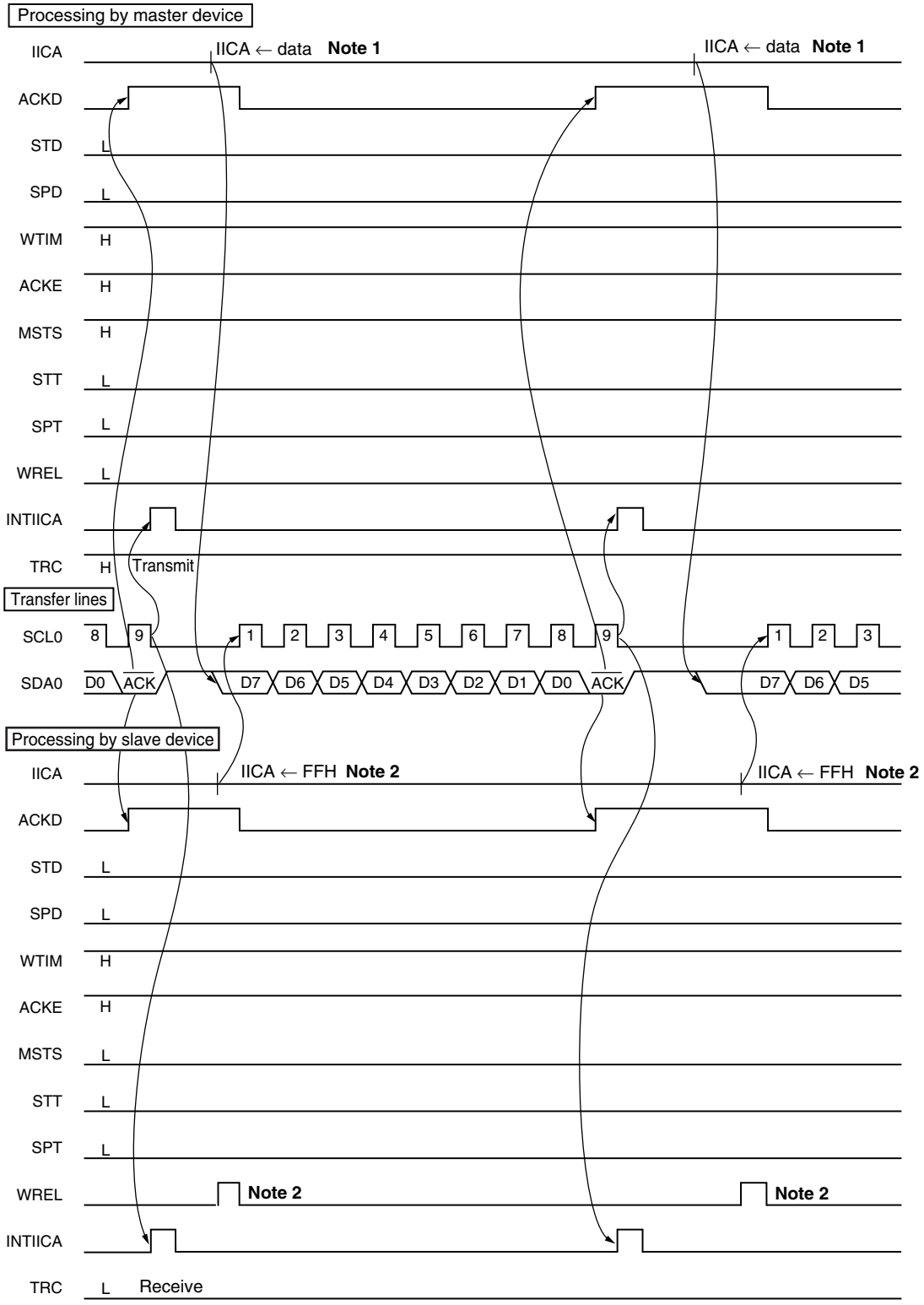


Notes 1. Write data to IICA, not setting WREL, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL.

Figure 12-32. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(2) Data

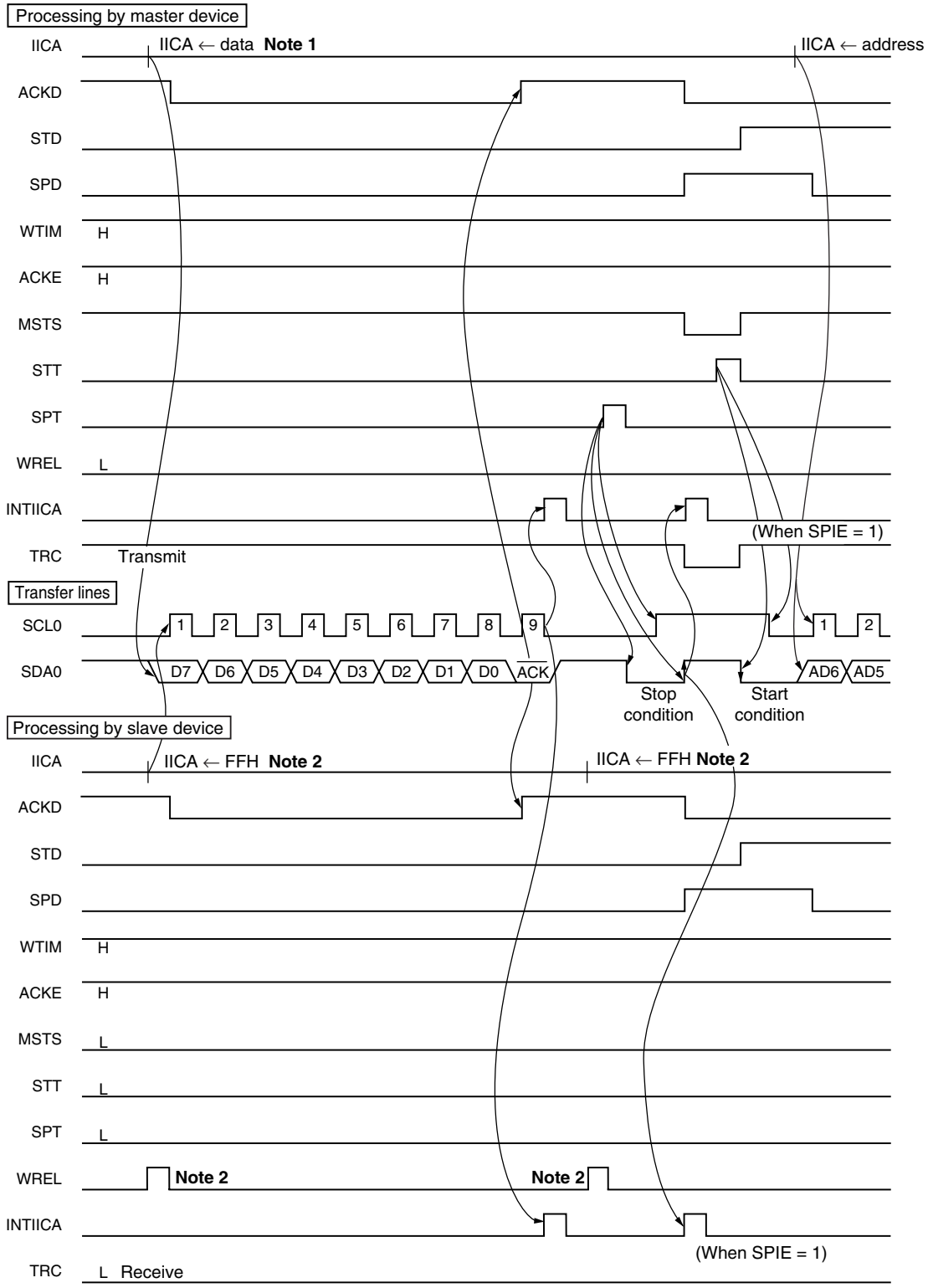


Notes 1. Write data to IICA, not setting WREL, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL.

Figure 12-32. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition

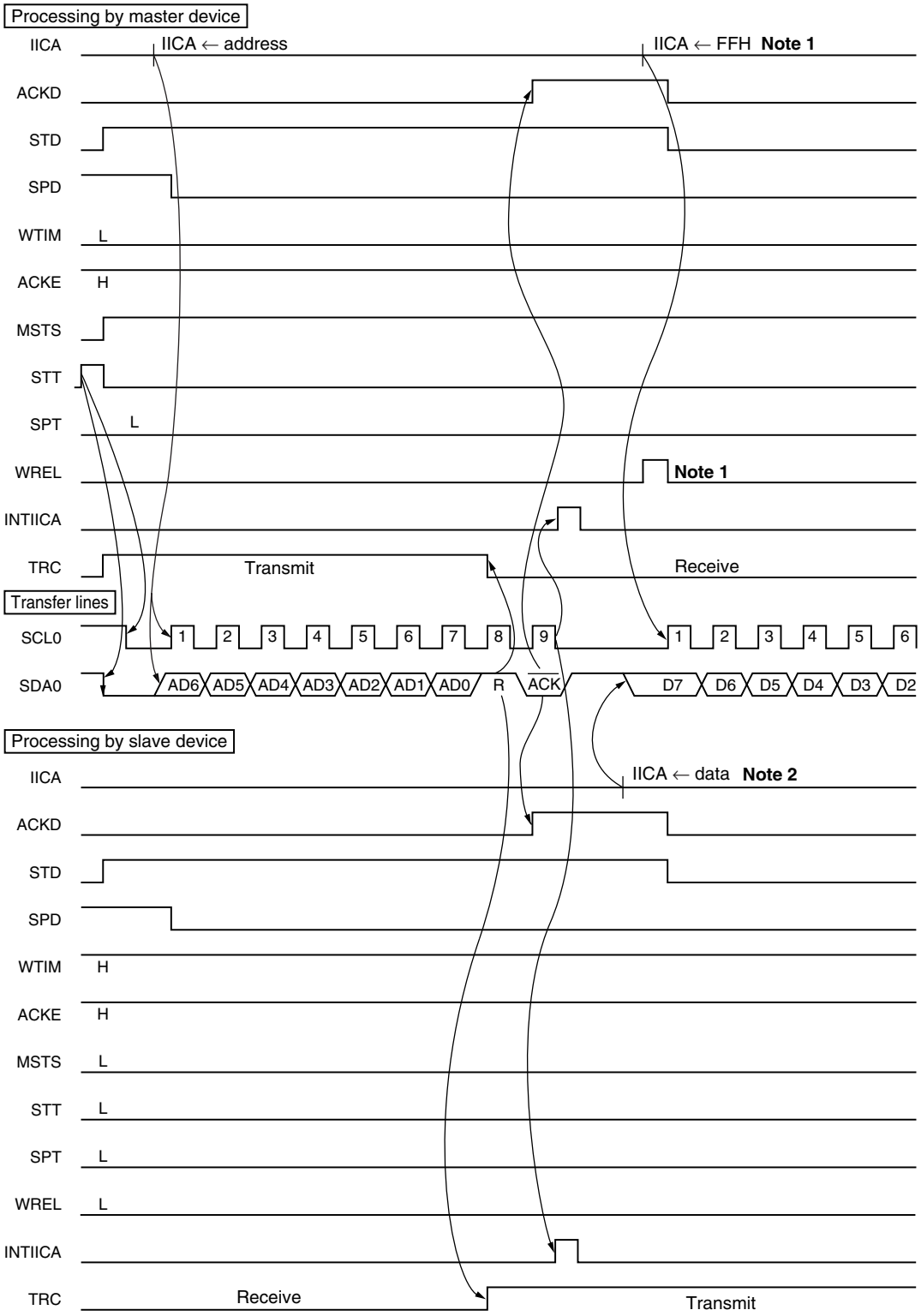


Notes 1. Write data to IICA, not setting WREL, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL.

Figure 12-33. 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

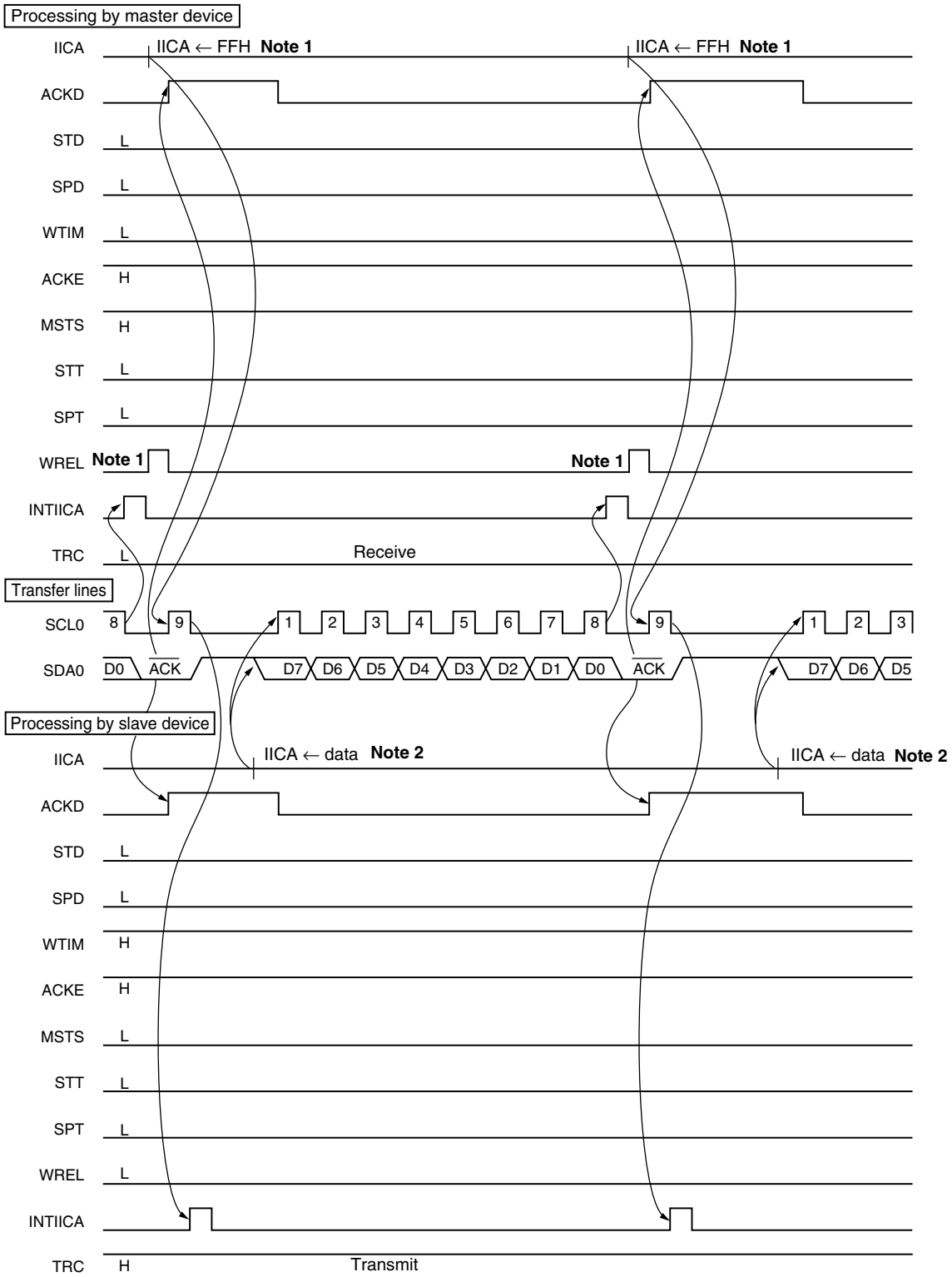


Notes 1. To cancel master wait, write “FFH” to IICA or set WREL.

2. Write data to IICA, not setting WREL, in order to cancel a wait state during slave transmission.

Figure 12-33. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Data

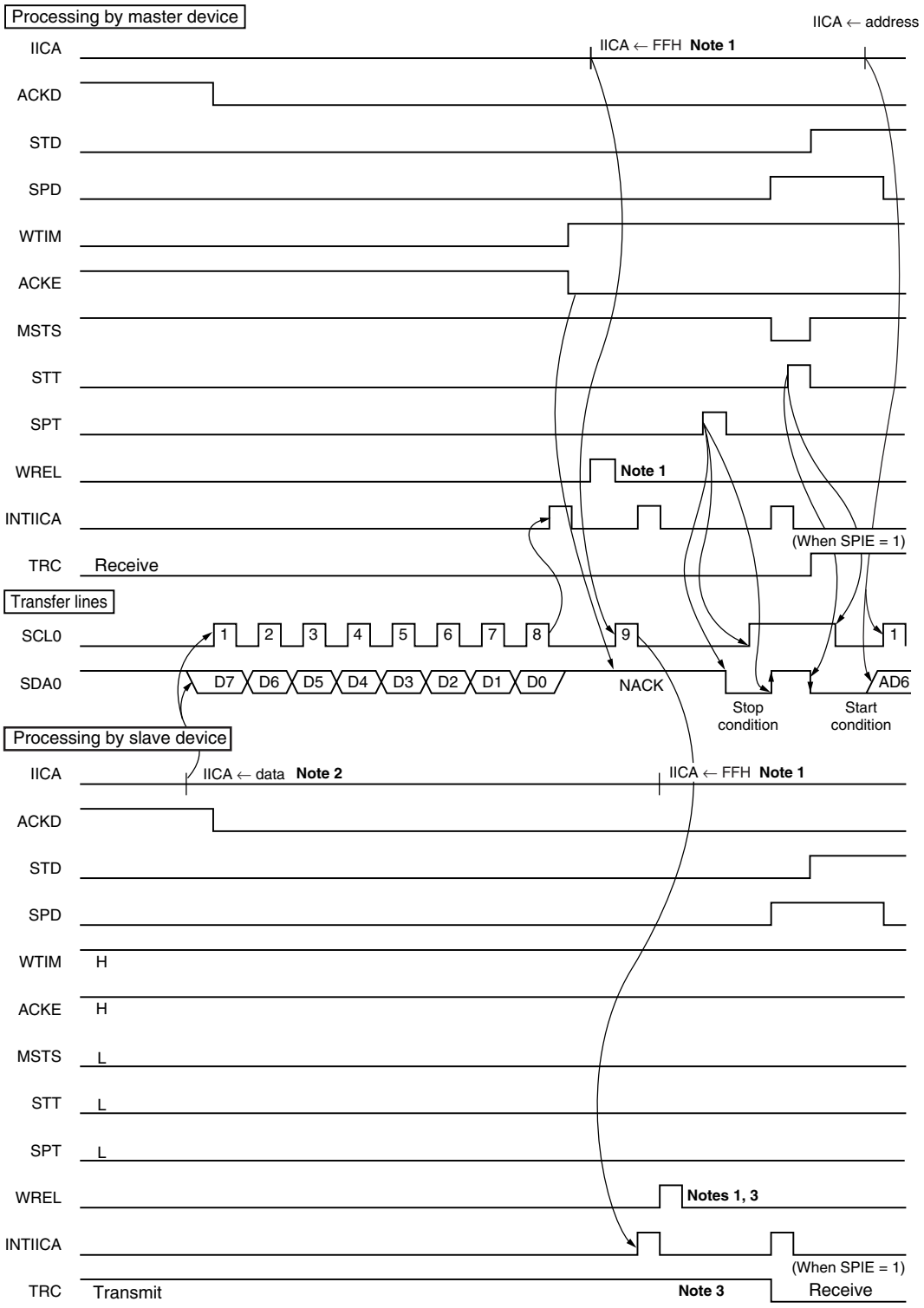


Notes 1. To cancel master wait, write “FFH” to IICA or set WREL.

2. Write data to IICA, not setting WREL, in order to cancel a wait state during slave transmission.

Figure 12-33. 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) Stop condition



Notes 1. To cancel wait, write “FFH” to IICA or set WREL.

2. Write data to IICA, not setting WREL, in order to cancel a wait state during slave transmission.

3. If a wait state during slave transmission is canceled by setting WREL, TRC will be cleared.

CHAPTER 13 USB FUNCTION CONTROLLER (USBF)

The 78K0R/KC3-L, KE3-L have an internal USB function controller (USBF) conforming to the Universal Serial Bus Specification. Data communication using the polling method is performed between the USB function controller and external host device by using a token-based protocol.

13.1 Overview

- Conforms to the Universal Serial Bus Specification
- Supports 12 Mbps (full-speed) transfer
- Endpoint for transfer incorporated

Endpoint Name	FIFO Size (Bytes)	Transfer Type	Remark
Endpoint0 Read	64	Control transfer	–
Endpoint0 Write	64	Control transfer	–
Endpoint1	64 × 2	Bulk 1 transfer (IN)	2-buffer configuration
Endpoint2	64 × 2	Bulk 1 transfer (OUT)	2-buffer configuration
Endpoint3	64 × 2	Bulk 2 transfer (IN)	2-buffer configuration
Endpoint4	64 × 2	Bulk 2 transfer (OUT)	2-buffer configuration
Endpoint7	64	Interrupt 1 transfer	Single buffer configuration
Endpoint8	64	Interrupt 2 transfer	Single buffer configuration

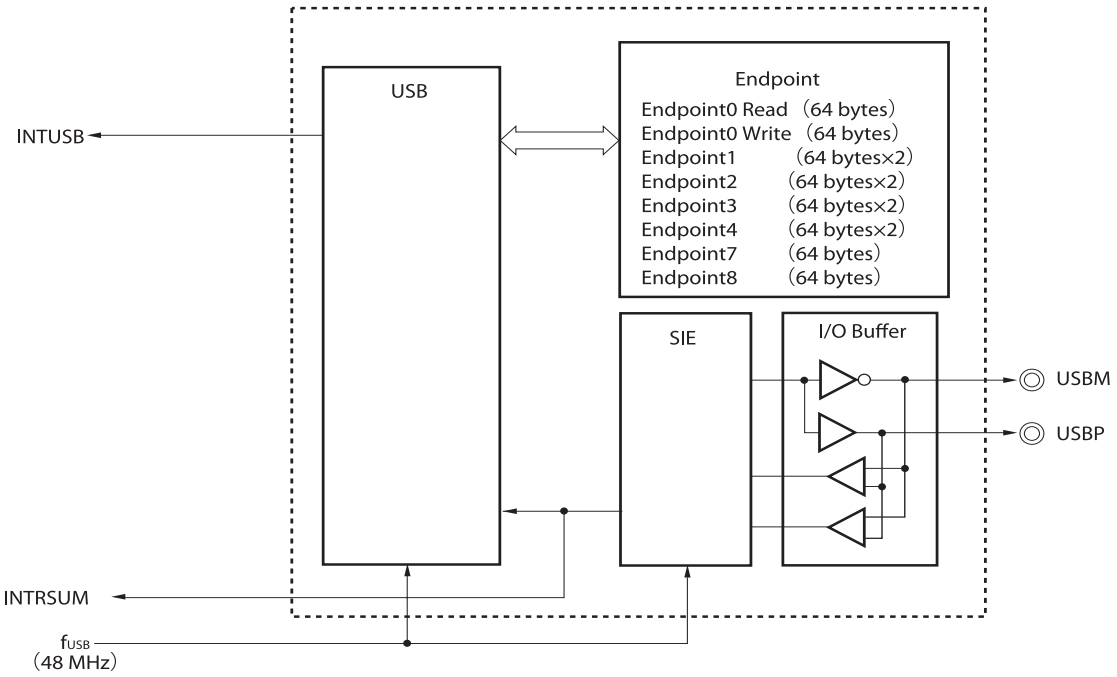
- **USB Clock: Internal clock (20 MHz external clock ÷ Internal 5 divider × Internal 12 multiplier = Internal 48 MHz/**
  external 16 MHz ÷ Internal 4 divider × Internal 12 multiplier = Internal 48 MHz/
  external 12 MHz ÷ Internal 2 divider × Internal 8 multiplier = Internal 48 MHz

<R> **Caution The registers listed in 13.6.2 USB function controller register must be accessed after specifying that the internal clock is supplied to the USB function controller without fail, or else the results of reading and writing can not be guaranteed.**

13.2 Configuration

13.2.1 Block diagram

Figure 13-1. Block Diagram of USB Function Controller



Remarks Inside broken lines: These functions are included in the USB function controller.

13.3 External Circuit Configuration

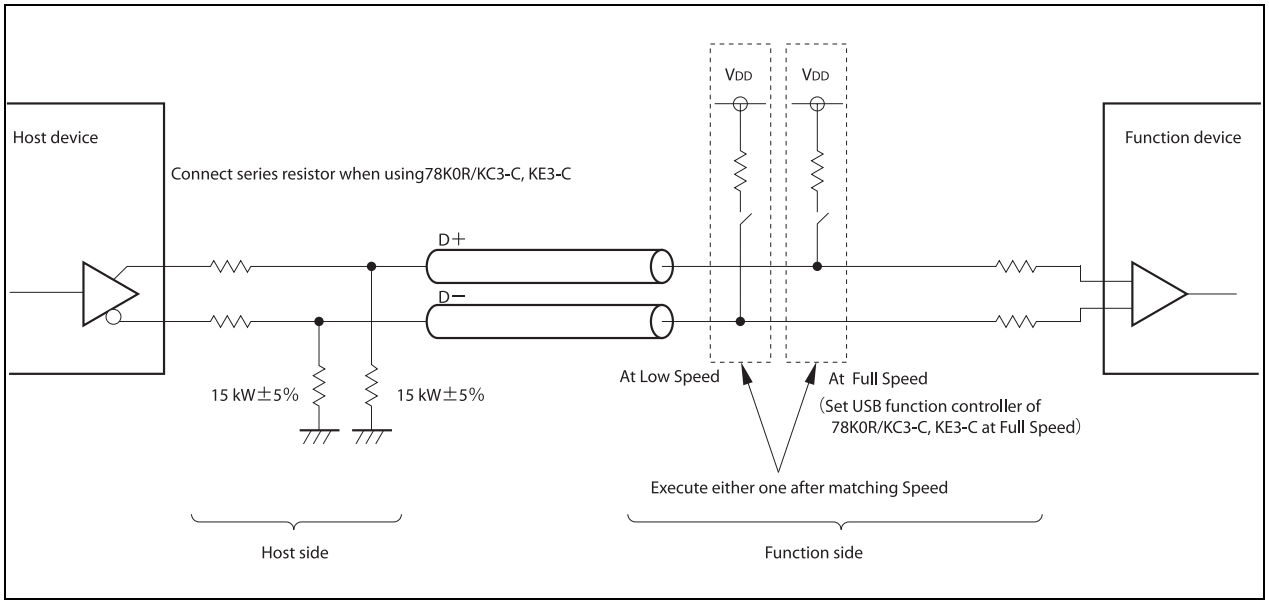
13.3.1 Outline

In USB transmission, when communication is performed with the host controller and function controller facing each other, pull-up/pull-down resistors must be connected to the USB signal (D+/D–) to identify the communication partner. Moreover in the 78K0R/KC3-L, KE3-L, series resistors must also be connected.

Because the 78K0R/KC3-L, KE3-L, do not include these pull-up/pull-down resistors and series resistors, be sure to connect them externally.

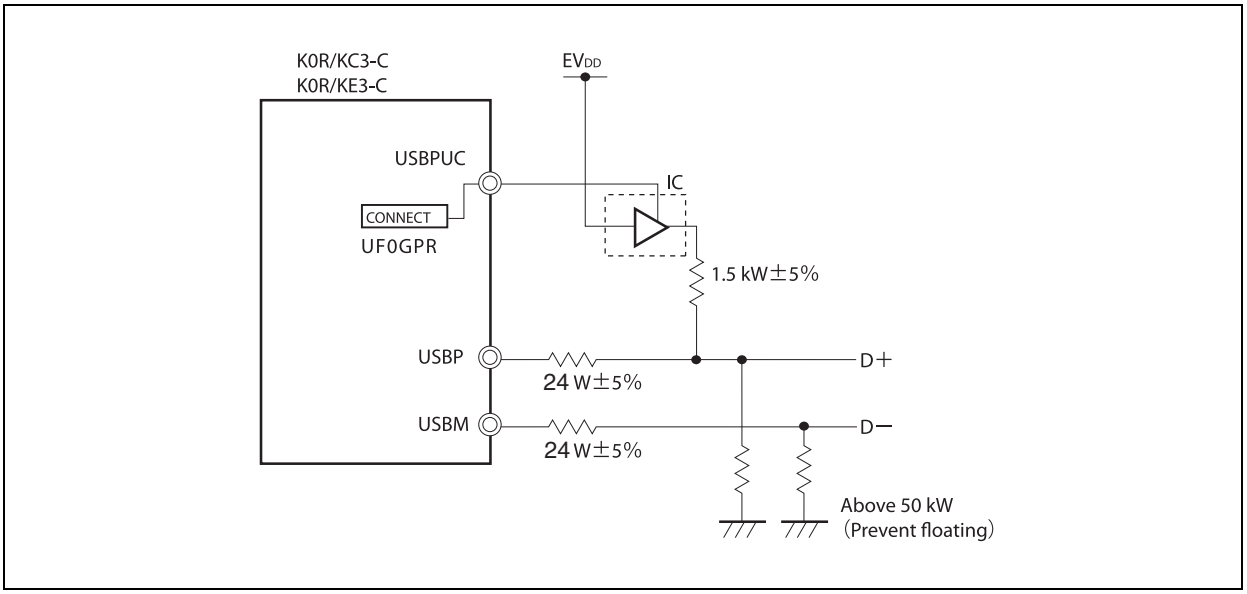
The following shows the outline configuration of the USB transmission line. For details of the external configuration, see the description provided in each section.

Figure 13-2. Outline Configuration of Pull-up, Pull-down, Series Resistors in USB Transmission Line



13.3.2 Connection configuration

<R> **Figure 13-3. Example of USB Function Controller Connection**



<R> **(1) Series resistor connection to D+/D–**

Connect series resistors of 24 Ω ±5% to the D+/D– pins (UFDP, UFDM) of the USB function controller in the 78K0R/KC3-L, KE3-L. If they are not connected, the impedance rating cannot be satisfied and the output waveform may be disturbed.

Allocate the series resistors adjacent to the 78K0R/KC3-L or KE3-L, and make the length of the wiring between the series resistors and the USB connectors the same, to make the impedance of D+ and D– equal (a differential with 90 Ω ±5% is recommended).

<R> **(2) Pull-up control of D+**

Because the function controller of the 78K0R/KC3-L, KE3-L is fixed to full speed (FS), be sure to pull up the D+ pin (UFDP) by 1.5 kΩ ±5% to EV_{DD}.

To disable a connection report (D+ pull up) to the USB host/HUB (such as during high priority servicing or initialization), control the pull-up resistor of D+ via a CONNECT bit of UF0GPR register in the system. In Figure 13-3, if the general-purpose port USBPUC is low level, pulling up of D+ is prohibited.

For the IC2 in Figure 13-3, use an IC to which voltage can be applied when the system power is off.

(3) Detection of USB cable connection/disconnection

When disconnecting the USB cable, it requires VBUS input signal. The voltage from the USB host or HUB (5 V) is applied as the VBUS input signal when the USB cable is connected to the USB host or HUB.

(4) Floating protection during initialization or when USBF is unused

When the USB function controller is initialized or unused, to avoid a floating status, pull the D+/D– pins down using a resistor of 50 kΩ or higher.

13.4 Cautions

(1) Clock accuracy

When using the USB function controller, external clock 12 MHz internal 2 divider × 8 multiplier = 48 MHz must be used as the USB clock, or external clock 16 MHz internal 4 divider × 12 multiplier = 48 MHz, external clock 20 MHz internal 5 divider × 12 multiplier = 48 MHz must be used as the USB clock. When the internal clock is used as the USB clock, use a resonator with an accuracy of 12, 16, 20 MHz ±500 ppm max. If the USB clock accuracy drops, the transmission data cannot satisfy the USB rating.

13.5 Requests

The USB standard has a request command that reports requests from the host device to the function device to execute response processing.

The requests are received in the SETUP stage of control transfer, and most can be automatically processed via the hardware of the USB function controller (USBF).

13.5.1 Automatic requests

(1) Decode

The following tables show the request format and the correspondence between requests and decoded values.

Table 13-2. Request Format

Offset	Field Name	
0	bmRequestType	
1	bRequest	
2	wValue	Lower side
3		Higher side
4	wIndex	Lower side
5		Higher side
6	wLength	Lower side
7		Higher side

Table 13-3. Correspondence Between Requests and Decoded Values

Request \ Offset	Decoded Value								Response			Data Stage
	bmRequestType	bRequest	wValue		wIndex		wLength		Df	Ad	Cf	
	0	1	3	2	5	4	7	6				
GET_INTERFACE	81H	0AH	00H	00H	00H	0nH	00H	01H	STALL	STALL	ACK NAK	√
GET_CONFIGURATION	80H	08H	00H	00H	00H	00H	00H	01H	ACK NAK	ACK NAK	ACK NAK	√
GET_DESCRIPTOR Device	80H	06H	01H	00H	00H	00H	XXH	XXH[Note 1]	ACK NAK	ACK NAK	ACK NAK	√
GET_DESCRIPTOR Configuration	80H	06H	02H	00H	00H	00H	XXH	XXH[Note 1]	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS Device	80H	00H	00H	00H	00H	00H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS Endpoint 0	82H	00H	00H	00H	00H	00H 80H	00H	02H	ACK NAK	ACK NAK	ACK NAK	√
GET_STATUS Endpoint X	82H	00H	00H	00H	00H	$$H	00H	02H	STALL	STALL	ACK NAK	√
CLEAR_FEATURE Device[Note 2]	00H	01H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE Endpoint 0[Note 2]	02H	01H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE Endpoint X[Note 2]	02H	01H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_FEATURE Device[Note 3]	00H	03H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE Endpoint 0[Note 3]	02H	03H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE Endpoint X[Note 3]	02H	03H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_INTERFACE	01H	0BH	00H	0#H	00H	0?H	00H	00H	STALL	STALL	ACK NAK	×
SET_CONFIGURATION[Note 4]	00H	09H	00H	00H 01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_ADDRESS	00H	05H	XXH	XXH	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×

Remark √: Data stage
×: No data stage

Notes 1. If the wLength value is lower than the prepared value, the wLength value is returned; if the wLength value is the prepared value or higher, the prepared value is returned.

2. The CLEAR_FEATURE request clears UF0 device status register L (UF0DSTL) and UF0 EPn status register L (UF0EnSL) (n = 0 - 4, 7, 8) when ACK is received in the status stage.

Notes 3. The SET_FEATURE request sets the UF0 device status register L (UF0DSTL) and UF0 EPn status register L (UF0EnSL) (n = 0 - 4, 7, 8) when ACK is received in the status stage. If the E0HALT bit of the UF0E0SL register is set, a STALL response is made in the status stage or data stage of control transfer for a request other than the GET_STATUS Endpoint0 request, SET_FEATURE Endpoint0 request, and a request generated by the CPUDEC interrupt request, until the CLEAR_FEATURE Endpoint0 request is received. A STALL response to an unsupported request does not set the E0HALT bit of the UF0E0SL register to 1, and the STALL response is cleared as soon as the next SETUP token has been received.

4. If the wValue is not the default value, an automatic STALL response is made.

Cautions 1. The sequence of control transfer defined by the Universal Serial Bus Specification is not satisfied under the following conditions. The operation is not guaranteed under these conditions.

- **If an IN/OUT token is suddenly received without a SETUP stage**
- **If DATA PID1 is sent in the data phase of the SETUP stage**
- **If a token of 128 addresses or more is received**
- **If the request data transmitted in the SETUP stage is of less than 8 bytes**

2. An ACK response is made even when the host transmits data other than a Null packet in the status stage.

3. If the wLength value is 00H during control transfer (read) of FW processing, a Null packet is automatically transmitted for control transfer (without data). The FW request does not automatically transmit a Null packet.

Remarks 1. Df: Default state, Ad: Addressed state, Cf: Configured state

2. n = 0- 4, 7, 8
It is determined by the setting of the UF0 active interface number register (UF0AIFN) whether a request with Interface number 1 - 4,7,8 is correctly responded to, depending on whether the Interface number of the target is valid or not.

3. $$: Valid endpoint number including transfer direction
The valid endpoint is determined by the currently set Alternate Setting number (see 13.6.3 (36) UF0 active alternative setting register (UF0AAS), (38) UF0 endpoint 1 interface mapping register (UF0E1IM) to (42) UF0 endpoint 7 interface mapping register (UF0E7IM)).

4. ? and #: Value transmitted from host (information on Interface numbers 0 to 4)
It is determined by the UF0 active interface number register (UF0AIFN) and UF0 active alternative setting register (UF0AAS) whether an Alternate Setting request corresponding to each Interface number is correctly responded to or not, depending on whether the Interface number and Alternate Setting of the target are valid or not.

(2) Processing

The processing of an automatic request in the Default state, Addressed state, and Configured state is described below.

Remark Default state: State in which an operation is performed with the Default address
Addressed state: State after an address has been allocated
Configured state: State after SET_CONFIGURATION wValue = 1 has been correctly received

(a) CLEAR_FEATURE() request

A STALL response is made in the status stage if the CLEAR_FEATURE() request cannot be cleared, if FEATURE does not exist, or if the target is an interface or an endpoint that does not exist. A STALL response is also made if the wLength value is other than 0.

- Default state: The correct response is made when the CLEAR_FEATURE() request has been received only if the target is a device or a request for Endpoint0; otherwise a STALL response is made in the status stage.
- Addressed state: The correct response is made when the CLEAR_FEATURE() request has been received only if the target is a device or a request for Endpoint0; otherwise a STALL response is made in the status stage.
- Configured state: The correct response is made when the CLEAR_FEATURE() request has been received only if the target is a device or a request for an endpoint that exists; otherwise a STALL response is made in the status stage.

When the CLEAR_FEATURE() request has been correctly processed, the corresponding bit of the UF0 CLR request register (UF0CLR) is set to 1, the EnHALT bit of the UF0 EPn status register L (UF0EnSL) is cleared to 0, and an interrupt is issued (n = 0 - 4, 7,8). If the CLEAR_FEATURE() request is received when the subject is an endpoint, the toggle bit (that controls switching between DATA0 and DATA1) of the corresponding endpoint is always re-set to DATA0.

(b) GET_CONFIGURATION() request

A STALL response is made in the data stage if any of wValue, wIndex, or wLength is other than the values shown in Table 13-3.

- Default state: The value stored in the UF0 configuration register (UF0CNF) is returned when the GET_CONFIGURATION() request has been received.
- Addressed state: The value stored in the UF0CNF register is returned when the GET_CONFIGURATION() request has been received.
- Configured state: The value stored in the UF0CNF register is returned when the GET_CONFIGURATION() request has been received.

(c) GET_DESCRIPTOR() request

If the subject descriptor has a length that is a multiple of wMaxPacketSize, a Null packet is returned to indicate the end of the data stage. If the length of the descriptor at this time is less than the wLength value, the entire descriptor is returned; if the length of the descriptor is greater than the wLength value, the descriptor up to the wLength value is returned.

- Default state: The value stored in UF0 device descriptor register n (UF0DDn) and UF0 configuration/interface/endpoint descriptor register m (UF0CIEm) is returned (n = 0 to 17, m = 0 to 255) when the GET_DESCRIPTOR() request has been received.
- Addressed state: The value stored in the UF0DDn register and UF0CIEm register is returned when the GET_DESCRIPTOR() request has been received.
- Configured state: The value stored in the UF0DDn register and UF0CIEm register is returned when the GET_DESCRIPTOR() request has been received.

A descriptor of up to 256 bytes can be stored in the UF0CIEm register. To return a descriptor of more than 256 bytes, set the CDCGDST bit of the UF0MODC register to 1 and process the GET_DESCRIPTOR() request by FW.

Store the value of the total number of bytes of the descriptor set by the UF0CIEm register - 1 in the UF0 descriptor length register (UF0DSCL). The transfer data is controlled by the value of this data + 1 and wLength.

(d) GET_INTERFACE() request

If either of wValue and wLength is other than that shown in Table 13-3, or if wIndex is other than that set by the UF0 active interface number register (UF0AIFN), a STALL response is made in the data stage.

- Default state: A STALL response is made in the data stage when the GET_INTERFACE() request has been received.
- Addressed state: A STALL response is made in the data stage when the GET_INTERFACE() request has been received.
- Configured state: The value stored in the UF0 interface n register (UF0IFn) corresponding to the wIndex value is returned (n = 0- 4) when the GET_INTERFACE() request has been received.

(e) GET_STATUS() request

A STALL response is made in the data stage if any of wValue, wIndex, or wLength is other than the values shown in Table 13-3. A STALL response is also made in the data stage if the target is an interface or an endpoint that does not exist.

- Default state: The value stored in the target status register[Note] is returned only when the GET_STATUS() request has been received and when the request is for a device or Endpoint0; otherwise a STALL response is made in the data stage.
- Addressed state: The value stored in the target status register[Note] is returned only when the GET_STATUS() request has been received and when the request is for a device or Endpoint0; otherwise a STALL response is made in the data stage.
- Configured state: The value stored in the target status register[Note] is returned only when the GET_STATUS() request has been received and when the request is for a device or an endpoint that exists; otherwise a STALL response is made in the data stage.

Note The target status register is as follows.

- **If the target is a device: UF0 device status register L (UF0DSTL)**
- **If the target is endpoint 0: UF0 EP0 status register L (UF0E0SL)**
- **If the target is endpoint n: UF0 EPn status register L (UF0EnSL) (n = 1 - 4, 7,8)**

(f) SET_ADDRESS() request

A STALL response is made in the status stage if either of wIndex or wLength is other than the values shown in Table 13-3. A STALL response is also made if the specified device address is greater than 127.

- Default state: The device enters the Addressed state and changes the USB Address value to be input to SIE into a specified address value if the specified address is other than 0 when the SET_ADDRESS() request has been received. If the specified address is 0, the device remains in the Default state.
- Addressed state: The device enters the Default state and returns the USB Address value to be input to SIE to the default address if the specified address is 0 when the SET_ADDRESS() request has been received. If the specified address is other than 0, the device remains in the Addressed state, and changes the USB Address value to be input to SIE into a specified new address value.
- Configured state: The device remains in the Configured state and returns the USB Address value to be input to SIE to the default address if the specified address is 0 when the SET_ADDRESS() request has been received. In this case, the endpoints other than endpoint 0 remain valid, and control transfer (IN), control transfer (OUT), bulk transfer and interrupt transfer for an endpoint other than endpoint 0 are also acknowledged. If the specified address is other than 0, the device remains in the Configured state and changes the USB Address value to be input to SIE into a specified new address value.

(g) SET_CONFIGURATION() request

If any of wValue, wIndex, or wLength is other than the values shown in Table 13-3, a STALL response is made in the status stage.

- Default state: The CONF bit of the UF0 mode status register (UF0MODS) and the UF0 configuration register (UF0CNF) are set to 1 if the specified configuration value is 1 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 0, the CONF bit of the UF0MODS register and UF0CNF register are cleared to 0. In other words, the device skips the Addressed state and moves to the Configured state in which it responds to the Default address.
- Addressed state: The CONF bit of the UF0MODS register and UF0CNF register are set to 1 and the device enters the Configured state if the specified configuration value is 1 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 0, the device remains in the Addressed state.
- Configured state: The CONF bit of the UF0MODS register and UF0CNF register are set to 1 and the device returns to the Addressed state if the specified configuration value is 0 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 1, the device remains in the Configured state.

If the SET_CONFIGURATION() request has been correctly processed, the target bit of the UF0 SET request register (UF0SET) is set to 1, and an interrupt is issued. All Halt Features are cleared after the SET_CONFIGURATION() request has been completed even if the specified configuration value is the same as the current configuration value. If the SET_CONFIGURATION() request has been correctly processed, the data toggle of all endpoints is always initialized to DATA0 again (it is defined that the default status, Alternative Setting 0, is set from when the SET_CONFIGURATION request is received to when the SET_INTERFACE request is received).

(h) SET_FEATURE() request

A STALL response is made in the status stage if the SET_FEATURE() request is for a Feature that cannot be set or does not exist, or if the target is an interface or an endpoint that does not exist. A STALL response is also made if the wLength value is other than 0.

- Default state: The correct response is made when the SET_FEATURE() request has been received, only if the request is for a device or Endpoint0; otherwise a STALL response is made in the status stage.
- Addressed state: The correct response is made when the SET_FEATURE() request has been received, only if the request is for a device or Endpoint0; otherwise a STALL response is made in the status stage.
- Configured state: The correct response is made when the SET_FEATURE() request has been received, only if the request is for a device or an endpoint that exists; otherwise a STALL response is made in the status stage.

When the SET_FEATURE() request has been correctly processed, the target bit of the UF0 SET request register (UF0SET) and the EnHALT bit of the UF0 EPn status register L (UF0EnSL) are set to 1, and an interrupt is issued (n = 0 - 4, 7, 8).

(i) SET_INTERFACE() request

If wLength is other than the values shown in Table 13-3, if wIndex is other than the value set to the UF0 active interface number register (UF0AIFN), or if wValue is other than the value set to the UF0 active alternative setting register (UF0AAS), a STALL response is made in the status stage.

- Default state: A STALL response is made in the status stage when the SET_INTERFACE() request has been received.
- Addressed state: A STALL response is made in the status stage when the SET_INTERFACE() request has been received.
- Configured state: Null packet is transmitted in the status stage when the SET_INTERFACE() request has been received.

When the SET_INTERFACE() request has been correctly processed, an interrupt is issued. All the Halt Features of the endpoint linked to the target Interface are cleared after the SET_INTERFACE() request has been cleared. The data toggle of all the endpoints related to the target Interface number is always initialized again to DATA0. When the currently selected Alternative Setting is to be changed by correctly processing the SET_INTERFACE() request, the FIFO of the endpoint that is affected is completely cleared, and all the related interrupt sources are also initialized.

When the SET_INTERFACE() request has been completed, the FIFO of all the endpoints linked to the target Interface are cleared. At the same time, Halt Feature and Data PID are initialized, and the related UF0 INT status n register (UF0ISn) is cleared to 0 (n = 0- 4). (Only Halt Feature and Data PID are cleared when the SET_CONFIGURATION request has been completed.)

13.5.2 Other requests

(1) Response and processing

The following table shows how other requests are responded to and processed.

Table 13-4. Response and Processing of Other Requests

Request	Response and Processing
GET_DESCRIPTOR String	Generation of CPUDEC interrupt request
GET_STATUS Interface	Automatic STALL response
CLEAR_FEATURE Interface	Automatic STALL response
SET_FEATURE Interface	Automatic STALL response
all SET_DESCRIPTOR	Generation of CPUDEC interrupt request
All other requests	Generation of CPUDEC interrupt request

13.6 Register Configuration

13.6.1 USB function control registers

(1) Clock control register

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F059CH	PLL control register	PLLC	R/W	√	√		01H
F059DH	USB clock control register	UCKC	R/W	√	√		00H

(2) EPC control register

(1/2)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F0540H	UF0 EP0NAK register	UF0E0N	R/W		√		00H
F0541H	UF0 EP0NAKALL register	UF0E0NA	R/W		√		00H
F0542H	UF0 EPNAK register	UF0EN	R/W		√		00H
F0543H	UF0 EPNAK mask register	UF0ENM	R/W		√		00H
F0544H	UF0 SNDSIE register	UF0SDS	R/W		√		00H
F0545H	UF0 CLR request register	UF0CLR	R		√		00H
F0546H	UF0 SET request register	UF0SET	R		√		00H
F0547H	UF0 EP status 0 register	UF0EPS0	R		√		00H
F0548H	UF0 EP status 1 register	UF0EPS1	R		√		00H
F0549H	UF0 EP status 2 register	UF0EPS2	R		√		00H
F0550H	UF0 INT status 0 register	UF0IS0	R		√		00H
F0551H	UF0 INT status 1 register	UF0IS1	R		√		00H
F0552H	UF0 INT status 2 register	UF0IS2	R		√		00H
F0553H	UF0 INT status 3 register	UF0IS3	R		√		00H
F0554H	UF0 INT status 4 register	UF0IS4	R		√		00H
F0557H	UF0 INT mask 0 register	UF0IM0	R/W		√		00H
F0558H	UF0 INT mask 1 register	UF0IM1	R/W		√		00H
F0559H	UF0 INT mask 2 register	UF0IM2	R/W		√		00H
F055AH	UF0 INT mask 3 register	UF0IM3	R/W		√		00H
F055BH	UF0 INT mask 4 register	UF0IM4	R/W		√		00H
F055EH	UF0 INT clear 0 register	UF0IC0	W		√		00H
F055FH	UF0 INT clear 1 register	UF0IC1	W		√		00H
F0560H	UF0 INT clear 2 register	UF0IC2	W		√		00H
F0561H	UF0 INT clear 3 register	UF0IC3	W		√		00H
F0562H	UF0 INT clear 4 register	UF0IC4	W		√		00H
F0570H	UF0 FIFO clear 0 register	UF0FIC0	W		√		00H
F0571H	UF0 FIFO clear 1 register	UF0FIC1	W		√		00H
F0575H	UF0 data end register	UF0DEND	R/W		√		00H
F0577H	UF0 GPR register	UF0GPR	R/W		√		00H
F057AH	UF0 mode control register	UF0MODC	R/W		√		00H
F057CH	UF0 mode status register	UF0MODS	R		√		00H

(2/2)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F06F0H	UF0 active interface number register	UF0AIFN	R/W		√		00H
F06F1H	UF0 active alternative setting register	UF0AAS	R/W		√		00H
F06F2H	UF0 alternative setting status register	UF0ASS	R		√		00H
F06F3H	UF0 endpoint 1 interface mapping register	UF0E1IM	R/W		√		00H
F06F4H	UF0 endpoint 2 interface mapping register	UF0E2IM	R/W		√		00H
F06F5H	UF0 endpoint 3 interface mapping register	UF0E3IM	R/W		√		00H
F06F6H	UF0 endpoint 4 interface mapping register	UF0E4IM	R/W		√		00H
F06F9H	UF0 endpoint 7 interface mapping register	UF0E7IM	R/W		√		00H
F06FAH	UF0 endpoint 8 interface mapping register	UF0E8IM	R/W		√		00H

(2) EPC data hold register

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F0580H	UF0 EP0 read register	UF0E0R	R		√		Undefined
F0581H	UF0 EP0 length register	UF0E0L	R		√		00H
F0582H	UF0 EP0 setup register	UF0E0ST	R		√		00H
F0583H	UF0 EP0 write register	UF0E0W	W		√		00H
F0584H	UF0 bulk-out 1 register	UF0BO1	R		√		Undefined
F0585H	UF0 bulk-out 1 length register	UF0BO1L	R		√		00H
F0586H	UF0 bulk-out 2 register	UF0BO2	R		√		Undefined
F0587H	UF0 bulk-out 2 length register	UF0BO2L	R		√		00H
F0588H	UF0 bulk-in 1 register	UF0BI1	W		√		00H
F0589H	UF0 bulk-in 2 register	UF0BI2	W		√		00H
F058AH	UF0 interrupt 1 register	UF0INT1	W		√		00H
F058BH	UF0 interrupt 2 register	UF0INT2	W		√		00H

(3) EPC request data register

(1/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F05A2H	UF0 device status register L	UF0DSTL	R/W		√		00H
F05A6H	UF0 EP0 status register L	UF0E0SL	R/W		√		00H
F05A8H	UF0 EP1 status register L	UF0E1SL	R/W		√		00H
F05AAH	UF0 EP2 status register L	UF0E2SL	R/W		√		00H
F05ACH	UF0 EP3 status register L	UF0E3SL	R/W		√		00H
F05AEH	UF0 EP4 status register L	UF0E4SL	R/W		√		00H
F05B4H	UF0 EP7 status register L	UF0E7SL	R/W		√		00H
F05B6H	UF0 EP8 status register L	UF0E7SL	R/W		√		00H
F05C0H	UF0 address register	UF0ADRS	R		√		00H
F05C1H	UF0 configuration register	UF0CNF	R		√		00H
F05C2H	UF0 interface 0 register	UF0IF0	R		√		00H
F05C3H	UF0 interface 1 register	UF0IF1	R		√		00H
F05C4H	UF0 interface 2 register	UF0IF2	R		√		00H
F05C5H	UF0 interface 3 register	UF0IF3	R		√		00H
F05C6H	UF0 interface 4 register	UF0IF4	R		√		00H
F05D0H	UF0 descriptor length register	UF0DSCL	R/W		√		00H
F05D1H	UF0 device descriptor register 0	UF0DD0	R/W		√		Undefined
F05D2H	UF0 device descriptor register 1	UF0DD1	R/W		√		Undefined
F05D3H	UF0 device descriptor register 2	UF0DD2	R/W		√		Undefined
F05D4H	UF0 device descriptor register 3	UF0DD3	R/W		√		Undefined
F05D5H	UF0 device descriptor register 4	UF0DD4	R/W		√		Undefined
F05D6H	UF0 device descriptor register 5	UF0DD5	R/W		√		Undefined
F05D7H	UF0 device descriptor register 6	UF0DD6	R/W		√		Undefined
F05D8H	UF0 device descriptor register 7	UF0DD7	R/W		√		Undefined
F05D9H	UF0 device descriptor register 8	UF0DD8	R/W		√		Undefined
F05DAH	UF0 device descriptor register 9	UF0DD9	R/W		√		Undefined
F05DBH	UF0 device descriptor register 10	UF0DD10	R/W		√		Undefined
F05DCH	UF0 device descriptor register 11	UF0DD11	R/W		√		Undefined
F05DDH	UF0 device descriptor register 12	UF0DD12	R/W		√		Undefined
F05DEH	UF0 device descriptor register 13	UF0DD13	R/W		√		Undefined
F05DFH	UF0 device descriptor register 14	UF0DD14	R/W		√		Undefined
F05E0H	UF0 device descriptor register 15	UF0DD15	R/W		√		Undefined
F05E1H	UF0 device descriptor register 16	UF0DD16	R/W		√		Undefined
F05E2H	UF0 device descriptor register 17	UF0DD17	R/W		√		Undefined
F05E3H	UF0 configuration/interface/endpoint descriptor register 0	UF0CIE0	R/W		√		Undefined
F05E4H	UF0 configuration/interface/endpoint descriptor register 1	UF0CIE1	R/W		√		Undefined
F05E5H	UF0 configuration/interface/endpoint descriptor register 2	UF0CIE2	R/W		√		Undefined
F05E6H	UF0 configuration/interface/endpoint descriptor register 3	UF0CIE3	R/W		√		Undefined

(2/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F05E7H	UF0 configuration/interface/endpoint descriptor register 4	UF0CIE4	R/W		√		Undefined
F05E8H	UF0 configuration/interface/endpoint descriptor register 5	UF0CIE5	R/W		√		Undefined
F05E9H	UF0 configuration/interface/endpoint descriptor register 6	UF0CIE6	R/W		√		Undefined
F05EAH	UF0 configuration/interface/endpoint descriptor register 7	UF0CIE7	R/W		√		Undefined
F05EBH	UF0 configuration/interface/endpoint descriptor register 8	UF0CIE8	R/W		√		Undefined
F05ECH	UF0 configuration/interface/endpoint descriptor register 9	UF0CIE9	R/W		√		Undefined
F05EDH	UF0 configuration/interface/endpoint descriptor register 10	UF0CIE10	R/W		√		Undefined
F05EEH	UF0 configuration/interface/endpoint descriptor register 11	UF0CIE11	R/W		√		Undefined
F05EFH	UF0 configuration/interface/endpoint descriptor register 12	UF0CIE12	R/W		√		Undefined
F05F0H	UF0 configuration/interface/endpoint descriptor register 13	UF0CIE13	R/W		√		Undefined
F05F1H	UF0 configuration/interface/endpoint descriptor register 14	UF0CIE14	R/W		√		Undefined
F05F2H	UF0 configuration/interface/endpoint descriptor register 15	UF0CIE15	R/W		√		Undefined
F05F3H	UF0 configuration/interface/endpoint descriptor register 16	UF0CIE16	R/W		√		Undefined
F05F4H	UF0 configuration/interface/endpoint descriptor register 17	UF0CIE17	R/W		√		Undefined
F05F5H	UF0 configuration/interface/endpoint descriptor register 18	UF0CIE18	R/W		√		Undefined
F05F6H	UF0 configuration/interface/endpoint descriptor register 19	UF0CIE19	R/W		√		Undefined
F05F7H	UF0 configuration/interface/endpoint descriptor register 20	UF0CIE20	R/W		√		Undefined
F05F8H	UF0 configuration/interface/endpoint descriptor register 21	UF0CIE21	R/W		√		Undefined
F05F9H	UF0 configuration/interface/endpoint descriptor register 22	UF0CIE22	R/W		√		Undefined
F05FAH	UF0 configuration/interface/endpoint descriptor register 23	UF0CIE23	R/W		√		Undefined
F05FBH	UF0 configuration/interface/endpoint descriptor register 24	UF0CIE24	R/W		√		Undefined
F05FCH	UF0 configuration/interface/endpoint descriptor register 25	UF0CIE25	R/W		√		Undefined

(3/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F05FDH	UF0 configuration/interface/endpoint descriptor register 26	UF0CIE26	R/W		√		Undefined
F05FEH	UF0 configuration/interface/endpoint descriptor register 27	UF0CIE27	R/W		√		Undefined
F05FFH	UF0 configuration/interface/endpoint descriptor register 28	UF0CIE28	R/W		√		Undefined
F0600H	UF0 configuration/interface/endpoint descriptor register 29	UF0CIE29	R/W		√		Undefined
F0601H	UF0 configuration/interface/endpoint descriptor register 30	UF0CIE30	R/W		√		Undefined
F0602H	UF0 configuration/interface/endpoint descriptor register 31	UF0CIE31	R/W		√		Undefined
F0603H	UF0 configuration/interface/endpoint descriptor register 32	UF0CIE32	R/W		√		Undefined
F0604H	UF0 configuration/interface/endpoint descriptor register 33	UF0CIE33	R/W		√		Undefined
F0605H	UF0 configuration/interface/endpoint descriptor register 34	UF0CIE34	R/W		√		Undefined
F0606H	UF0 configuration/interface/endpoint descriptor register 35	UF0CIE35	R/W		√		Undefined
F0607H	UF0 configuration/interface/endpoint descriptor register 36	UF0CIE36	R/W		√		Undefined
F0608H	UF0 configuration/interface/endpoint descriptor register 37	UF0CIE37	R/W		√		Undefined
F0609H	UF0 configuration/interface/endpoint descriptor register 38	UF0CIE38	R/W		√		Undefined
F060AH	UF0 configuration/interface/endpoint descriptor register 39	UF0CIE39	R/W		√		Undefined
F060BH	UF0 configuration/interface/endpoint descriptor register 40	UF0CIE40	R/W		√		Undefined
F060CH	UF0 configuration/interface/endpoint descriptor register 41	UF0CIE41	R/W		√		Undefined
F060DH	UF0 configuration/interface/endpoint descriptor register 42	UF0CIE42	R/W		√		Undefined
F060EH	UF0 configuration/interface/endpoint descriptor register 43	UF0CIE43	R/W		√		Undefined
F060FH	UF0 configuration/interface/endpoint descriptor register 44	UF0CIE44	R/W		√		Undefined
F0610H	UF0 configuration/interface/endpoint descriptor register 45	UF0CIE45	R/W		√		Undefined
F0611H	UF0 configuration/interface/endpoint descriptor register 46	UF0CIE46	R/W		√		Undefined
F0612H	UF0 configuration/interface/endpoint descriptor register 47	UF0CIE47	R/W		√		Undefined

(4/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F0613H	UF0 configuration/interface/endpoint descriptor register 48	UF0CIE48	R/W		√		Undefined
F0614H	UF0 configuration/interface/endpoint descriptor register 49	UF0CIE49	R/W		√		Undefined
F0615H	UF0 configuration/interface/endpoint descriptor register 50	UF0CIE50	R/W		√		Undefined
F0616H	UF0 configuration/interface/endpoint descriptor register 51	UF0CIE51	R/W		√		Undefined
F0617H	UF0 configuration/interface/endpoint descriptor register 52	UF0CIE52	R/W		√		Undefined
F0618H	UF0 configuration/interface/endpoint descriptor register 53	UF0CIE53	R/W		√		Undefined
F0619H	UF0 configuration/interface/endpoint descriptor register 54	UF0CIE54	R/W		√		Undefined
F061AH	UF0 configuration/interface/endpoint descriptor register 55	UF0CIE55	R/W		√		Undefined
F061BH	UF0 configuration/interface/endpoint descriptor register 56	UF0CIE56	R/W		√		Undefined
F061CH	UF0 configuration/interface/endpoint descriptor register 57	UF0CIE57	R/W		√		Undefined
F061DH	UF0 configuration/interface/endpoint descriptor register 58	UF0CIE58	R/W		√		Undefined
F061EH	UF0 configuration/interface/endpoint descriptor register 59	UF0CIE59	R/W		√		Undefined
F061FH	UF0 configuration/interface/endpoint descriptor register 60	UF0CIE60	R/W		√		Undefined
F0620H	UF0 configuration/interface/endpoint descriptor register 61	UF0CIE61	R/W		√		Undefined
F0621H	UF0 configuration/interface/endpoint descriptor register 62	UF0CIE62	R/W		√		Undefined
F0622H	UF0 configuration/interface/endpoint descriptor register 63	UF0CIE63	R/W		√		Undefined
F0623H	UF0 configuration/interface/endpoint descriptor register 64	UF0CIE64	R/W		√		Undefined
F0624H	UF0 configuration/interface/endpoint descriptor register 65	UF0CIE65	R/W		√		Undefined
F0625H	UF0 configuration/interface/endpoint descriptor register 66	UF0CIE66	R/W		√		Undefined
F0626H	UF0 configuration/interface/endpoint descriptor register 67	UF0CIE67	R/W		√		Undefined
F0627H	UF0 configuration/interface/endpoint descriptor register 68	UF0CIE68	R/W		√		Undefined
F0628H	UF0 configuration/interface/endpoint descriptor register 69	UF0CIE69	R/W		√		Undefined

(5/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F0629H	UF0 configuration/interface/endpoint descriptor register 70	UF0CIE70	R/W		√		Undefined
F062AH	UF0 configuration/interface/endpoint descriptor register 71	UF0CIE71	R/W		√		Undefined
F062BH	UF0 configuration/interface/endpoint descriptor register 72	UF0CIE72	R/W		√		Undefined
F062CH	UF0 configuration/interface/endpoint descriptor register 73	UF0CIE73	R/W		√		Undefined
F062DH	UF0 configuration/interface/endpoint descriptor register 74	UF0CIE74	R/W		√		Undefined
F062EH	UF0 configuration/interface/endpoint descriptor register 75	UF0CIE75	R/W		√		Undefined
F062FH	UF0 configuration/interface/endpoint descriptor register 76	UF0CIE76	R/W		√		Undefined
F0630H	UF0 configuration/interface/endpoint descriptor register 77	UF0CIE77	R/W		√		Undefined
F0631H	UF0 configuration/interface/endpoint descriptor register 78	UF0CIE78	R/W		√		Undefined
F0632H	UF0 configuration/interface/endpoint descriptor register 79	UF0CIE79	R/W		√		Undefined
F0633H	UF0 configuration/interface/endpoint descriptor register 80	UF0CIE80	R/W		√		Undefined
F0634H	UF0 configuration/interface/endpoint descriptor register 81	UF0CIE81	R/W		√		Undefined
F0635H	UF0 configuration/interface/endpoint descriptor register 82	UF0CIE82	R/W		√		Undefined
F0636H	UF0 configuration/interface/endpoint descriptor register 83	UF0CIE83	R/W		√		Undefined
F0637H	UF0 configuration/interface/endpoint descriptor register 84	UF0CIE84	R/W		√		Undefined
F0638H	UF0 configuration/interface/endpoint descriptor register 85	UF0CIE85	R/W		√		Undefined
F0639H	UF0 configuration/interface/endpoint descriptor register 86	UF0CIE86	R/W		√		Undefined
F063AH	UF0 configuration/interface/endpoint descriptor register 87	UF0CIE87	R/W		√		Undefined
F063BH	UF0 configuration/interface/endpoint descriptor register 88	UF0CIE88	R/W		√		Undefined
F063CH	UF0 configuration/interface/endpoint descriptor register 89	UF0CIE89	R/W		√		Undefined
F063DH	UF0 configuration/interface/endpoint descriptor register 90	UF0CIE90	R/W		√		Undefined
F063EH	UF0 configuration/interface/endpoint descriptor register 91	UF0CIE91	R/W		√		Undefined

(6/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F063FH	UF0 configuration/interface/endpoint descriptor register 92	UF0CIE92	R/W		√		Undefined
F0640H	UF0 configuration/interface/endpoint descriptor register 93	UF0CIE93	R/W		√		Undefined
F0641H	UF0 configuration/interface/endpoint descriptor register 94	UF0CIE94	R/W		√		Undefined
F0642H	UF0 configuration/interface/endpoint descriptor register 95	UF0CIE95	R/W		√		Undefined
F0643H	UF0 configuration/interface/endpoint descriptor register 96	UF0CIE96	R/W		√		Undefined
F0644H	UF0 configuration/interface/endpoint descriptor register 97	UF0CIE97	R/W		√		Undefined
F0645H	UF0 configuration/interface/endpoint descriptor register 98	UF0CIE98	R/W		√		Undefined
F0646H	UF0 configuration/interface/endpoint descriptor register 99	UF0CIE99	R/W		√		Undefined
F0647H	UF0 configuration/interface/endpoint descriptor register 100	UF0CIE100	R/W		√		Undefined
F0648H	UF0 configuration/interface/endpoint descriptor register 101	UF0CIE101	R/W		√		Undefined
F0649H	UF0 configuration/interface/endpoint descriptor register 102	UF0CIE102	R/W		√		Undefined
F064AH	UF0 configuration/interface/endpoint descriptor register 103	UF0CIE103	R/W		√		Undefined
F064BH	UF0 configuration/interface/endpoint descriptor register 104	UF0CIE104	R/W		√		Undefined
F064CH	UF0 configuration/interface/endpoint descriptor register 105	UF0CIE105	R/W		√		Undefined
F064DH	UF0 configuration/interface/endpoint descriptor register 106	UF0CIE106	R/W		√		Undefined
F064EH	UF0 configuration/interface/endpoint descriptor register 107	UF0CIE107	R/W		√		Undefined
F064FH	UF0 configuration/interface/endpoint descriptor register 108	UF0CIE108	R/W		√		Undefined
F0650H	UF0 configuration/interface/endpoint descriptor register 109	UF0CIE109	R/W		√		Undefined
F0651H	UF0 configuration/interface/endpoint descriptor register 110	UF0CIE110	R/W		√		Undefined
F0652H	UF0 configuration/interface/endpoint descriptor register 111	UF0CIE111	R/W		√		Undefined
F0653H	UF0 configuration/interface/endpoint descriptor register 112	UF0CIE112	R/W		√		Undefined
F0654H	UF0 configuration/interface/endpoint descriptor register 113	UF0CIE113	R/W		√		Undefined

(7/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F0655H	UF0 configuration/interface/endpoint descriptor register 114	UF0CIE114	R/W		√		Undefined
F0656H	UF0 configuration/interface/endpoint descriptor register 115	UF0CIE115	R/W		√		Undefined
F0657H	UF0 configuration/interface/endpoint descriptor register 116	UF0CIE116	R/W		√		Undefined
F0658H	UF0 configuration/interface/endpoint descriptor register 117	UF0CIE117	R/W		√		Undefined
F0659H	UF0 configuration/interface/endpoint descriptor register 118	UF0CIE118	R/W		√		Undefined
F065AH	UF0 configuration/interface/endpoint descriptor register 119	UF0CIE119	R/W		√		Undefined
F065BH	UF0 configuration/interface/endpoint descriptor register 120	UF0CIE120	R/W		√		Undefined
F065CH	UF0 configuration/interface/endpoint descriptor register 121	UF0CIE121	R/W		√		Undefined
F065DH	UF0 configuration/interface/endpoint descriptor register 122	UF0CIE122	R/W		√		Undefined
F065EH	UF0 configuration/interface/endpoint descriptor register 123	UF0CIE123	R/W		√		Undefined
F065FH	UF0 configuration/interface/endpoint descriptor register 124	UF0CIE124	R/W		√		Undefined
F0660H	UF0 configuration/interface/endpoint descriptor register 125	UF0CIE125	R/W		√		Undefined
F0661H	UF0 configuration/interface/endpoint descriptor register 126	UF0CIE126	R/W		√		Undefined
F0662H	UF0 configuration/interface/endpoint descriptor register 127	UF0CIE127	R/W		√		Undefined
F0663H	UF0 configuration/interface/endpoint descriptor register 128	UF0CIE128	R/W		√		Undefined
F0664H	UF0 configuration/interface/endpoint descriptor register 129	UF0CIE129	R/W		√		Undefined
F0665H	UF0 configuration/interface/endpoint descriptor register 130	UF0CIE130	R/W		√		Undefined
F0666H	UF0 configuration/interface/endpoint descriptor register 131	UF0CIE131	R/W		√		Undefined
F0667H	UF0 configuration/interface/endpoint descriptor register 132	UF0CIE132	R/W		√		Undefined
F0668H	UF0 configuration/interface/endpoint descriptor register 133	UF0CIE133	R/W		√		Undefined
F0669H	UF0 configuration/interface/endpoint descriptor register 134	UF0CIE134	R/W		√		Undefined
F066AH	UF0 configuration/interface/endpoint descriptor register 135	UF0CIE135	R/W		√		Undefined

(8/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F066BH	UF0 configuration/interface/endpoint descriptor register 136	UF0CIE136	R/W		√		Undefined
F066CH	UF0 configuration/interface/endpoint descriptor register 137	UF0CIE137	R/W		√		Undefined
F066DH	UF0 configuration/interface/endpoint descriptor register 138	UF0CIE138	R/W		√		Undefined
F066EH	UF0 configuration/interface/endpoint descriptor register 139	UF0CIE139	R/W		√		Undefined
F066FH	UF0 configuration/interface/endpoint descriptor register 140	UF0CIE140	R/W		√		Undefined
F0670H	UF0 configuration/interface/endpoint descriptor register 141	UF0CIE141	R/W		√		Undefined
F0671H	UF0 configuration/interface/endpoint descriptor register 142	UF0CIE142	R/W		√		Undefined
F0672H	UF0 configuration/interface/endpoint descriptor register 143	UF0CIE143	R/W		√		Undefined
F0673H	UF0 configuration/interface/endpoint descriptor register 144	UF0CIE144	R/W		√		Undefined
F0674H	UF0 configuration/interface/endpoint descriptor register 145	UF0CIE145	R/W		√		Undefined
F0675H	UF0 configuration/interface/endpoint descriptor register 146	UF0CIE146	R/W		√		Undefined
F0676H	UF0 configuration/interface/endpoint descriptor register 147	UF0CIE147	R/W		√		Undefined
F0677H	UF0 configuration/interface/endpoint descriptor register 148	UF0CIE148	R/W		√		Undefined
F0678H	UF0 configuration/interface/endpoint descriptor register 149	UF0CIE149	R/W		√		Undefined
F0679H	UF0 configuration/interface/endpoint descriptor register 150	UF0CIE150	R/W		√		Undefined
F067AH	UF0 configuration/interface/endpoint descriptor register 151	UF0CIE151	R/W		√		Undefined
F067BH	UF0 configuration/interface/endpoint descriptor register 152	UF0CIE152	R/W		√		Undefined
F067CH	UF0 configuration/interface/endpoint descriptor register 153	UF0CIE153	R/W		√		Undefined
F067DH	UF0 configuration/interface/endpoint descriptor register 154	UF0CIE154	R/W		√		Undefined
F067EH	UF0 configuration/interface/endpoint descriptor register 155	UF0CIE155	R/W		√		Undefined
F067FH	UF0 configuration/interface/endpoint descriptor register 156	UF0CIE156	R/W		√		Undefined
F0680H	UF0 configuration/interface/endpoint descriptor register 157	UF0CIE157	R/W		√		Undefined

(9/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F0681H	UF0 configuration/interface/endpoint descriptor register 158	UF0CIE158	R/W		√		Undefined
F0682H	UF0 configuration/interface/endpoint descriptor register 159	UF0CIE159	R/W		√		Undefined
F0683H	UF0 configuration/interface/endpoint descriptor register 160	UF0CIE160	R/W		√		Undefined
F0684H	UF0 configuration/interface/endpoint descriptor register 161	UF0CIE161	R/W		√		Undefined
F0685H	UF0 configuration/interface/endpoint descriptor register 162	UF0CIE162	R/W		√		Undefined
F0686H	UF0 configuration/interface/endpoint descriptor register 163	UF0CIE163	R/W		√		Undefined
F0687H	UF0 configuration/interface/endpoint descriptor register 164	UF0CIE164	R/W		√		Undefined
F0688H	UF0 configuration/interface/endpoint descriptor register 165	UF0CIE165	R/W		√		Undefined
F0689H	UF0 configuration/interface/endpoint descriptor register 166	UF0CIE166	R/W		√		Undefined
F068AH	UF0 configuration/interface/endpoint descriptor register 167	UF0CIE167	R/W		√		Undefined
F068BH	UF0 configuration/interface/endpoint descriptor register 168	UF0CIE168	R/W		√		Undefined
F068CH	UF0 configuration/interface/endpoint descriptor register 169	UF0CIE169	R/W		√		Undefined
F068DH	UF0 configuration/interface/endpoint descriptor register 170	UF0CIE170	R/W		√		Undefined
F068EH	UF0 configuration/interface/endpoint descriptor register 171	UF0CIE171	R/W		√		Undefined
F068FH	UF0 configuration/interface/endpoint descriptor register 172	UF0CIE172	R/W		√		Undefined
F0690H	UF0 configuration/interface/endpoint descriptor register 173	UF0CIE173	R/W		√		Undefined
F0691H	UF0 configuration/interface/endpoint descriptor register 174	UF0CIE174	R/W		√		Undefined
F0692H	UF0 configuration/interface/endpoint descriptor register 175	UF0CIE175	R/W		√		Undefined
F0693H	UF0 configuration/interface/endpoint descriptor register 176	UF0CIE176	R/W		√		Undefined
F0694H	UF0 configuration/interface/endpoint descriptor register 177	UF0CIE177	R/W		√		Undefined
F0695H	UF0 configuration/interface/endpoint descriptor register 178	UF0CIE178	R/W		√		Undefined
F0696H	UF0 configuration/interface/endpoint descriptor register 179	UF0CIE179	R/W		√		Undefined

(10/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F0697H	UF0 configuration/interface/endpoint descriptor register 180	UF0CIE180	R/W		√		Undefined
F0698H	UF0 configuration/interface/endpoint descriptor register 181	UF0CIE181	R/W		√		Undefined
F0699H	UF0 configuration/interface/endpoint descriptor register 182	UF0CIE182	R/W		√		Undefined
F069AH	UF0 configuration/interface/endpoint descriptor register 183	UF0CIE183	R/W		√		Undefined
F069BH	UF0 configuration/interface/endpoint descriptor register 184	UF0CIE184	R/W		√		Undefined
F069CH	UF0 configuration/interface/endpoint descriptor register 185	UF0CIE185	R/W		√		Undefined
F069DH	UF0 configuration/interface/endpoint descriptor register 186	UF0CIE186	R/W		√		Undefined
F069EH	UF0 configuration/interface/endpoint descriptor register 187	UF0CIE187	R/W		√		Undefined
F069FH	UF0 configuration/interface/endpoint descriptor register 188	UF0CIE188	R/W		√		Undefined
F06A0H	UF0 configuration/interface/endpoint descriptor register 189	UF0CIE189	R/W		√		Undefined
F06A1H	UF0 configuration/interface/endpoint descriptor register 190	UF0CIE190	R/W		√		Undefined
F06A2H	UF0 configuration/interface/endpoint descriptor register 191	UF0CIE191	R/W		√		Undefined
F06A3H	UF0 configuration/interface/endpoint descriptor register 192	UF0CIE192	R/W		√		Undefined
F06A4H	UF0 configuration/interface/endpoint descriptor register 193	UF0CIE193	R/W		√		Undefined
F06A5H	UF0 configuration/interface/endpoint descriptor register 194	UF0CIE194	R/W		√		Undefined
F06A6H	UF0 configuration/interface/endpoint descriptor register 195	UF0CIE195	R/W		√		Undefined
F06A7H	UF0 configuration/interface/endpoint descriptor register 196	UF0CIE196	R/W		√		Undefined
F06A8H	UF0 configuration/interface/endpoint descriptor register 197	UF0CIE197	R/W		√		Undefined
F06A9H	UF0 configuration/interface/endpoint descriptor register 198	UF0CIE198	R/W		√		Undefined
F06AAH	UF0 configuration/interface/endpoint descriptor register 199	UF0CIE199	R/W		√		Undefined
F06ABH	UF0 configuration/interface/endpoint descriptor register 200	UF0CIE200	R/W		√		Undefined
F06ACH	UF0 configuration/interface/endpoint descriptor register 201	UF0CIE201	R/W		√		Undefined

(11/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F06ADH	UF0 configuration/interface/endpoint descriptor register 202	UF0CIE202	R/W		√		Undefined
F06AEH	UF0 configuration/interface/endpoint descriptor register 203	UF0CIE203	R/W		√		Undefined
F06AFH	UF0 configuration/interface/endpoint descriptor register 204	UF0CIE204	R/W		√		Undefined
F06B0H	UF0 configuration/interface/endpoint descriptor register 205	UF0CIE205	R/W		√		Undefined
F06B1H	UF0 configuration/interface/endpoint descriptor register 206	UF0CIE206	R/W		√		Undefined
F06B2H	UF0 configuration/interface/endpoint descriptor register 207	UF0CIE207	R/W		√		Undefined
F06B3H	UF0 configuration/interface/endpoint descriptor register 208	UF0CIE208	R/W		√		Undefined
F06B4H	UF0 configuration/interface/endpoint descriptor register 209	UF0CIE209	R/W		√		Undefined
F06B5H	UF0 configuration/interface/endpoint descriptor register 210	UF0CIE210	R/W		√		Undefined
F06B6H	UF0 configuration/interface/endpoint descriptor register 211	UF0CIE211	R/W		√		Undefined
F06B7H	UF0 configuration/interface/endpoint descriptor register 212	UF0CIE212	R/W		√		Undefined
F06B8H	UF0 configuration/interface/endpoint descriptor register 213	UF0CIE213	R/W		√		Undefined
F06B9H	UF0 configuration/interface/endpoint descriptor register 214	UF0CIE214	R/W		√		Undefined
F06BAH	UF0 configuration/interface/endpoint descriptor register 215	UF0CIE215	R/W		√		Undefined
F06BBH	UF0 configuration/interface/endpoint descriptor register 216	UF0CIE216	R/W		√		Undefined
F06BCH	UF0 configuration/interface/endpoint descriptor register 217	UF0CIE217	R/W		√		Undefined
F06BDH	UF0 configuration/interface/endpoint descriptor register 218	UF0CIE218	R/W		√		Undefined
F06BEH	UF0 configuration/interface/endpoint descriptor register 219	UF0CIE219	R/W		√		Undefined
F06BFH	UF0 configuration/interface/endpoint descriptor register 220	UF0CIE220	R/W		√		Undefined
F06C0H	UF0 configuration/interface/endpoint descriptor register 221	UF0CIE221	R/W		√		Undefined
F06C1H	UF0 configuration/interface/endpoint descriptor register 222	UF0CIE222	R/W		√		Undefined
F06C2H	UF0 configuration/interface/endpoint descriptor register 223	UF0CIE223	R/W		√		Undefined

(12/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F06C3H	UF0 configuration/interface/endpoint descriptor register 224	UF0CIE224	R/W		√		Undefined
F06C4H	UF0 configuration/interface/endpoint descriptor register 225	UF0CIE225	R/W		√		Undefined
F06C5H	UF0 configuration/interface/endpoint descriptor register 226	UF0CIE226	R/W		√		Undefined
F06C6H	UF0 configuration/interface/endpoint descriptor register 227	UF0CIE227	R/W		√		Undefined
F06C7H	UF0 configuration/interface/endpoint descriptor register 228	UF0CIE228	R/W		√		Undefined
F06C8H	UF0 configuration/interface/endpoint descriptor register 229	UF0CIE229	R/W		√		Undefined
F06C9H	UF0 configuration/interface/endpoint descriptor register 230	UF0CIE230	R/W		√		Undefined
F06CAH	UF0 configuration/interface/endpoint descriptor register 231	UF0CIE231	R/W		√		Undefined
F06CBH	UF0 configuration/interface/endpoint descriptor register 232	UF0CIE232	R/W		√		Undefined
F06CCH	UF0 configuration/interface/endpoint descriptor register 233	UF0CIE233	R/W		√		Undefined
F06CDH	UF0 configuration/interface/endpoint descriptor register 234	UF0CIE234	R/W		√		Undefined
F06CEH	UF0 configuration/interface/endpoint descriptor register 235	UF0CIE235	R/W		√		Undefined
F06CFH	UF0 configuration/interface/endpoint descriptor register 236	UF0CIE236	R/W		√		Undefined
F06D0H	UF0 configuration/interface/endpoint descriptor register 237	UF0CIE237	R/W		√		Undefined
F06D1H	UF0 configuration/interface/endpoint descriptor register 238	UF0CIE238	R/W		√		Undefined
F06D2H	UF0 configuration/interface/endpoint descriptor register 239	UF0CIE239	R/W		√		Undefined
F06D3H	UF0 configuration/interface/endpoint descriptor register 240	UF0CIE240	R/W		√		Undefined
F06D4H	UF0 configuration/interface/endpoint descriptor register 241	UF0CIE241	R/W		√		Undefined
F06D5H	UF0 configuration/interface/endpoint descriptor register 242	UF0CIE242	R/W		√		Undefined
F06D6H	UF0 configuration/interface/endpoint descriptor register 243	UF0CIE243	R/W		√		Undefined
F06D7H	UF0 configuration/interface/endpoint descriptor register 244	UF0CIE244	R/W		√		Undefined
F06D8H	UF0 configuration/interface/endpoint descriptor register 245	UF0CIE245	R/W		√		Undefined

(13/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F06D9H	UF0 configuration/interface/endpoint descriptor register 246	UF0CIE246	R/W		√		Undefined
F06DAH	UF0 configuration/interface/endpoint descriptor register 247	UF0CIE247	R/W		√		Undefined
F06DBH	UF0 configuration/interface/endpoint descriptor register 248	UF0CIE248	R/W		√		Undefined
F06DCH	UF0 configuration/interface/endpoint descriptor register 249	UF0CIE249	R/W		√		Undefined
F06DDH	UF0 configuration/interface/endpoint descriptor register 250	UF0CIE250	R/W		√		Undefined
F06DEH	UF0 configuration/interface/endpoint descriptor register 251	UF0CIE251	R/W		√		Undefined
F06DFH	UF0 configuration/interface/endpoint descriptor register 252	UF0CIE252	R/W		√		Undefined
F06E0H	UF0 configuration/interface/endpoint descriptor register 253	UF0CIE253	R/W		√		Undefined
F06E1H	UF0 configuration/interface/endpoint descriptor register 254	UF0CIE254	R/W		√		Undefined
F06E2H	UF0 configuration/interface/endpoint descriptor register 255	UF0CIE255	R/W		√		Undefined

(4) EPC peripheral control register

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
F059EH	Buffer control register	UF0BC	R/W		√		0000H

13. 6. 2 Clock control register

(1) PLL control register (PLLC)

It is a register which sets PLL operation mode.

PLLC is set using 1 bit or 8 bit memory manipulation instruction.

It changes to 01H after RESET signal

Address:F059CH After reset:01H R/W

Symbol	7	6	5	4	3	2	1	0
PLLC	0	0	0	0	0	PLLM1	PLLM	PLLSTOP

PLLM1	PLLM	PLL supply clock/ Select PLL multiplier	
		Supply clock	Select multiplier
0	0	f_{MX}/2	8Multiplier[Note1]
0	1	f_{MX}/4	12Multiplier[Note2]
1	0	f_{MX}/5	12Multiplier[Note3]

PLLSTOP	PLL Operation control
0	PLL Start
1	PLL Stop

Notes 1. When f_{MX} = 12 MHz, f_{USB} = 48 MHz

2. When f_{MX} = 16 MHz, f_{USB} = 48 MHz

3. When f_{MX} = 20 MHz, f_{USB} = 48 MHz

Cautions 1. When using USB, set PLL supply clock from default settings after settings.

<Setting procedure >

<1>Stop PLL(PLLSTOP = 1)

<2>Select PLLM1, PLLM. When(PLLM1 = 0, PLLM = 0: f_{MX}12 MHz, when PLLM1 = 0, PLLM = 1: f_{MX}16 MHz, when PLLM1 = 1, PLLM0 = 0 : f_{MX}20 MHz)

<3>Enable PLL operation(PLLSTOP = 0)

2. Execute PLL Oscillation start (PLLSTOP = 0)when USB clock (f_{USB}) supply is in stop mode(UCKCNT = 0)

3. Change in multiplication rate (PLLM, PLLM1 bit change) when PLL is in peration mode is prohibited.

4. Clock supply (UCKCNT = 1) to USB function controller is prohibited when PLL (PLLSTOP = 1) is in stop mode

5. It is not possible to stop PLL (PLLSTOP = 1) when clock supply to USB function controller is on.

6. It is not possible to change multiplication rate settings (PLLM, PLLM1 bit change) when clock supply to USB function controller is on.

Remark f_{MX}: High speed system clock frequency

(2)USB Clock control register (UCKC)

It is a register which controls USB clock supply (f_{USB}) to the USB function controller

PLLC is set using 1 bit or 8 bit memory manipulation instruction.

It changes to 00H after RESET signal.

Address:F059DH After reset:00H R/W

Symbol	7	6	5	4	3	2	1	0
UCKC	UCKCNT	0	0	0	0	0	0	0

UCKCNT	USB Clock supply control to USB function controller
0	USB Clock supply stop
1	USB Clock supply on

Caution Stop clock supply to USB function controller before switching to STOP mode.

After STOP mode release, count PLL oscillation stabilization wait time (800 μs) using software and then start clock supply to USB function controller after PLL oscillation stabilization wait time ends.

13.6.3 EPC control registers

(1) UF0 EP0NAK register (UF0E0N)

This register controls NAK of Endpoint0 (except an automatically executed request).

This register can be read or written in 8-bit units (however, bit 0 can only be read).

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read, Endpoint2, and Endpoint4, a write access to the EP0NKR bit is ignored.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW	F0540H	00H

Bit position	Bit name	Function
1	EP0NKR	This bit controls NAK to the OUT token to Endpoint0 (except an automatically executed request). It is automatically set to 1 by hardware when Endpoint0 has correctly received data. It is also cleared to 0 by hardware when the data of the UF0E0R register has been read by FW (counter value = 0). 1: Transmit NAK. 0: Do not transmit NAK (default value). Set this bit to 1 by FW when data should not be received from the USB bus for some reason even when USBF is ready for receiving data. In this case, USBF continues transmitting NAK until this bit is cleared to 0 by FW. This bit is also cleared to 0 as soon as the UF0E0R register has been cleared.
0	EP0NKW	This bit indicates how NAK to the IN token to Endpoint0 is controlled (except an automatically executed request). This bit is automatically cleared to 0 by hardware when the data of Endpoint0 is transmitted and the host correctly receives the transmitted data. The data of the UF0E0W register is retained until this bit is cleared. Therefore, it is not necessary to rewrite this bit even in the case of a retransmission request that is made if the host could not receive data correctly. To send a short packet, be sure to set the E0DED bit of the UF0DEND register to 1. This bit is automatically set to 1 when the FIFO is full. As soon as the E0DED bit of the UF0DEND register is set to 1, the EP0NKW bit is automatically set to 1 at the same time. 1: Do not transmit NAK. 0: Transmit NAK (default value). If control transfer enters the status stage while ACK cannot be correctly received in the data stage, this bit is cleared to 0 as soon as the UF0E0W register is cleared. This bit is also cleared to 0 when UF0E0W is cleared by FW.

Next, the procedure of a SETUP transaction that uses IN/OUT tokens is explained below.

(a) When IN token is used (except a request automatically executed by hardware)

FW should be used to clear the PROT bit of the UF0IS1 register to 0 after receiving the CPUDEC interrupt and before reading data from the UF0E0ST register. Next, perform processing in accordance with the request and, if it is necessary to return data by an IN token, write data to the UF0E0W register. Confirm that the PROT bit of the UF0IS1 register is 0 after writing has been completed, and set the E0DED bit of the UF0DEND register to 1. The hardware sends out data at the first IN token after the EP0NKW bit has been set to 1. If the PROT bit of the UF0IS1 register is 1, it indicates that a SETUP transaction has occurred again before completion of control transfer. In this case, clear the PROT bit of the UF0IS1 register to 0 by clearing the PROTC bit of the UF0IC1 register to 0, and then read data from the UF0E0ST register again. A request received later can be read.

(b) When OUT token is used (except a request automatically executed by hardware)

FW should be used to clear the PROT bit of the UF0IS1 register after receiving the CPUDEC interrupt and before reading data from the UF0E0ST register. Confirm that the PROT bit of the UF0IS1 register is 0 before reading data from the UF0E0R register. If the PROT bit is 1, it means that invalid data is retained. Clear the FIFO by FW (the EP0NKR bit is automatically cleared to 0). If the PROT bit of the UF0IS1 register is 0, read the data of the UF0E0L register and read as many data from the UF0E0R register as set. When reading data from the UF0E0R register has been completed (when the counter of the UF0E0R register has been cleared to 0), the hardware automatically clears the EP0NKR bit to 0.

(2) UF0 EP0NAKALL register (UF0E0NA)

This register controls NAK to all the requests of Endpoint0. It is also valid for automatically executed requests.

This register can be read or written in 8-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0NA	0	0	0	0	0	0	0	EP0NKA	F0541H	00H

Bit position	Bit name	Function
0	EP0NKA	This bit controls NAK to a transaction other than a SETUP transaction to Endpoint0 (including an automatically executed request). This bit is manipulated by FW. 1: Transmit NAK. 0: Do not transmit NAK (default value). This register is used to prevent a conflict between a write access by FW and a read access from SIE when the data used for an automatically executed request is to be changed. It postpones reflecting a write access on this bit from FW while an access from SIE is being made. Before rewriting the request data register from FW, confirm that this bit has been correctly set to 1. Setting this bit to 1 is reflected only in the following cases. • Immediately after USBF has been reset and a SETUP token has never been received • Immediately after reception of Bus Reset and a SETUP token has never been received • PID of a SETUP token has been detected • The stage has been changed to the status stage Clearing this bit to 0 is reflected immediately, except while an IN token is being received and a NAK response is being made. Setting the EP0NKA bit to 1 is reflected in the above four cases during Endpoint0 transfer, but it is reflected immediately after data has been written to the bit while Endpoint0 is transferring no data.

(3) UF0 EPNAK register (UF0EN)

This register controls NAK of endpoints other than Endpoint0.

This register can be read or written in 8-bit units (however, bits 5, 4, 1, and 0 can only be read).

The BKO2NK bit can be written only when the BKO2NKM bit of the UF0ENM register is 1 and the BKO1NK bit can be written only when the BKO1NKM bit of the UF0ENM register is 1.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read, Endpoint2, and Endpoint4, a write access to the BKO1NK and BKO2NK bits is ignored.

Be sure to clear bits 7,6 - “0”. If it is set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0EN	0	0	IT2NK	IT1NK	BKO2NK	BKO1NK	BKI2NK	BKI1NK	F0542H	00H

Bit position	Bit name	Function
5	IT2NK	This bit controls NAK to Endpoint8 (interrupt 2 transfer). It is automatically set to 1 and transmission is started when the UF0INT2 register has become full as a result of writing data to it. To send a short packet that does not make the FIFO full, set the IT2DEND bit of the UF0DEND register to 1. As soon as the IT2DEND bit has been set to 1, this bit is automatically set to 1. 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is also cleared to 0 when the UF0INT2 register has been cleared.
4	IT1NK	This bit controls NAK to Endpoint7 (interrupt 1 transfer). It is automatically set to 1 and transmission is started when the UF0INT1 register has become full as a result of writing data to it. To send a short packet that does not make the FIFO full, set the IT1DEND bit of the UF0DEND register to 1. As soon as the IT1DEND bit has been set to 1, this bit is automatically set to 1. 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is also cleared to 0 when the UF0INT1 register has been cleared.

(2/3)

Bit position	Bit name	Function
3	BKO2NK	This bit controls NAK to Endpoint4 (bulk 2 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK (default value). This bit is set to 1 only when the FIFO connected to the SIE side of the UF0BO2 register (64-byte FIFO of bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied. • Data correctly received is stored in the FIFO connected to the SIE side. • The value of the FIFO counter connected to the CPU side is 0 (completion of reading). FW should be used to read data of the UF0BO2L register when it has received the BLKO2DT interrupt request and read as many data from the UF0BO2 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 by FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 as soon as the UF0BO2 register has been cleared.
2	BKO1NK	This bit controls NAK to Endpoint2 (bulk 1 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK (default value). This bit is set to 1 only when the FIFO connected to the SIE side of the UF0BO1 register (64-byte FIFO of bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied. • Data correctly received is stored in the FIFO connected to the SIE side. • The value of the FIFO counter connected to the CPU side is 0 (completion of reading). FW should be used to read data of the UF0BO1L register when it has received the BLKO1DT interrupt request and read as many data from the UF0BO1 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 by FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 as soon as the UF0BO1 register has been cleared.

(3/4)

Bit position	Bit name	Function
1	BKI2NK	This bit controls NAK to Endpoint3 (bulk 2 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is cleared to 0 only when the FIFO connected to the SIE side of the UF0BI2 register (64-byte FIFO of bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed (the data of the UF0BI2 register is retained until transmission has been correctly completed). The bank is changed (toggle operation) when the following conditions are satisfied. • Data is correctly written to the FIFO connected to the CPU bus side (writing has been completed and the FIFO is full or the UF0DEND register is set). • The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO.
0	BKI0NK	This bit controls NAK to Endpoint1 (bulk 1 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is cleared to 0 only when the FIFO connected to the SIE side of the UF0BI1 register (64-byte FIFO of bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed (the data of the UF0BI1 register is retained until transmission has been correctly completed). The bank is changed (toggle operation) when the following conditions are satisfied. • Data is correctly written to the FIFO connected to the CPU bus side (writing has been completed and the FIFO is full or the UF0DEND register is set). • The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it by DMA while the BKI1T bit of the UF0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI1DED bit of the UF0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI1DED bit to 1 after completing writing data. When the BKI1DED bit is set to 1, a toggle operation is performed and at the same time, this bit is automatically set to 1. This bit is also cleared to 0 as soon as the UF0BI1 register has been cleared.

(4) UF0 EPNAK mask register (UF0ENM)

This register controls masking a write access to the UF0EN register.

This register can be read or written in 8-bit units.

Be sure to clear bits 7- 4, 1, and 0 to “0”. If it is set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ENM	0	0	0	0	BKO2NKM	BKO1NKM	0	0	F0543H	00H

Bit position	Bit name	Function
3	BKO2NKM	This bit specifies whether a write access to bit 3 (BKO2NK) of the UF0EN register is masked or not. 1: Do not mask. 0: Mask (default value).
2	BKO1NKM	This bit specifies whether a write access to bit 2 (BKO1NK) of the UF0EN register is masked or not. 1: Do not mask. 0: Mask (default value).

(5) UF0 SNDSIE register (UF0SDS)

This register performs manipulation such as no handshake. It can directly manipulate the pins of SIE.

This register can be read or written in 8-bit units.

Be sure to clear bit 2 to “0”. If it is set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0SDS	0	0	0	0	SNDSTL	0	0	RSUMIN	F0544H	00H

Bit position	Bit name	Function
3	SNDSTL	This bit makes Endpoint0 issue a STALL handshake. Setting this bit to 1 if a request for CPUDEC processing is not supported by the system results in a STALL handshake response. If an unsupported wValue is sent by the SET_CONFIGURATION or SET_INTERFACE request, the hardware sets this bit to 1. If a problem occurs in Endpoint0 due to overrun of an automatically executed request, this bit is also set to 1. However, the E0HALT bit of the UF0E0SL register is not set to 1. 1: Respond with STALL handshake. 0: Do not respond with STALL handshake (default value). This bit is cleared to 0 and the handshake response to the bus is other than STALL when the next SETUP token is received. To set the SNDSTL bit to 1 by FW, do not write data to the UF0E0W register. Depending on the timing of setting this bit, the STALL response is not made in time, and it may be made to the next transfer after a NAK response has been made. Setting this bit is valid only while an FW-executed request is under execution when this bit is set to 1. It is automatically cleared to 0 when the next SETUP token is received. **Remark** The SNDSTL bit is valid only for an FW-executed request.
0	RSUMIN	This bit outputs the Resume signal onto the USB bus. Writing this bit is invalid unless the RMWK bit of the UF0DSTL register is set to 1. 1: Generate the Resume signal. 0: Do not generate the Resume signal (default value). While this bit is set to 1, the Resume signal continues to be generated. Clear this bit to 0 by FW after a specific time has elapsed. Because the signal is internally sampled at the clock, the operation is guaranteed only while CLK is supplied. Care must be exercised when CLK of the system is stopped.

(6) UF0 CLR request register (UF0CLR)

This register indicates the target of the received CLEAR_FEATURE request.

This register is read-only, in 8-bit units.

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CLR	0	CLREP7	CLREP4	CLREP3	CLREP2	CLREP1	CLREP0	CLRDEV	F0545H	00H

Bit position	Bit name	Function
6 - 1	CLREPn	These bits indicate that a CLEAR_FEATURE Endpoint n request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
0	CLRDEV	This bit indicates that a CLEAR_FEATURE Device request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)

Remark n = 0- 4, 7, 8

(7) UF0 SET request register (UF0SET)

This register indicates the target of the automatically processed SET_XXXX (except SET_INTERFACE) request.

This register is read-only, in 8-bit units.

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0SET	SETCON	0	0	0	0	SETEP	0	SETDEV	F0546H	00H

Bit position	Bit name	Function
7	SETCON	This bit indicates that a SET_CONFIGURATION request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
2	SETEP	This bit indicates that a SET_FEATURE Endpoint n request (n = 0- 4, 7, 8) is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
0	SETDEV	This bit indicates that a SET_FEATURE Device request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)

(8) UF0 EP status 0 register (UF0EPS0)

This register indicates the USB bus status and the presence or absence of register data.

This register is read-only, in 8-bit units.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate writing to the UF0FIC0 and UF0FIC1 registers from reading from the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS0	IT2	IT1	BKOUT2	BKOUT1	BKIN2	BKIN1	EP0W	EP0R	F0547H	00H

Bit position	Bit name	Function
7	IT2	These bits indicate that data is in the UF0INT2 register (FIFO). By setting the IT2DEND bit of the UF0DEND register to 1, the status in which data is in the UF0INT2 register can be created even if data is not written to the register (Null data transmission). As soon as the IT2DEND bit of the UF0DEND register is set to 1 even when the counter of the UF0INT2 register is 0, this bit is set to 1 by hardware. It is cleared to 0 after correct transmission. 1: Data is in the register. 0: No data is in the register (default value).
6	IT1	These bits indicate that data is in the UF0INT1 register (FIFO). By setting the IT1DEND bit of the UF0DEND register to 1, the status in which data is in the UF0INT1 register can be created even if data is not written to the register (Null data transmission). As soon as the IT1DEND bit of the UF0DEND register is set to 1 even when the counter of the UF0INT1 register is 0, this bit is set to 1 by hardware. It is cleared to 0 after correct transmission. 1: Data is in the register. 0: No data is in the register (default value).
5, 4	BKOUTn	These bits indicate that data is in the UF0BOn register (FIFO) connected to the CPU side. When the FIFO configuring the UF0BOn register is toggled, this bit is automatically set to 1 by hardware. It is automatically cleared to 0 by hardware when reading the UF0BOn register (FIFO) connected to the CPU side has been completed (counter value = 0). It is not set to 1 when Null data is received (toggling the FIFO does not take place either). 1: Data is in the register. 0: No data is in the register (default value).

Remark n = 1, 2

(2/2)

Bit position	Bit name	Function
3, 2	BKINn	These bits indicate that data is in the UF0BIn register (FIFO) connected to the CPU side. By setting the BKInDED bit of the UF0DEND register to 1, the status in which data is in the UF0BIn register can be created even if data is not written to the register (Null data transmission). As soon as the BKInDED bit of the UF0DEND register has been set to 1 while the counter of the UF0BIn register is 0, this bit is set to 1 by hardware. It is cleared to 0 when a toggle operation is performed. 1: Data is in the register. 0: No data is in the register (default value).
1	EP0W	This bit indicates that data is in the UF0E0W register (FIFO). By setting the E0DED bit of the UF0DEND register to 1, the status in which data is in the UF0E0W register can be created even if data is not written to the register (Null data transmission). As soon as the E0DED bit of the UF0DEND register is set to 1 even when the counter of the UF0E0W register is 0, this bit is set to 1 by hardware. It is cleared to 0 after correct transmission. 1: Data is in the register. 0: No data is in the register (default value).
0	EP0R	This bit indicates that data is in the UF0E0R register (FIFO). It is automatically cleared to 0 by hardware when reading the UF0E0R register (FIFO) has been completed (counter value = 0). It is not set to 1 if Null data is received. 1: Data is in the register. 0: No data is in the register (default value).

(9) UF0 EP status 1 register (UF0EPS1)

This register indicates the USB bus status and the presence or absence of register data.

This register is read-only, in 8-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS1	RSUM	0	0	0	0	0	0	0	F0548H	00H

Bit position	Bit name	Function
7	RSUM	This bit indicates that the USB bus is in the Resume status. This bit is meaningful only when an interrupt request is generated. 1: Suspend status 0: Resume status (default value) Because sampling is internally performed with the clock, the operation is guaranteed only when CLK is supplied. Care must be exercised when CLK of the system is stopped. The INTRSUM1 signal of SIE operates even when CLK is stopped. It can therefore be supported by making the interrupt control register (UFIC1) valid or lowering the frequency of CLK to the USBF. This bit is automatically cleared to 0 when it is read.

(10) UF0 EP status 2 register (UF0EPS2)

This register indicates the USB bus status and the presence or absence of register data.

This register is read-only, in 8-bit units.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS2	0	HALT8	HALT7	HALT4	HALT3	HALT2	HALT1	HALT0	F0549H	00H

Bit position	Bit name	Function
6 – 0	HALTn	These bits indicate that Endpoint n is currently stalled. They are set to 1 when a stall condition, such as occurrence of an overrun and reception of an undefined request, is satisfied. These bits are automatically set to 1 by hardware. 1: Endpoint is stalled. 0: Endpoint is not stalled (default value). The SNDSTL bit is set to 1 as soon as the HALT0 bit has been set to 1 as a result of occurrence of an overrun or reception of an undefined request. If the next SETUP token is received in this status, the SNDSTL bit is cleared to 0 and, therefore, the HALT0 bit is also cleared to 0. If Endpoint0 is stalled by the SET_FEATURE Endpoint0 request, this bit is not cleared to 0 until the CLEAR_FEATURE Endpoint0 request is received or Halt Feature is cleared by FW. If the GET_STATUS Endpoint0, CLEAR_FEATURE Endpoint0, or SET_FEATURE Endpoint0 request is received, or if a request to be processed by FW is received due to the CPUDEC interrupt request, the HALT0 bit is masked and cleared to 0, until the next SETUP token is received. The HALTn bit is not cleared to 0 until Endpoint n receives the CLEAR_FEATURE Endpoint request, Halt Feature is cleared by the SET_INTERFACE or SET_CONFIGURATION request to the interface to which the endpoint is linked, or Halt Feature is cleared by FW. When the SET_INTERFACE or SET_CONFIGURATION request is correctly processed, the Halt Feature of all the target endpoints, except Endpoint0, is cleared after the request has been processed, even if the wValue is the same as the currently set value, and these bits are also cleared to 0. Halt Feature of Endpoint0 cannot be cleared if it is set because the STALL response is made in response to the SET_INTERFACE and SET_CONFIGURATION requests.

Remark n = 0- 4, 7, 8

(11) UF0 INT status 0 register (UF0IS0)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSB signal becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSB) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC0 register.

Caution In the USBF, multiple interrupt sources, such as Bus Reset, Resume, and Short, are ORed internally and are issued as a single interrupt request (INTUSB). Therefore, in the case of the occurrence of multiple interrupt sources, they are ORed and issued as an INTUSB interrupt request.

For example, if a Bus Reset interrupt source and Resume interrupt source occur, the two sources are ORed and an INTUSB interrupt request is issued.

Under these conditions, if the Bus Reset interrupt source is cleared to 0 (UF0IC0.BUSRSTC = 0), the 78K0R/KC3-L, KE3-L internal INTUSB interrupt request may remain set to 1 since the Resume interrupt source will still remain. The new interrupt request flag (US0BIC.US0BIF), therefore, might not be set to 1.

In this case, after performing clear processing for each interrupt request with the INTUSB interrupt servicing routine, confirm the flag status for the UF0IS0 and UF0IS1 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS0	BUSRST	RSUSPD	0	0	0	SETRQ	CLRRQ	EPHALT	F0550H	00H

Bit position	Bit name	Function
7	BUSRST	This bit indicates that Bus Reset has occurred. 1: Bus Reset has occurred (interrupt request is generated). 0: Not Bus Reset status (default value)
6	RSUSPD	This bit indicates that the Resume or Suspend status has occurred. Reference bit 7 of the UF0EPS1 register by FW. 1: Resume or Suspend status has occurred (interrupt request is generated). 0: Resume or Suspend status has not occurred (default value).
2	SETRQ	This bit indicates that the SET_XXXX request to be automatically processed has been received and automatically processed (XXXX = CONFIGURATION or FEATURE). 1: SET_XXXX request to be automatically processed has been received (interrupt request is generated). 0: SET_XXXX request to be automatically processed has not been received (default value). This bit is set to 1 after completion of the status stage. Reference the UF0SET register to identify what is the target of the request. This bit is not automatically cleared to 0 even if the UF0SET register is read by FW. The EPHALT bit is also set to 1 when the SET_FEATURE Endpoint request has been received.

(2/2)

Bit position	Bit name	Function
1	CLRRQ	This bit indicates that the CLEAR_FEATURE request has been received and automatically processed. 1: CLEAR_FEATURE request has been received (interrupt request is generated). 0: CLEAR_FEATURE request has not been received (default value). This bit is set to 1 after completion of the status stage. Reference the UF0CLR register to identify what is the target of the request. This bit is not automatically cleared to 0 even if the UF0CLR register is read by FW.
0	EPHALT	This bit indicates that an endpoint has stalled. 1: Endpoint has stalled (interrupt request is generated). 0: Endpoint has not stalled (default value). This bit is also set to 1 when an endpoint has stalled by setting FW. Identify the endpoint that has stalled, by referencing the UF0EPS2 register. This bit is not automatically cleared to 0 even when the CLEAR_FEATURE Endpoint, SET_INTERFACE, or SET_CONFIGURATION request is received. It is not automatically cleared to 0, either, if the next SETUP token is received in case of overrun of Endpoint0. **Caution Even if Halt Feature of Endpoint0 is set and this interrupt request is generated, bit 0 of the UF0EPS2 register is masked and cleared to 0 between when a SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, or GET_STATUS Endpoint0 request, or FW-processed request is received and when a SETUP token other than the above is received.**

(12) UF0 INT status 1 register (UF0IS1)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSB signal becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSB) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC1 register. However, the SUCES and STG bits of the UF0IS1 register are automatically cleared to 0 when the next SETUP token has been received.

Caution In the USBF, multiple interrupt sources, such as Bus Reset, Resume, and Short, are ORed internally and are issued as a single interrupt request (INTUSB). Therefore, in the case of the occurrence of multiple interrupt sources, they are ORed and issued as an INTUSB interrupt request.

For example, if a Bus Reset interrupt source and Resume interrupt source occur, the two sources are ORed and an INTUSB interrupt request is issued.

Under these conditions, if the Bus Reset interrupt source is cleared to 0 (UF0IC0.BUSRSTC = 0), the 78K0R/KC3-L, KE3-L internal INTUSB interrupt request may remain set to 1 since the Resume interrupt source will still be remaining. The new interrupt request flag (US0BIC.US0BIF), therefore, might not be set to 1.

In this case, after performing clear processing for each interrupt request with the INTUSB interrupt servicing routine, confirm the flag status for the UF0IS0 and UF0IS1 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS1	0	E0IN	E0INDT	E0ODT	SUCES	STG	PROT	CPU DEC	F0551H	00H

Bit position	Bit name	Function
6	E0IN	This bit indicates that an IN token for Endpoint0 has been received and that the hardware has automatically transmitted NAK. 1: IN token is received and NAK is transmitted (interrupt request is generated). 0: IN token is not received (default value).
5	E0INDT	This bit indicates that data has been correctly transmitted from the UF0E0W register. 1: Transmission from UF0E0W register is completed (interrupt request is generated). 0: Transmission from UF0E0W register is not completed (default value). Data is transmitted in synchronization with the IN token next to the one that set the EP0NKW bit of the UF0E0N register to 1. This bit is automatically set to 1 by hardware when the host correctly receives that data. It is also set to 1 even if the data is a Null packet. This bit is automatically cleared to 0 by hardware when the first write access is made to the UF0E0W register.

(2/2)

Bit position	Bit name	Function
4	E0ODT	This bit indicates that data has been correctly received in the UF0E0R register. 1: Data is in UF0E0R register (interrupt request is generated). 0: Data is not in UF0E0R register (default value). This bit is automatically set to 1 by hardware when data has been correctly received. At the same time, the EP0R bit of the UF0EPS0 register is also set to 1. If a Null packet has been received, this bit is not set to 1. It is automatically cleared to 0 by hardware when the FW reads the UF0E0R register and the value of the UF0E0L register becomes 0.
3	SUCES	This bit indicates that either an FW-processed or hardware-processed request has been received and that the status stage has been correctly completed. 1: Control transfer has been correctly processed (interrupt request is generated). 0: Control transfer has not been processed correctly (default value). This bit is set to 1 upon completion of the status stage. It is automatically cleared to 0 by hardware when the next SETUP token is received. This bit is also set to 1 when data with Data PID of 0 (Null data) is received in the status stage of control transfer.
2	STG	This bit is set to 1 when the stage of control transfer has changed to the status stage. It is valid for both FW-processed and hardware-processed requests. This bit is also set to 1 when the stage of control transfer (without data) has changed to the status stage. 1: Status stage (interrupt request is generated) 0: Not status stage (default value) This bit is automatically cleared to 0 by hardware when the next SETUP token is received. It is also set to 1 when the stage of control transfer has changed to the status stage while ACK cannot be correctly received in the data stage. In this case, the EP0NKW bit of the UF0E0N register is also cleared to 0 as soon as the UF0E0W register has been cleared, if the FW is processing control transfer (read).
1	PROT	This bit indicates that a SETUP token has been received. It is valid for both FW-processed and hardware-processed requests. 1: SETUP token is correctly received (interrupt request is generated). 0: SETUP token is not received (default value). This bit is set to 1 when data has been correctly received in the UF0E0ST register. Clear this bit to 0 by FW when the first read access is made to the UF0E0ST register. If it is not cleared to 0 by FW, reception of the next SETUP token cannot be correctly recognized. This bit is used to accurately recognize that a SETUP transaction has been executed again during control transfer. If the SETUP transaction is re-executed during control transfer and if a second request is executed by hardware, the CPUDEC bit is not set to 1, but the PROT bit can be used for recognition of the re-execution.
0	CPUDEC	This bit indicates that the UF0E0ST register has a request that is to be decoded by FW. 1: Data is in UF0E0ST register (interrupt request is generated). 0: Data is not in UF0E0ST register (default value). This bit is automatically cleared to 0 by hardware when all the data of the UF0E0ST register is read.

(13) UF0 INT status 2 register (UF0IS2)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSB signal becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSB) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC2 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS2	BKI2IN	BKI2DT	BKI1IN	BKI1DT	0	0	IT2DT	IT1DT	F0552H	00H

Bit position	Bit name	Function
7, 5	BKInIN	These bits indicate that an IN token has been received in the UF0BIn register (Endpoint m) and that NAK has been returned. 1: IN token is received and NAK is transmitted (interrupt request is generated). 0: IN token is not received (default value).
6, 4	BKInDT	These bits indicate that the FIFO of the UF0BIn register (Endpoint m) has been toggled. This means that data can be written to Endpoint m. 1: FIFO has been toggled (interrupt request is generated). 0: FIFO has not been toggled (default value). The data written to Endpoint m is transmitted in synchronization with the IN token next to the one that set the BKInNK bit of the UF0EN register to 1. When the FIFO has been toggled and then data can be written from the CPU, this bit is automatically set to 1 by hardware. It is also set to 1 when the FIFO has been toggled, even if the data is a Null packet. This bit is automatically cleared to 0 by hardware when the first write access is made to the UF0BIn register.
1, 0	ITnDT	These bits indicate that data has been correctly received from the UF0INTn register (Endpoint x). 1: Transmission is completed (interrupt request is generated). 0: Transmission is not completed (default value). Data is transmitted in synchronization with the IN token next to the one that set the ITnNK bit of the UF0EN register to 1. This bit is automatically set to 1 by hardware when the host has correctly received that data. It is automatically cleared to 0 by hardware when the first write access is made to the UF0INTn register. This bit is also set to 1 even when the data is a Null packet.

Remark n = 1, 2
m = 1 and x = 7 where n = 1
m = 3 where n = 2

(14) UF0 INT status 3 register (UF0IS3)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSB signal becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSB) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC3 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS3	BKO2FL	BKO2NL	BKO2 NAK	BKO2DT	BKO1FL	BKO1NL	BKO1 NAK	BKO1DT	F0553H	00H

Bit position	Bit name	Function
7, 3	BKOnFL	These bits indicate that data has been correctly received in the UF0BOn register (Endpoint m) and that both the FIFOs of the CPU and SIE hold the data. 1: Received data is in both the FIFOs of the UF0BOn register (interrupt request is generated). 0: Received data is not in the FIFO on the SIE side of the UF0BOn register (default value). If data is held in both the FIFOs of the CPU and SIE, these bits are automatically set to 1 by hardware. They are automatically cleared to 0 by hardware when the FIFO is toggled.
6, 2	BKOnNL	These bits indicate that a Null packet (packet with a length of 0) has been received in the UF0BOn register (Endpoint m). 1: Null packet is received (interrupt request is generated). 0: Null packet is not received (default value). These bits are set to 1 immediately after reception of a Null packet when the FIFO is empty. They are set to 1 when the FIFO on the CPU side has been completely read if data is in that FIFO.
5, 1	BKOnNAK	These bits indicate that an OUT token has been received to the UF0BOn register (Endpoint m) and that NAK has been returned. 1: OUT token is received and NAK is transmitted (interrupt request is generated). 0: OUT token is not received (default value).

Remark **n = 1, 2**
m = 2 where n = 1
m = 4 where n = 2

(2/2)

Bit position	Bit name	Function
4, 0	BKOnDT	These bits indicate that data has been correctly received in the UF0BOn register (Endpoint m). 1: Reception has been completed correctly (interrupt request is generated). 0: Reception has not been completed (default value). These bits are automatically set to 1 by hardware when data has been correctly received and the FIFO has been toggled. At the same time, the corresponding bits of the UF0EPS0 register are also set to 1. They are not set to 1 when the data is a Null packet. These bits are automatically cleared to 0 by hardware when the value of the UF0BOnL register becomes 0 as a result of reading the UF0BOn register by FW. These bits are automatically cleared to 0 when all the contents of the FIFO on the CPU side have been read. However, the interrupt request is not cleared if data is in the FIFO on the SIE side at this time, and the INTRSUM signal does not become inactive. The signal is kept active if data is successively received.

Remark n = 1, 2
m = 2 where n = 1
m = 4 where n = 2

(15) UF0 INT status 4 register (UF0IS4)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSB signal becomes active.

This register is read-only, in 8-bit units.

If an interrupt request (INTUSB) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC4 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS4	0	0	SETINT	0	0	0	0	0	F0554H	00H

Bit position	Bit name	Function
5	SETINT	This bit indicates that the SET_INTERFACE request has been received and automatically processed. 1: The request has been automatically processed (interrupt request is generated). 0: The request has not been automatically processed (default value). The current setting of this bit can be identified by reading the UF0ASS or UF0IFn register (n = 0- 4).

(16) UF0 INT mask 0 register (UF0IM0)

This register controls masking of the interrupt sources indicated by the UFBF register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSB) by writing 1 to the corresponding bit of this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM0	BUS RSTM	RSU SPDM	1 Note	1 Note	1 Note	SET RQM	CLR RQM	EP HALTM	F0557H	00H

Bit position	Bit name	Function
7	BUSRSTM	This bit masks the Bus Reset interrupt. 1: Mask 0: Do not mask (default value)
6	RSUSPDM	This bit masks the Resume/Suspend interrupt. 1: Mask 0: Do not mask (default value)
2	SETRQM	This bit masks the SET_RQ interrupt. 1: Mask 0: Do not mask (default value)
1	CLRRQM	This bit masks the CLR_RQ interrupt. 1: Mask 0: Do not mask (default value)
0	EPHALTM	This bit masks the EP_Halt interrupt. 1: Mask 0: Do not mask (default value)

(17) UF0 INT mask 1 register (UF0IM1)

This register controls masking of the interrupt sources indicated by the UFBF register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSB) by writing 1 to the corresponding bit of this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM1	0	E0INM	E0 INDTM	E0 ODTM	SUCESM	STGM	PROTM	CPU DECM	F0558H	00H

Bit position	Bit name	Function
6	E0INM	This bit masks the EP0IN interrupt. 1: Mask 0: Do not mask (default value)
5	E0INDTM	This bit masks the EP0INDT interrupt. 1: Mask 0: Do not mask (default value)
4	E0ODTM	This bit masks the EP0OUTDT interrupt. 1: Mask 0: Do not mask (default value)
3	SUCESM	This bit masks the Success interrupt. 1: Mask 0: Do not mask (default value)
2	STGM	This bit masks the Stg interrupt. 1: Mask 0: Do not mask (default value)
1	PROTM	This bit masks the Protect interrupt. 1: Mask 0: Do not mask (default value)
0	CPUDECM	This bit masks the CPUDEC interrupt. 1: Mask 0: Do not mask (default value)

(18) UF0 INT mask 2 register (UF0IM2)

This register controls masking of the interrupt sources indicated by the UFBF register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSB) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM2	BKI2INM	BKI2 DTM	BKI1INM	BKI1 DTM	0	0	IT2DTM	IT1DTM	F0559H	00H

Bit position	Bit name	Function
7, 5	BKInINM	These bits mask the BLKInIN interrupt. 1: Mask 0: Do not mask (default value)
6, 4	BKInDTM	These bits mask the BLKInDT interrupt. 1: Mask 0: Do not mask (default value)
1,0	ITnDTM	These bits mask the INTnDT interrupt. 1: Mask 0: Do not mask (default value)

Remark n = 1, 2

(19) UF0 INT mask 3 register (UF0IM3)

This register controls masking of the interrupt sources indicated by the UFBF register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSB) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM3	BKO2 FLM	BKO2 NLM	BKO2 NAKM	BKO2 DTM	BKO1 FLM	BKO1 NLM	BKO1 NAKM	BKO1 DTM	F055AH	00H

Bit position	Bit name	Function
7, 3	BKOnFLM	These bits mask the BLKOnFL interrupt. 1: Mask 0: Do not mask (default value)
6, 2	BKOnNLM	These bits mask the BLKOnNL interrupt. 1: Mask 0: Do not mask (default value)
5, 1	BKOnNAKM	These bits mask the BLKOnNK interrupt. 1: Mask 0: Do not mask (default value)
4, 0	BKOnDTM	These bits mask the BLKOnDT interrupt. 1: Mask 0: Do not mask (default value)

Remark n = 1, 2

(20) UF0 INT mask 4 register (UF0IM4)

This register controls masking of the interrupt sources indicated by the UFBF register.

This register can be read or written in 8-bit units.

FW can mask occurrence of an interrupt request from USBF (INTUSB) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM4	1 Note	1 Note	SETINTM	1 Note	1 Note	1 Note	1 Note	1 Note	F055BH	00H

Bit position	Bit name	Function
5	SETINTM	This bit masks the SET_INT interrupt. 1: Mask 0: Do not mask (default value)

(21) UF0 INT clear 0 register (UF0IC0)

This register controls clearing the interrupt sources indicated by the UF0IS0 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC0	BUS RSTC	RSU SPDC	1	1	1	SET RQC	CLR RQC	EP HALTC	F055EH	FFH

Bit position	Bit name	Function
7	BUSRSTC	This bit clears the Bus Reset interrupt. 0: Clear
6	RSUSPDC	This bit clears the Resume/Suspend interrupt. 0: Clear
2	SETRQC	This bit clears the SET_RQ interrupt. 0: Clear
1	CLRRQC	This bit clears the CLR_RQ interrupt. 0: Clear
0	EPHALTC	This bit clears the EP_Halt interrupt. 0: Clear

(22) UF0 INT clear 1 register (UF0IC1)

This register controls clearing the interrupt sources indicated by the UF0IS1 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC1	1	E0INC	E0 INDTC	E0ODTC	SUCESC	STGC	PROTC	CPU DECC	F055FH	FFH

Bit position	Bit name	Function
6	E0INC	This bit clears the EP0IN interrupt. 0: Clear
5	E0INDTC	This bit clears the EP0INDT interrupt. 0: Clear
4	E0ODTC	This bit clears the EP0OUTDT interrupt. 0: Clear
3	SUCESC	This bit clears the Success interrupt. 0: Clear
2	STGC	This bit clears the Stg interrupt. 0: Clear
1	PROTC	This bit clears the Protect interrupt. 0: Clear
0	CPUDECC	This bit clears the CPUDEC interrupt. 0: Clear

(23) UF0 INT clear 2 register (UF0IC2)

This register controls clearing the interrupt sources indicated by the UF0IS2 register.

This register is write-only, in 8-bit units. If this register is read, the value 00H is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 3, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC2	BKI2INC	BKI2 DTC	BKI1INC	BKI1 DTC	1	1	IT2DTC	IT1DTC	F0560H	FFH

Bit position	Bit name	Function
7, 5	BKInINC	These bits clear the BLKInIN interrupt. 0: Clear
6, 4	BKInDTC	These bits clear the BLKInDT interrupt. 0: Clear
1, 0	ITnDTC	These bits clear the INTnDT interrupt. 0: Clear

Remark n = 1, 2

(24) UF0 INT clear 3 register (UF0IC3)

This register controls clearing the interrupt sources indicated by the UF0IS3 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC3	BKO2 FLC	BKO2 NLC	BKO2 NAKC	BKO2 DTC	BKO1 FLC	BKO1 NLC	BKO1 NAKC	BKO1 DTC	F0561H	FFH

Bit position	Bit name	Function
7, 3	BKOnFLC	These bits clear the BLKOnFL interrupt. 0: Clear
6, 2	BKOnNLC	These bits clear the BLKOnNL interrupt. 0: Clear
5, 1	BKOnNAKC	These bits clear the BLKOnNK interrupt. 0: Clear
4, 0	BKOnDTC	These bits clear the BLKOnDT interrupt. 0: Clear

Remark n = 1, 2

(25) UF0 INT clear 4 register (UF0IC4)

This register controls clearing the interrupt sources indicated by the UF0IS4 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC4	1	1	SETINTC	1	1	1	1	1	F0562H	FFH

Bit position	Bit name	Function
5	SETINTC	This bit clears the SET_INT interrupt. 0: Clear

(26) UFO FIFO clear 0 register (UF0FIC0)

This register clears each FIFO.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

FW can clear the target FIFO by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 4, 7, 8) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0FIC0	BKI2SC	BKI2CC	BKI1SC	BKI1CC	ITR2C	ITR1C	EP0WC	EP0RC	F0570H	00H

Bit position	Bit name	Function
7, 5	BKInSC	These bits clear only the FIFO on the SIE side of the UF0BIn register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint m is being processed with the BKInNK bit set to 1. The BKInNK bit is automatically cleared to 0 by clearing the FIFO. Make sure that the FIFO on the CPU side is empty when these bits are used.
6, 4	BKInCC	These bits clear only the FIFO on the CPU side of the UF0BIn register (reset the counter). 1: Clear
3	ITR2C	These bits clear the UF0INT2 register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint 8 is being processed with the IT2NK bit set to 1. The IT2NK bit is automatically cleared to 0 by clearing the FIFO.
2	ITR1C	These bits clear the UF0INT1 register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint 7 is being processed with the IT1NK bit set to 1. The IT1NK bit is automatically cleared to 0 by clearing the FIFO.
1	EP0WC	This bit clears the UF0E0W register (resets the counter). 1: Clear Writing this bit is invalid while an IN token for Endpoint0 is being processed with the EP0NKW bit set to 1. The EP0NKW bit is automatically cleared to 0 by clearing the FIFO.
0	EP0RC	This bit clears the UF0E0R register (resets the counter). 1: Clear When the EP0NKR bit is set to 1 (except when it has been set by FW), the EP0NKR bit is automatically cleared to 0 by clearing the FIFO.

Remark **n = 1, 2**
m = 1 where n = 1
m = 3 where n = 2

(27) UF0 FIFO clear 1 register (UF0FIC1)

This register clears each FIFO.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

FW can clear the target FIFO by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2, 4) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0FIC1	0	0	0	0	BKO2C	BKO2CC	BKO1C	BKO1CC	F0571H	00H

Bit position	Bit name	Function
3, 1	BKOnC	These bits clear the FIFOs on both the SIE and CPU sides of the UF0BOn register (reset the counter). 1: Clear When the BKOnNK bit is set to 1 (except when it has been set by FW), the BKOnNK bit is automatically cleared to 0 by clearing the FIFO.
2, 0	BKOnCC	These bits clear only the FIFO on the CPU side of the UF0BOn register (reset the counter). 1: Clear When the BKOnNK bit is set to 1 (except when it has been set by FW), the BKOnNK bit is automatically cleared to 0 by clearing the FIFO.

Remark n = 1, 2

(28) UF0 data end register (UF0DEND)

This register reports the end of writing to the transmission system.

This register is write-only, in 8-bit units.

FW can start data transfer of the target endpoint by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1- 4, 7, 8) and the current setting of the interface.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0DEND	0	0	0	IT2DEND	IT1DEND	BKI2DED	BKI1DED	E0DED	F0575H	00H

Bit position	Bit name	Function
4	IT2DEND	Set these bits to 1 to transmit the data of the UF0INT2 register. When these bits are set to 1, the IT2NK bit is set to 1 and data transfer is executed. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the ITR2C bit of the UF0FIC1 register is set to 1 and then these bits are set to 1 (counter of UF0INT2 register = 0 and the corresponding bit of the UF0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0INT2 register and if these bits are set to 1 (counter of UF0INT2 register ≠ 0 and the corresponding bit of the UF0EPS0 register = 1), a short packet is transmitted. These bits are automatically controlled by hardware when the FIFO is full.
3	IT1DEND	Set these bits to 1 to transmit the data of the UF0INT1 register. When these bits are set to 1, the IT1NK bit is set to 1 and data transfer is executed. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the ITR1C bit of the UF0FIC0 register is set to 1 and then these bits are set to 1 (counter of UF0INT1 register = 0 and the corresponding bit of the UF0EPS1 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0INT1 register and if these bits are set to 1 (counter of UF0INT1 register ≠ 0 and the corresponding bit of the UF0EPS0 register = 1), a short packet is transmitted. These bits are automatically controlled by hardware when the FIFO is full.

Remark n = 1, 2

(2/2)

Bit position	Bit name	Function
2, 1	BKInDED	Set these bits to 1 when writing transmit data to the UF0BIn register has been completed. When these bits are set to 1, the FIFO is toggled as soon as possible, the BKInNK bit is set to 1, and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). These bits control the FIFO on the CPU side. If the BKInCC bit of the UF0FIC0 register is set to 1 and then these bits are set to 1 (counter of UF0BIn register = 0), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0BIn register and if these bits are set to 1 (counter of UF0BIn register ≠ 0), and if the FIFO is not full, a short packet is transmitted.
0	E0DED	Set this bit to 1 to transmit data of the UF0E0W register. When this bit is set to 1, the EP0NKW bit is set to 1 and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the EP0WC bit of the UF0FIC0 register is set to 1 and if this bit is set to 1 (counter of UF0E0W register = 0 and bit 1 of UF0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0E0W register and if this bit is set to 1 (counter of UF0E0W register ≠ 0 and bit 1 of the UF0EPS0 register = 1), and if the FIFO is not full, a short packet is transmitted.

Remark n = 1, 2

(29) UF0 GPR register (UF0GPR)

This register controls USBF and the USB interface.

This register is write-only, in 8-bit units. If this register is read, 00H is read. Be sure to clear bits 7 to 1 to “0”.

FW can reset the USBF by writing 1 to bit 0 of this register. This bit is automatically cleared to 0 after 1 has been written to it. Writing 0 to this bit is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0GPR	0	0	0	0	0	0	CONNECT	MRST	F0577H	00H

Bit position	Bit name	Function
1	CONNECT	Set output level of USBPUC pin which controls the on/ off function of pull up resistor connecting to D+ 0: USBPUC pin= Low level 1: USBPUC pin= High level See 13.3.2 connection configuration for USBPUC pin connection.
0	MRST	Set this bit to 1 to reset USBF. 1: Reset Actually, USBF is reset two USB clocks after this bit has been set to 1 by FW and the write signal has become inactive. Resetting USBF by the MRST bit while the system clock is operating has the same result as resetting by the $\overline{\text{RESET}}$ pin (hardware reset) (register value back to default value).

(30) UF0 mode control register (UF0MODC)

This register controls CPUDEC processing.

This register can be read or written in 8-bit units.

By setting each bit of this register, the setting of the UF0MODS register can be changed. The bit of this register is automatically cleared to 0 only at hardware reset and when the MRST bit of the UF0GPR register has been set to 1.

Even if the bit of this register has automatically been set to 1 by hardware, the setting by FW takes precedence.

Be sure to clear bits 7 and 5 to 2 to “0”. If they are set to 1, the operation is not guaranteed.

Caution This register is provided for debugging purposes. Usually, do not set this register except for verifying the operation or when a special mode is used.

	7	6	5	4	3	2	1	0	Address	After reset
UF0MODC	0	CDC GDST	0	0	0	0	0	0	F057AH	00H

Bit position	Bit name	Function
6	CDCGDST	Set this bit to 1 to switch the GET_DESCRIPTOR Configuration request to CPUDEC processing. By setting this bit to 1, the CDCGD bit of the UF0MODS register can be forcibly set to 1. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing (sets the CDCGD bit of the UF0MODS register to 1). 0: Automatically process the GET_DESCRIPTOR Configuration request (default value).

(31) UF0 mode status register (UF0MODS)

This register indicates the configuration status.

This register is read-only, in 8-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
UF0MODS	0	CDCGD	0	MPACK	DFLT	CONF	0	0	F057CH	00H

Bit position	Bit name	Function
6	CDCGD	This bit specifies whether CPUDEC processing is performed for the GET_DESCRIPTOR Configuration request. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing. 0: Automatically process the GET_DESCRIPTOR Configuration request (default value).
4	MPACK	This bit indicates the transmit packet size of Endpoint0. 1: Transmit a packet of other than 8 bytes. 0: Transmit a packet of 8 bytes (default value). This bit is automatically set to 1 by hardware after the GET_DESCRIPTOR Device request has been processed (on normal completion of the status stage). It is not cleared to 0 until the USBF has been reset (it is not cleared to 0 by Bus Reset). If this bit is not set to 1, the hardware transfers only the automatically-executed request in 8-byte units. Therefore, even if data of more than 8 bytes is sent by the OUT token to be processed by FW before completion of the GET_DESCRIPTOR Device request, the data is correctly received. This bit is ignored if the size of Endpoint0 is 8 bytes.
3	DFLT	This bit indicates the default status (DFLT bit = 1). 1: Enables response. 0: Disables response (always no response) (default value). This bit is automatically set to 1 by Bus Reset. The transaction for all the endpoints is not responded to until this bit is set to 1.
2	CONF	This bit indicates whether the SET_CONFIGURATION request has been completed. 1: SET_CONFIGURATION request has been completed. 0: SET_CONFIGURATION request has not been completed (default value). This bit is set to 1 when Configuration value = 1 is received by the SET_CONFIGURATION request. Unless this bit is set to 1, access to an endpoint other than Endpoint0 is ignored. This bit is cleared to 0 when Configuration value = 0 is received by the SET_CONFIGURATION request. It is also cleared to 0 when Bus Reset is detected.

(32) UF0 active interface number register (UF0AIFN)

This register sets the valid Interface number that correctly responds to the GET/SET_INTERFACE request. Because Interface 0 is always valid, Interfaces 1 to 4 can be selected.

This register can be read or written in 8-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
UF0AIFN	ADDIF	0	0	0	0	0	IFNO1	IFNO0	F06F0H	00H

Bit position	Bit name	Function
7	ADDIF	This bit allows use of Interfaces numbered other than 0. 1: Support up to the Interface number specified by the IFNO1 and IFNO0 bits. 0: Support only Interface 0 (default value). Setting bits 1 and 0 of this register is invalid when this bit is not set to 1.
1, 0	IFNO1, IFNO0	These bits specify the range of Interface numbers to be supported.

IFNO1	IFNO0	Valid Interface No.
1	1	0, 1, 2, 3, 4
1	0	0, 1, 2, 3
0	1	0, 1, 2
0	0	0, 1

(33) UF0 active alternative setting register (UF0AAS)

This register specifies a link between the Interface number and Alternative Setting.

This register can be read or written in 8-bit units.

USBF of the 78K0R/KC3-L, KE3-L can set a five-series Alternative Setting (Alternate Setting 0, 1, 2, 3, and 4 can be defined) and a two-series Alternative Setting (Alternative Setting 0 and 1 can be defined) for one Interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0AAS	ALT2	IFAL21	IFAL20	ALT2EN	ALT5	IFAL51	IFAL50	ALT5EN	F06F1H	00H

Bit position	Bit name	Function
7, 3	ALTn	These bits specify whether an n-series Alternative Setting is linked with Interface 0. When these bits are set to 1, the setting of the IFALn1 and IFALn0 bits is invalid. 1: Link n-series Alternative Setting with Interface 0. 0: Do not link n-series Alternative Setting with Interface 0 (default value).
6, 5, 2, 1	IFALn1, IFALn0	These bits specify the Interface number to be linked with the n-series Alternative Setting. If the linked Interface number is outside the range specified by the UF0AIFN register, the n-series Alternative Setting is invalid (ALTnEN bit = 0).

IFALn1	IFALn0	Interface number to be linked
1	1	Links Interface 4.
1	0	Links Interface 3.
0	1	Links Interface 2.
0	0	Links Interface 1.

Do not link a five-series Alternative Setting and a two-series Alternative Setting with the same Interface number.

4, 0	ALTnEN	These bits validate the n-series Alternative Setting. Unless these bits are set to 1, the setting of the ALTn, IFALn1, and IFALn0 bits is invalid. 1: Validate the n-series Alternative Setting. 0: Do not validate the n-series Alternative Setting (default value).

Remark n = 2, 5

For example, when the UF0AIFN register is set to 82H and the UF0AAS register is set to 15H, Interfaces 0, 1, 2, and 3 are valid. Interfaces 0 and 2 support only Alternative Setting 0. Interface 1 supports Alternative Setting 0 and 1, and Interface 3 supports Alternative Setting 0, 1, 2, 3, and 4. With this setting, requests GET_INTERFACE wIndex = 0/1/2/3, SET_INTERFACE wValue = 0 & wIndex = 0/2, SET_INTERFACE wValue = 0/1 & wIndex = 1, and SET_INTERFACE wValue = 0/1/2/3/4 & wIndex = 3 are automatically responded to, and a STALL response is made to the other GET/SET_INTERFACE requests.

(34) UF0 alternative setting status register (UF0ASS)

This register indicates the current status of the Alternative Setting.

This register is read-only, in 8-bit units.

Check this register when the SET_INT interrupt request has been issued. The value received by the SET_INTERFACE request is reflected on the UF0IFn register (n = 0- 4) as well as on this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ASS	0	0	0	0	AL5ST3	AL5ST2	AL5ST1	AL2ST	F06F2H	00H

Bit position	Bit name	Function
3- 1	AL5ST3 - AL5ST1	These bits indicate the current status of the five-series Alternative Setting.
0	AL2ST	This bit indicates the current status of the two-series Alternative Setting (selected Alternative Setting number). 1: Alternative Setting 1 0: Alternative Setting 0

AL5ST3	AL5ST2	AL5ST1	Selected Alternative Setting number
1	0	0	Alternative Setting 4
0	1	1	Alternative Setting 3
0	1	0	Alternative Setting 2
0	0	1	Alternative Setting 1
0	0	0	Alternative Setting 0

(35) UF0 endpoint 1 interface mapping register (UF0E1IM)

This register specifies for which Interface and Alternative Setting Endpoint1 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint1 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint1 request and the IN transaction to Endpoint1 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E1IM	E1EN2	E1EN1	E1EN0	E12AL1	E15AL4	E15AL3	E15AL2	E15AL1	F06F3H	00H

Bit position	Bit name	Function
7 - 5	E1EN2 - E1EN0	These bits set a link between the Interface of Endpoint1 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E1EN2	E1EN1	E1EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E12AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint1 is valid.

4	E12AL1	This bit validates Endpoint1 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E15AL4 to E15AL1 bits are 0000.
3 - 0	E15ALn	These bits validate Endpoint1 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1 - 4

(36) UF0 endpoint 2 interface mapping register (UF0E2IM)

This register specifies for which Interface and Alternative Setting Endpoint2 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint2 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint2 request and the OUT transaction to Endpoint2 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E2IM	E2EN2	E2EN1	E2EN0	E22AL1	E25AL4	E25AL3	E25AL2	E25AL1	F06F4H	00H

Bit position	Bit name	Function
7- 5	E2EN2 to E2EN0	These bits set a link between the Interface of Endpoint2 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E2EN2	E2EN1	E2EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E22AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint2 is valid.

4	E22AL1	This bit validates Endpoint2 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E25AL4 to E25AL1 bits are 0000.
3 to 0	E25ALn	These bits validate Endpoint2 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1 - 4

(37) UF0 endpoint 3 interface mapping register (UF0E3IM)

This register specifies for which Interface and Alternative Setting Endpoint3 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint3 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint3 request and the IN transaction to Endpoint3 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E3IM	E3EN2	E3EN1	E3EN0	E32AL1	E35AL4	E35AL3	E35AL2	E35AL1	F06F5H	00H

Bit position	Bit name	Function
7 - 5	E3EN2 - E3EN0	These bits set a link between the Interface of Endpoint3 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E3EN2	E3EN1	E3EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E32AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint3 is valid.

4	E32AL1	This bit validates Endpoint3 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E35AL4 to E35AL1 bits are 0000.
3 - 0	E35ALn	These bits validate Endpoint3 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1- 4

(38) UF0 endpoint 4 interface mapping register (UF0E4IM)

This register specifies for which Interface and Alternative Setting Endpoint4 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint4 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint4 request and the OUT transaction to Endpoint4 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E4IM	E4EN2	E4EN1	E4EN0	E42AL1	E45AL4	E45AL3	E45AL2	E45AL1	F06F6H	00H

Bit position	Bit name	Function
7 - 5	E4EN2 - E4EN0	These bits set a link between the Interface of Endpoint4 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E4EN2	E4EN1	E4EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E42AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint4 is valid.

4	E42AL1	This bit validates Endpoint4 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E45AL4 to E45AL1 bits are 0000.
3 - 0	E45ALn	These bits validate Endpoint4 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1 - 4

(39) UF0 endpoint 7 interface mapping register (UF0E7IM)

This register specifies for which Interface and Alternative Setting Endpoint7 is valid.

This register can be read or written in 8-bit units.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint7 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint7 request and the IN transaction to Endpoint7 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E7IM	E7EN2	E7EN1	E7EN0	E72AL1	E75AL4	E75AL3	E75AL2	E75AL1	F06F9H	00H

Bit position	Bit name	Function
7 - 5	E7EN2 - E7EN0	These bits set a link between the Interface of Endpoint7 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E7EN2	E7EN1	E7EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E72AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint7 is valid.

4	E72AL1	This bit validates Endpoint7 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E75AL4 to E75AL1 bits are 0000.
3 - 0	E75ALn	These bits validate Endpoint7 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1 - 4

(40) UF0 endpoint 8 interface mapping register (UF0E8IM)

This register specifies for which Interface and Alternative Setting Endpoint8 is valid.
This register can be read or written in 8-bit units.
The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint8 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint8 request and the IN transaction to Endpoint8 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	Initial value
UF0E8IM	E8EN2	E8EN1	E8EN0	E82AL1	E85AL4	E85AL3	E85AL2	E85AL1	F06FAH	00H

Bit position	Bit name	Function
7 - 5	E8EN2 - E8EN0	These bits set a link between the Interface of Endpoint8 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E8EN2	E8EN1	E8EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternative Setting 0
1	0	0	Linked with Interface 3 and Alternative Setting 0
0	1	1	Linked with Interface 2 and Alternative Setting 0
0	1	0	Linked with Interface 1 and Alternative Setting 0
0	0	1	Linked with Interface 0 and Alternative Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E82AL1 bit is cleared to 0.
If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint8 is valid.

4	E82AL1	This bit validates Endpoint8 when the two-series Alternative Setting and the Alternative Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E85AL4 - E85AL1 bits are 0000.
3 - 0	E85ALn	These bits validate Endpoint8 when the five-series Alternative Setting and the Alternative Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1 (default value).

Remark n = 1- 4

13.6.4 Data hold registers

(1) UF0 EP0 read register (UF0E0R)

The UF0E0R register is a 64-byte FIFO that stores the OUT data sent from the host in the data stage of control transfer to/from Endpoint0.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The hardware automatically transfers data to the UF0E0R register when it has received the data from the host. When the data has been correctly received, the E0ODT bit of the UF0IS1 register is set to 1. The UF0E0L register holds the quantity of the received data, and an interrupt request (INTUSB) is issued. The UF0E0L register always updates the length of the received data while it is receiving data. If the final transfer is correct reception, the interrupt request is generated. If the reception is abnormal, the UF0E0L register is cleared to 0 and the interrupt request is not generated.

The data held by the UF0E0R register must be read by FW up to the value of the amount of data read by the UF0E0L register. Check that all data has been read by using the EP0R bit of the UF0EPS0 register (EP0R bit = 0 when all data has been read). If the value of the UF0E0L register is 0, the EP0NKR bit of the UF0E0N register is cleared to 0, and the UF0E0R register is ready for reception. The UF0E0R register is cleared when the next SETUP token has been received.

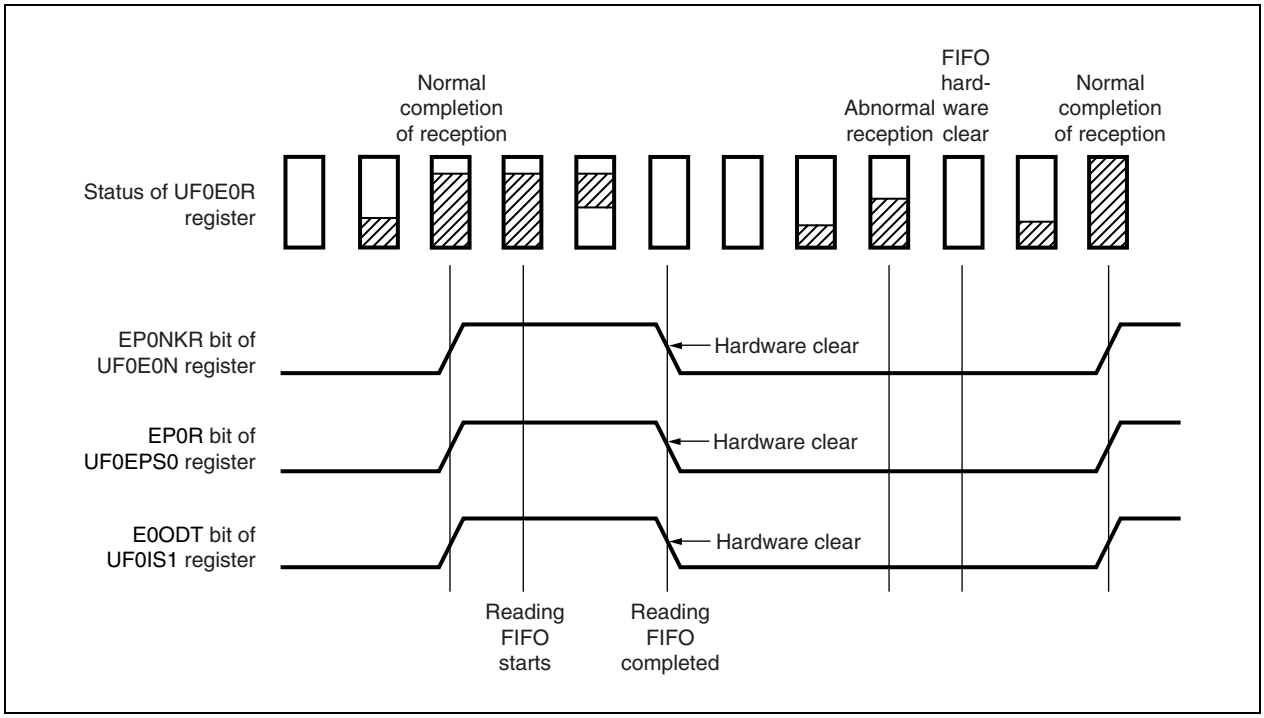
Caution Read all the data stored. Clear the FIFO to discard some data.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0R	E0R7	E0R6	E0R5	E0R4	E0R3	E0R2	E0R1	E0R0	F0580H	Undefined

Bit position	Bit name	Function
7 - 0	E0R7 - E0R0	These bits store the OUT data sent from the host in the data stage of control transfer to/from Endpoint0.

The operation of the UF0E0R register is illustrated below.

Figure 13-4. Operation of UF0E0R Register



(2) UF0 EP0 length register (UF0E0L)

The UF0E0L register stores the data length held by the UF0E0R register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0E0L register always updates the length of the received data while it is receiving data. If the final transfer is abnormal reception, the UF0E0L register is cleared to 0 and the interrupt request is not generated. The interrupt request is generated only when the reception is normal, and the FW can read as many data from the UF0E0R register as the value read from the UF0E0L register. The value of the UF0E0L register is decremented each time the UF0E0R register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0L	E0L7	E0L6	E0L5	E0L4	E0L3	E0L2	E0L1	E0L0	F0581H	00H

Bit position	Bit name	Function
7 - 0	E0L7 - E0L0	These bits store the data length held by the UF0E0R register.

(3) UF0 EP0 setup register (UF0E0ST)

The UF0E0ST register holds the SETUP data sent from the host.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0E0ST register always writes data when a SETUP transaction has been received. The hardware sets the PROT bit of the UF0IS1 register when it has correctly received the SETUP transaction. It sets the CPUDEC bit of the UF0IS1 register in the case of an FW-processed request. Then an interrupt request (INTUSB) is issued. In the case of an FW-processed request, be sure to read the request in 8-byte units. If it is not read in 8-byte units, the subsequent requests cannot be correctly decoded. The read counter of the UF0E0ST register is not cleared even when Bus Reset is received. Always read this counter in 8-byte units regardless of whether Bus Reset is received or not.

Because the UF0E0ST register always enables writing, the hardware overwrites data to this register even if a SETUP transaction is received while the data of the register is being read. Even if the SETUP transaction cannot be correctly received, the CPUDEC interrupt request and Protect interrupt request are not generated, but the previous data is discarded. If a SETUP token of less than 8 bytes is received, however, the received SETUP token is discarded, and the previously received SETUP data is retained. If the SETUP token is received more than once when control transfer is executed once, be sure to check the PROT bit of the UF0IS1 register under the conditions below. If PROT bit = 1, read the UF0E0ST register again because the SETUP transaction has been received more than once.

<1> If a request is decoded by FW and the UF0E0R register is read or the UF0E0W register is written

<2> When preparing for a STALL response for the request to which the decode result does not correspond

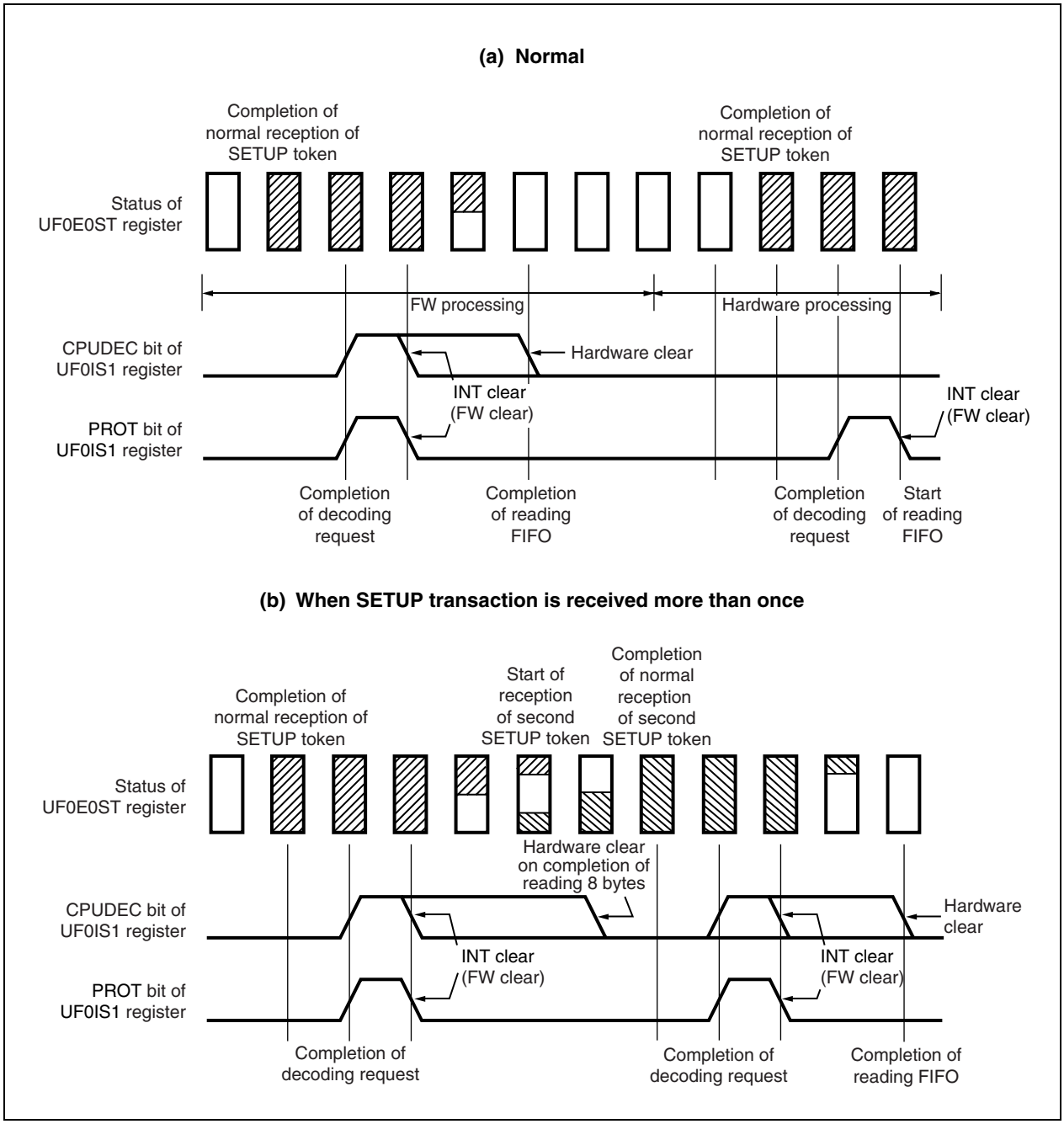
Caution Be sure to read all the stored data. The UF0E0ST register is always updated by the request in the SETUP transaction.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0ST	E0S7	E0S6	E0S5	E0S4	E0S3	E0S2	E0S1	E0S0	F0582H	00H

Bit position	Bit name	Function
7 - 0	E0S7 - E0S0	These bits hold the SETUP data sent from the host.

The operation of the UF0E0ST register is illustrated below.

Figure 13-5. Operation of UF0E0ST Register



(4) UF0 EP0 write register (UF0E0W)

The UF0E0W register is a 64-byte FIFO that stores the IN data (passes it to SIE) sent to the host in the data stage to Endpoint0.

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with an IN token only when the EP0NKW bit of the UF0E0N register is set to 1 (when NAK is not transmitted). When data is transmitted and when the host correctly receives the data, the EP0NKW bit of the UF0E0N register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the UF0E0W register and the E0DED bit of the UF0DEND register is set to 1 (EP0W bit of the UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0E0W register is cleared and the E0DED bit of the UF0DEND register is set to 1 (EP0W bit of the UF0EPS0 register = 1 (data exists)).

The UF0E0W register is cleared to 0 when the next SETUP token is received while transmission has not been completed yet. If the stage of control transfer (read) changes to the status stage while ACK has not been correctly received in the data stage, the UF0E0W register is automatically cleared to 0. At the same time, it is also cleared to 0 if the EP0NKW bit of the UF0E0N register is 1.

If the UF0E0W register is read while no data is in it, 00H is read.

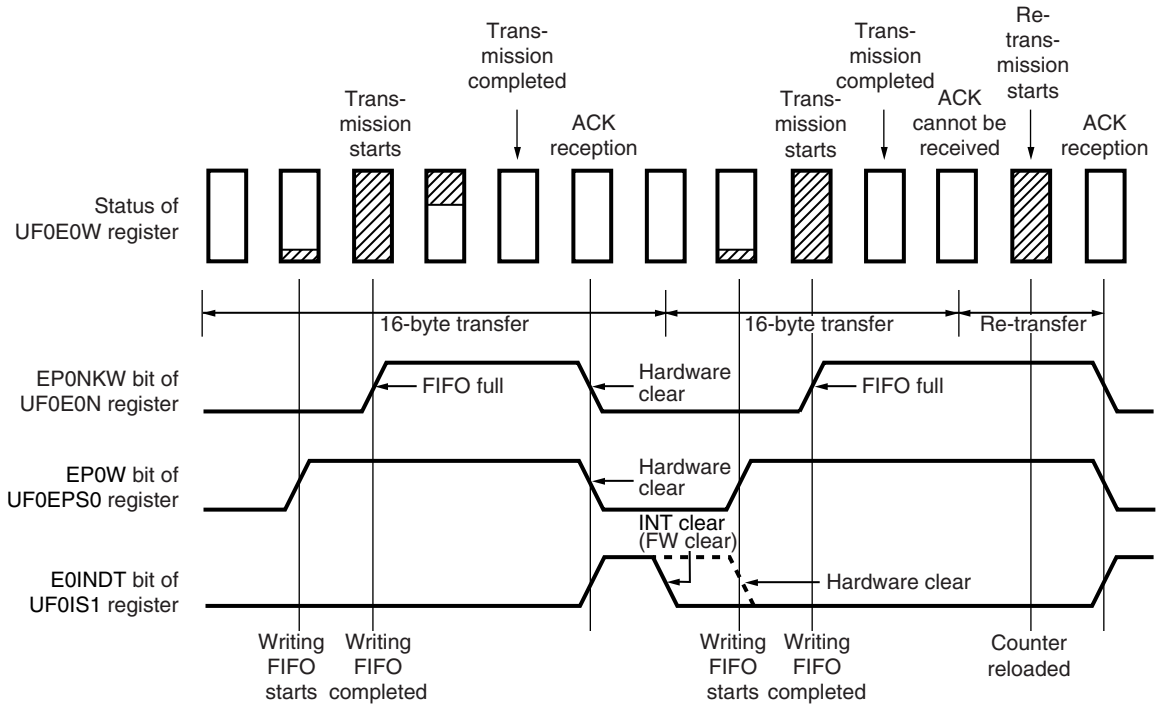
	7	6	5	4	3	2	1	0	Address	After reset
UF0E0W	E0W7	E0W6	E0W5	E0W4	E0W3	E0W2	E0W1	E0W0	F0583H	Undefined

Bit position	Bit name	Function
7 - 0	E0W7 - E0W0	These bits store the IN data sent to the host in the data stage to Endpoint0.

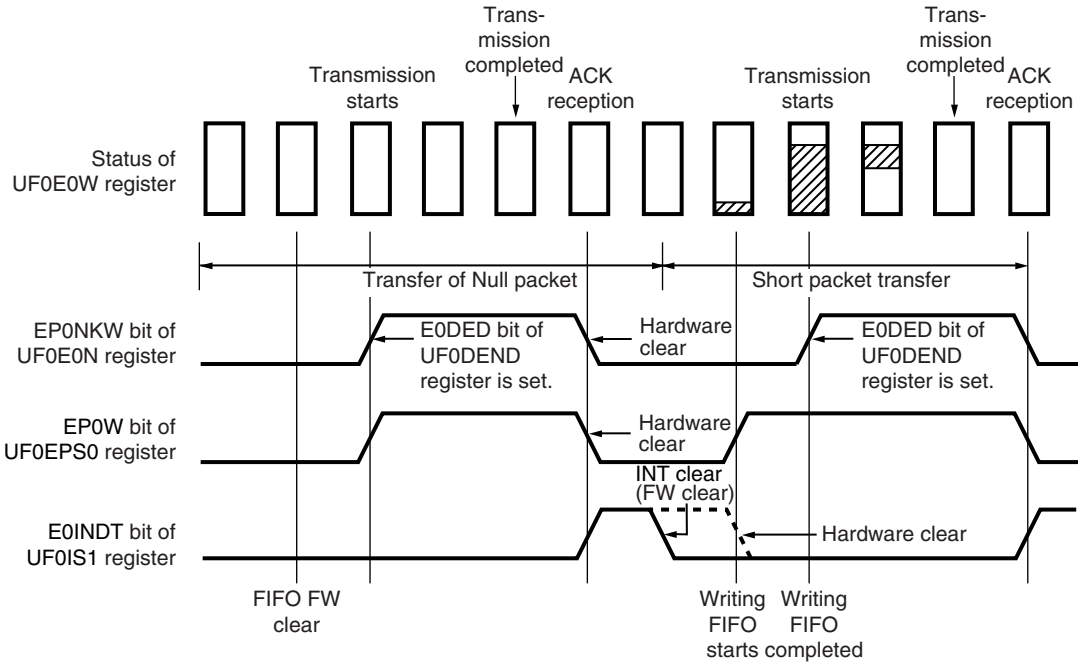
The operation of the UF0E0W register is illustrated below.

Figure 13-6. Operation of UF0E0W Register

(a) 16-byte transmission



(b) When Null packet or short packet is transmitted



(5) UF0 bulk-out 1 register (UF0BO1)

The UF0BO1 register is a 64-byte × 2 FIFO that stores data for Endpoint2. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when data is in the FIFO on the SIE side and when no data is in the FIFO on the CPU side (counter value = 0).

This register is read-only, in 8-bit units. A write access to this register is ignored.

When the hardware receives data for Endpoint2 from the host, it automatically transfers the data to the UF0BO1 register. When the register correctly receives the data, a FIFO toggle operation occurs. As a result, the BKO1DT bit of the UF0IS3 register is set to 1, the quantity of the received data is held by the UF0BO1L register, and an interrupt request is issued to the CPU.

Read the data held by the UF0BO1 register by FW, up to the value of the amount of data read by the UF0BO1L register. When the correct received data is held by the FIFO connected to the SIE side and the value of the UF0BO1L register reaches 0, the toggle operation of the FIFO occurs, and the BKO1NK bit of the UF0EN register is automatically cleared to 0. If data greater than the value of the UF0BO1L register is read and if the FIFO toggle condition is satisfied, the toggle operation of the FIFO occurs. As a result, the next packet may be read by mistake. Note that, if the toggle condition is not satisfied, the first data is repeatedly read.

If overrun data is received while data is held by the FIFO connected to the CPU side, Endpoint2 stalls, and the FIFO on the CPU side is cleared.

When the UF0BO1 register is read while no data is in it, an undefined value is read.

Caution Be sure to read all the data stored in this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO1	BKO17	BKO16	BKO15	BKO14	BKO13	BKO12	BKO11	BKO10	F0584H	Undefined

Bit position	Bit name	Function
7 - 0	BKO17 - BKO10	These bits store data for Endpoint2.

The operation of the UF0BO1 register is illustrated below.

Figure 13-7. Operation of UF0BO1 Register (1/2)

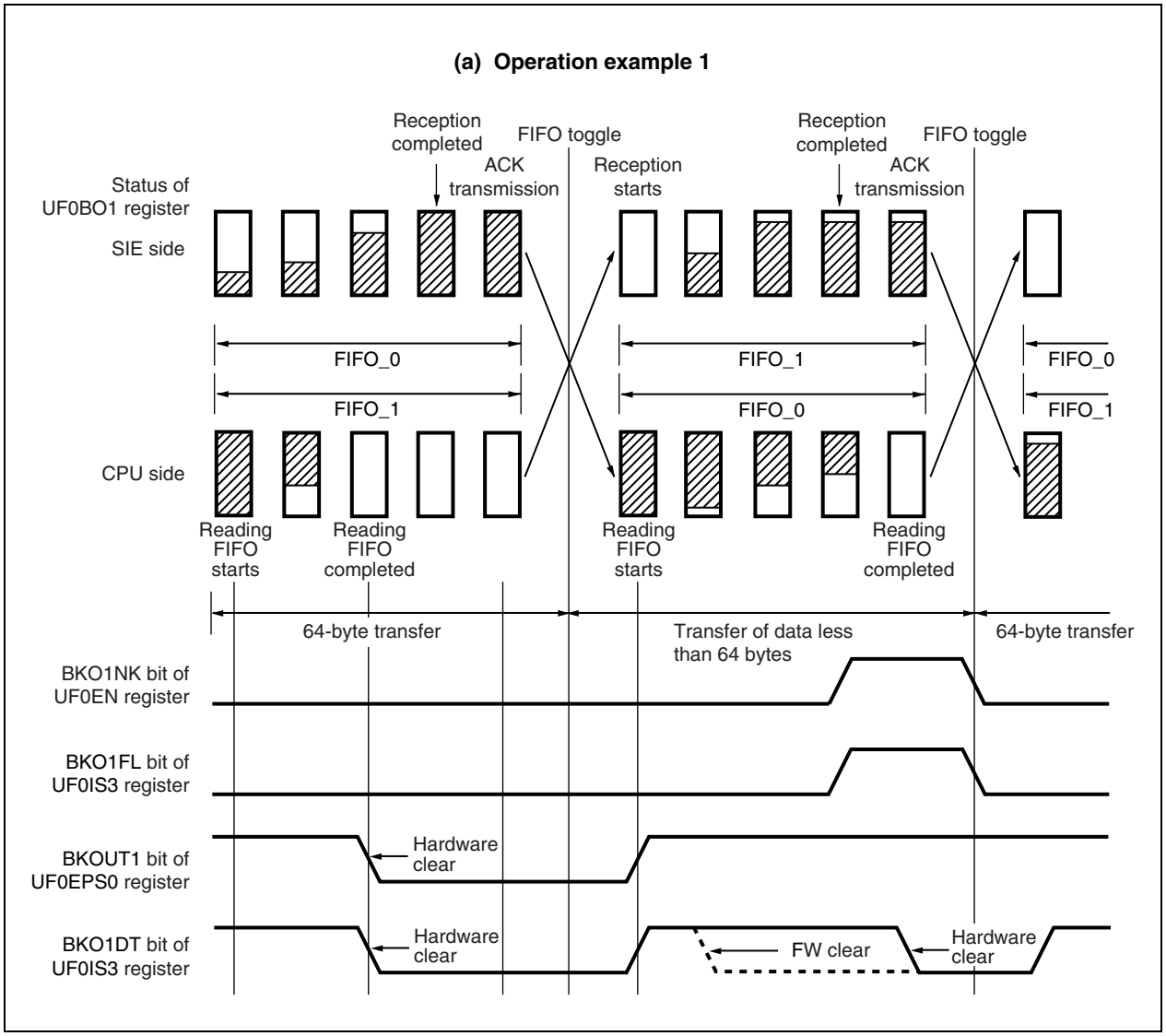
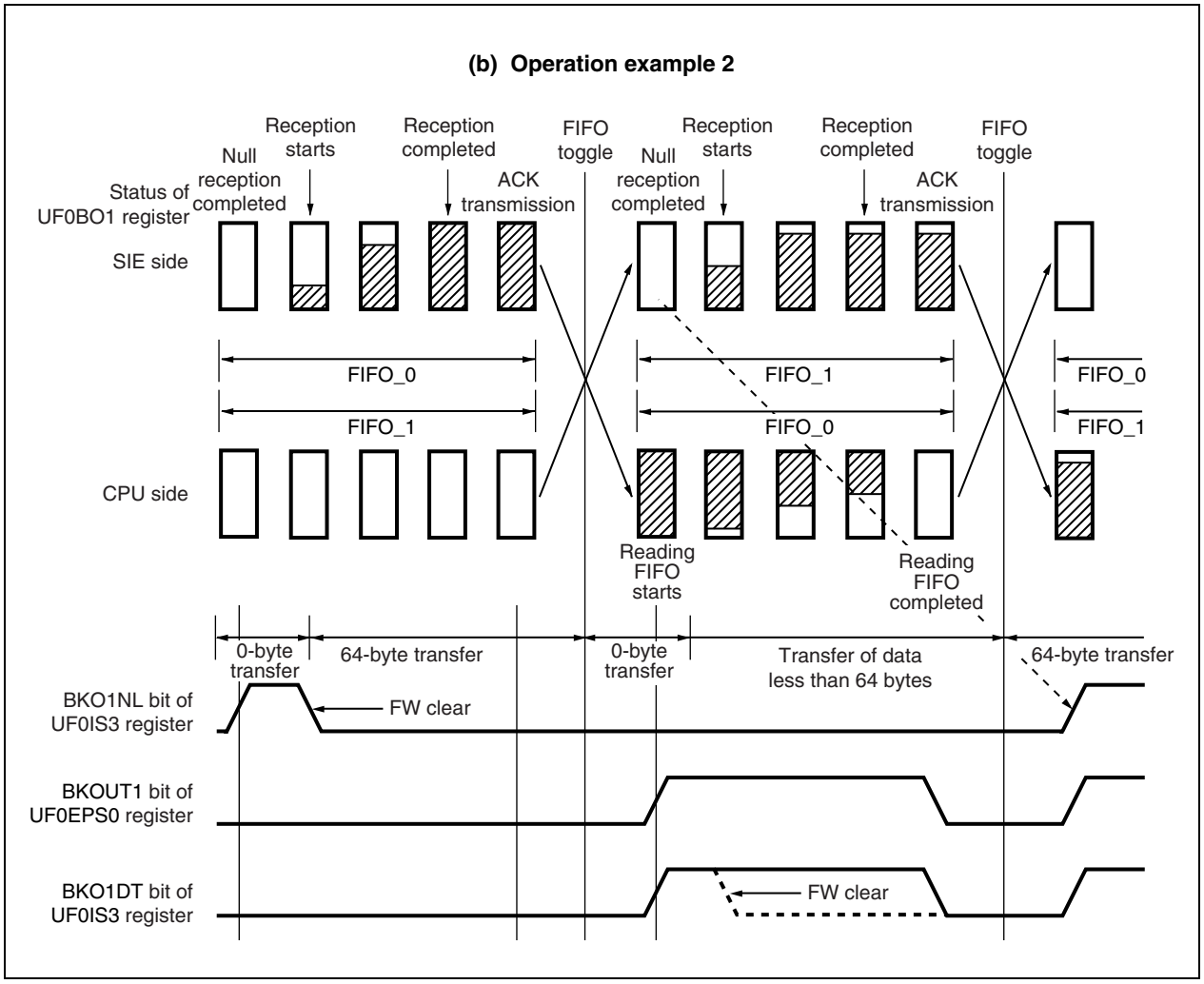


Figure 13-7. Operation of UF0BO1 Register (2/2)



(6) UF0 bulk-out 1 length register (UF0BO1L)

The UF0BO1L register stores the length of the data held by the UF0BO1 register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0BO1L register always updates the received data length while it is receiving data. If the final transfer is abnormal reception, the UF0BO1L register is cleared to 00H, and an interrupt request is not generated. Only if the reception is normal, the interrupt request is generated, and FW can read as much data from the UF0BO1 register as the value read from the UF0BO1L register. The value of the UF0BO1L register is decremented each time the UF0BO1 register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO1L	BKO1L7	BKO1L6	BKO1L5	BKO1L4	BKO1L3	BKO1L2	BKO1L1	BKO1L0	F0585H	00H

Bit position	Bit name	Function
7 - 0	BKO1L7 - BKO1L0	These bits store the length of the data held by the UF0BO1 register.

(7) UF0 bulk-out 2 register (UF0BO2)

The UF0BO2 register is a 64-byte × 2 FIFO that stores data for Endpoint4. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when data is in the FIFO on the SIE side and when no data is in the FIFO on the CPU side (counter value = 0).

This register is read-only, in 8-bit units. A write access to this register is ignored.

When the hardware receives data for Endpoint4 from the host, it automatically transfers the data to the UF0BO2 register. When the register correctly receives the data, a FIFO toggle operation occurs. As a result, the BKO2DT bit of the UF0IS3 register is set to 1, the quantity of the received data is held by the UF0BO2L register, and an interrupt request is issued to the CPU.

Read the data held by the UF0BO2 register by FW, up to the value of the amount of data read by the UF0BO2L register. When the correct received data is held by the FIFO connected to the SIE side and the value of the UF0BO2L register reaches 0, the toggle operation of the FIFO occurs, and the BKO2NK bit of the UF0EN register is automatically cleared to 0. If data greater than the value of the UF0BO2L register is read and if the FIFO toggle condition is satisfied, the toggle operation of the FIFO occurs. As a result, the next packet may be read by mistake. Note that, if the toggle condition is not satisfied, the first data is repeatedly read.

If overrun data is received while data is held by the FIFO connected to the CPU side, Endpoint4 stalls, and the FIFO on the CPU side is cleared.

When the UF0BO2 register is read while no data is in it, an undefined value is read.

Caution Be sure to read all the data stored in this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO2	BKO27	BKO26	BKO25	BKO24	BKO23	BKO22	BKO21	BKO20	F0586H	Undefined

Bit position	Bit name	Function
7 - 0	BKO27 - BKO20	These bits store data for Endpoint4.

The operation of the UF0BO2 register is illustrated below.

Figure 13-8. Operation of UF0BO2 Register (1/2)

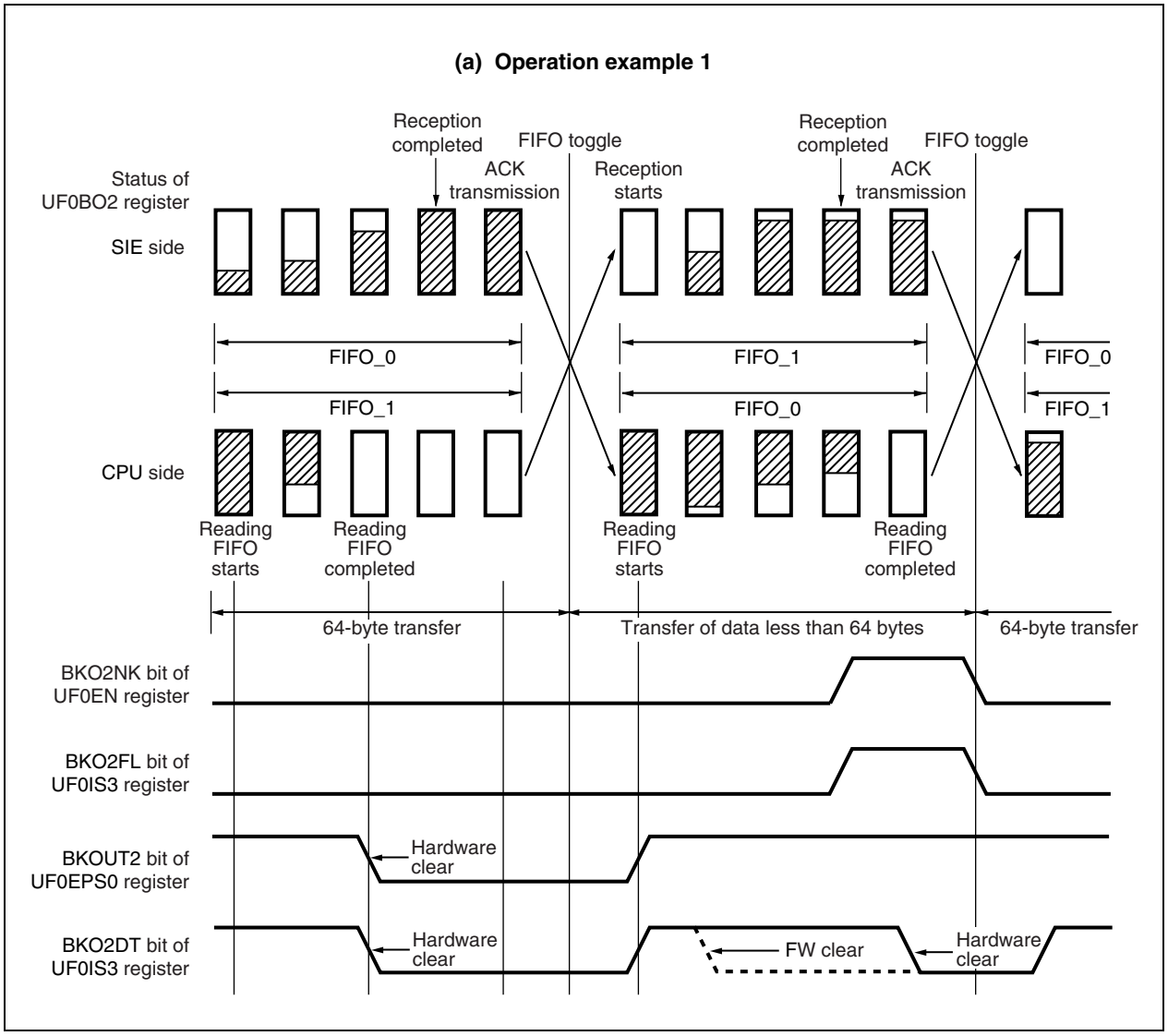
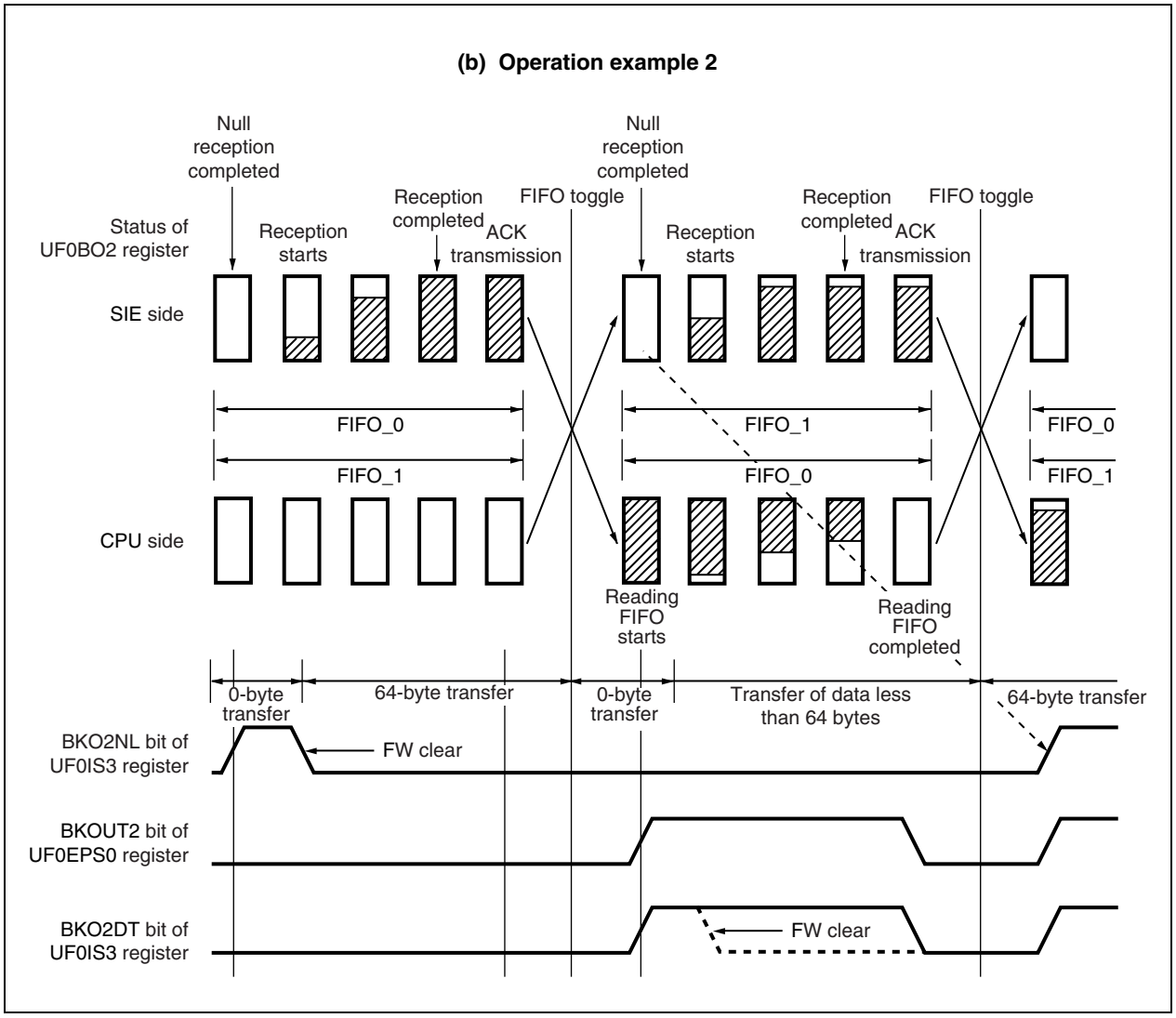


Figure 13-8. Operation of UF0BO2 Register (2/2)



(8) UF0 bulk-out 2 length register (UF0BO2L)

The UF0BO2L register stores the length of the data held by the UF0BO2 register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0BO2L register always updates the received data length while it is receiving data. If the final transfer is abnormal reception, the UF0BO2L register is cleared to 00H, and an interrupt request is not generated. Only if the reception is normal, the interrupt request is generated, and FW can read as much data from the UF0BO2 register as the value read from the UF0BO2L register. The value of the UF0BO2L register is decremented each time the UF0BO2 register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO2L	BKO2L7	BKO2L6	BKO2L5	BKO2L4	BKO2L3	BKO2L2	BKO2L1	BKO2L0	F0587H	00H

Bit position	Bit name	Function
7 - 0	BKO2L7 - BKO2L0	These bits store the length of the data held by the UF0BO2 register.

(9) UF0 bulk-in 1 register (UF0BI1)

The UF0BI1 register is a 64-byte × 2 FIFO that stores data for Endpoint1. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when no data is in the FIFO on the SIE side (counter value = 0) and when the FIFO on the CPU side is correctly written (FIFO full or BKI1DED bit = 1).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint1 only when the BKI1NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, FW can transmit data to the host only by writing the data to the UF0BI1 register sequentially. A short packet is transmitted when data is written to the UF0BI1 register and the BKI1DED bit of the UF0DEND register is set to 1 (BKIN1 bit of UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0BI1 register is cleared and the BKI1DED bit of the UF0DEND register is set to 1 (BKIN1 bit of the UF0EPS0 register = 1 (data exists)). When the data is transmitted correctly, a FIFO toggle operation occurs. The BKI1DT bit of the UF0IS2 register is set to 1, and an interrupt request is generated for the CPU.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BI1	BKI17	BKI16	BKI15	BKI14	BKI13	BKI12	BKI11	BKI10	F0588H	Undefined

Bit position	Bit name	Function
7 - 0	BKI17 - BKI10	These bits store data for Endpoint1.

The operation of the UF0BI1 register is illustrated below.

Figure 13-9. Operation of UF0BI1 Register (1/3)

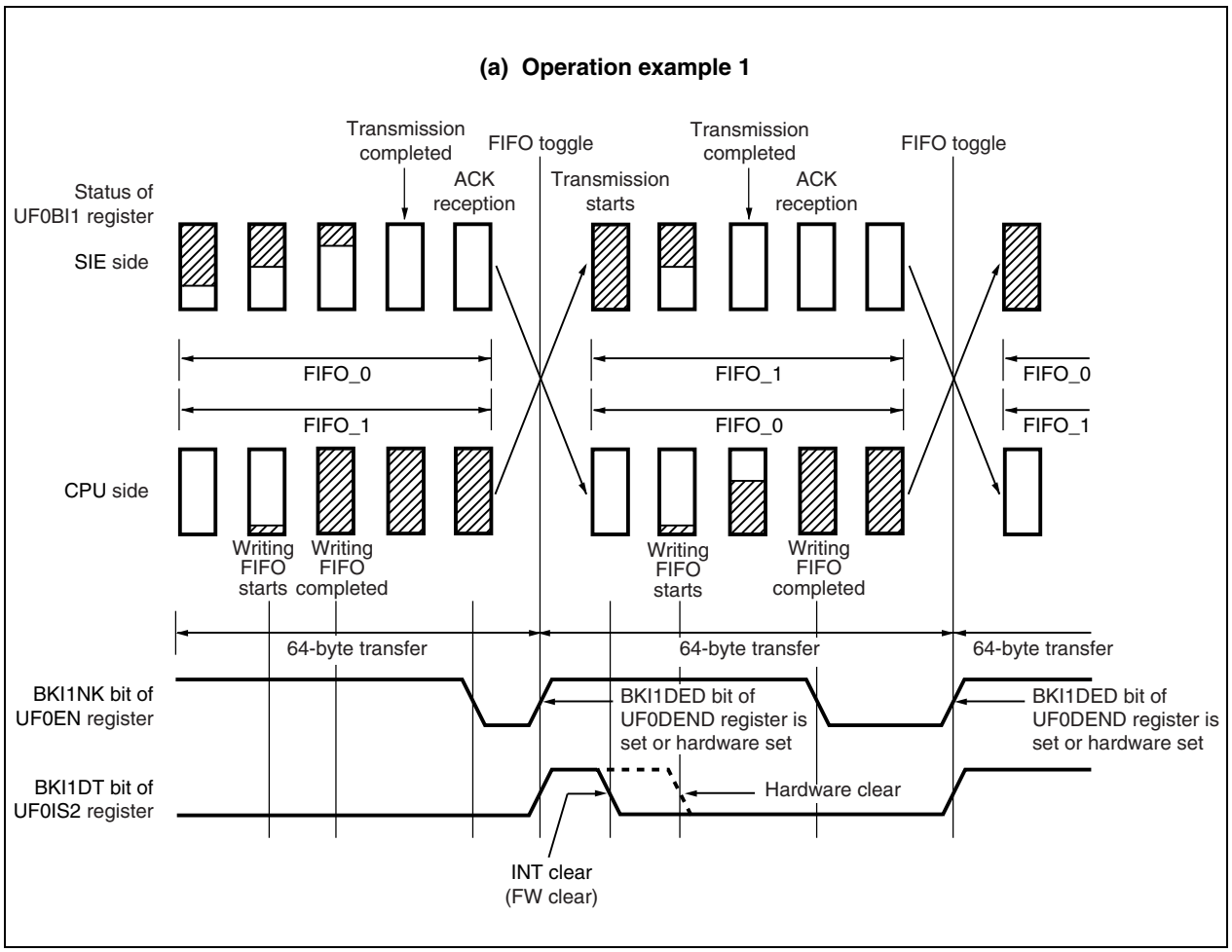


Figure 13-9. Operation of UF0BI1 Register (2/3)

(b) Operation example 2

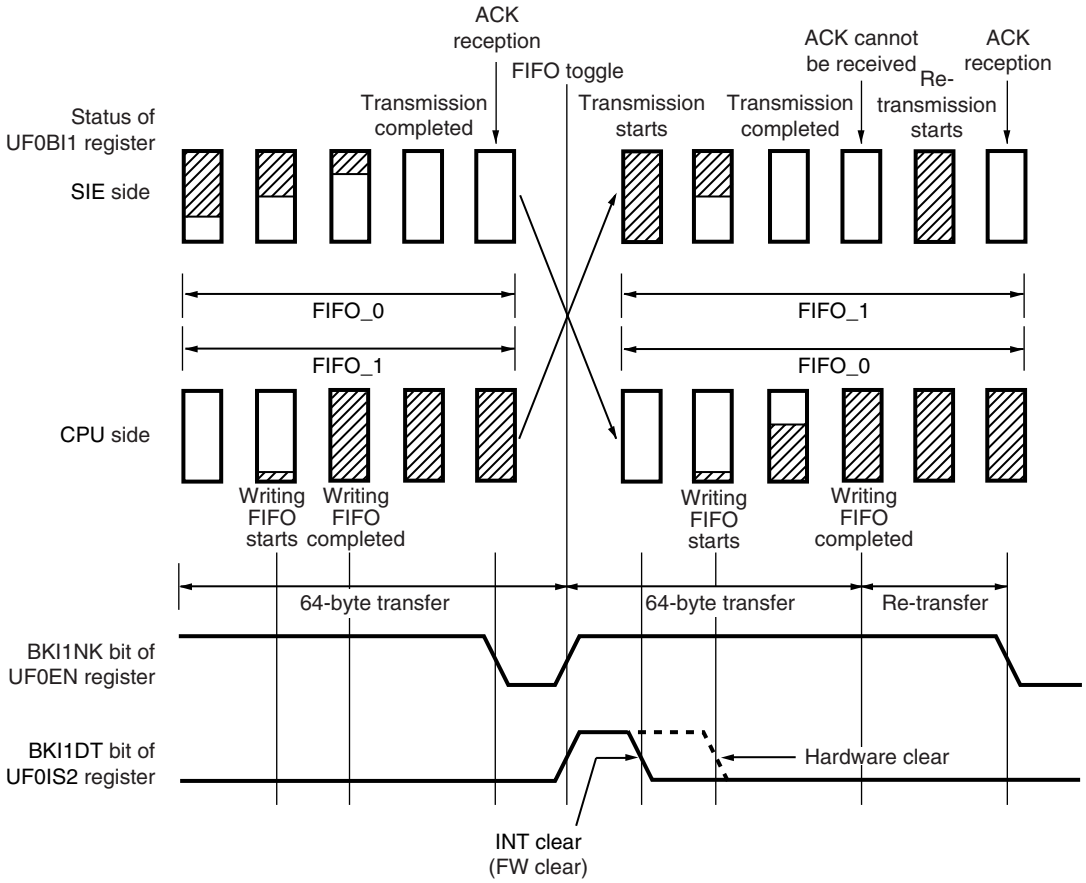
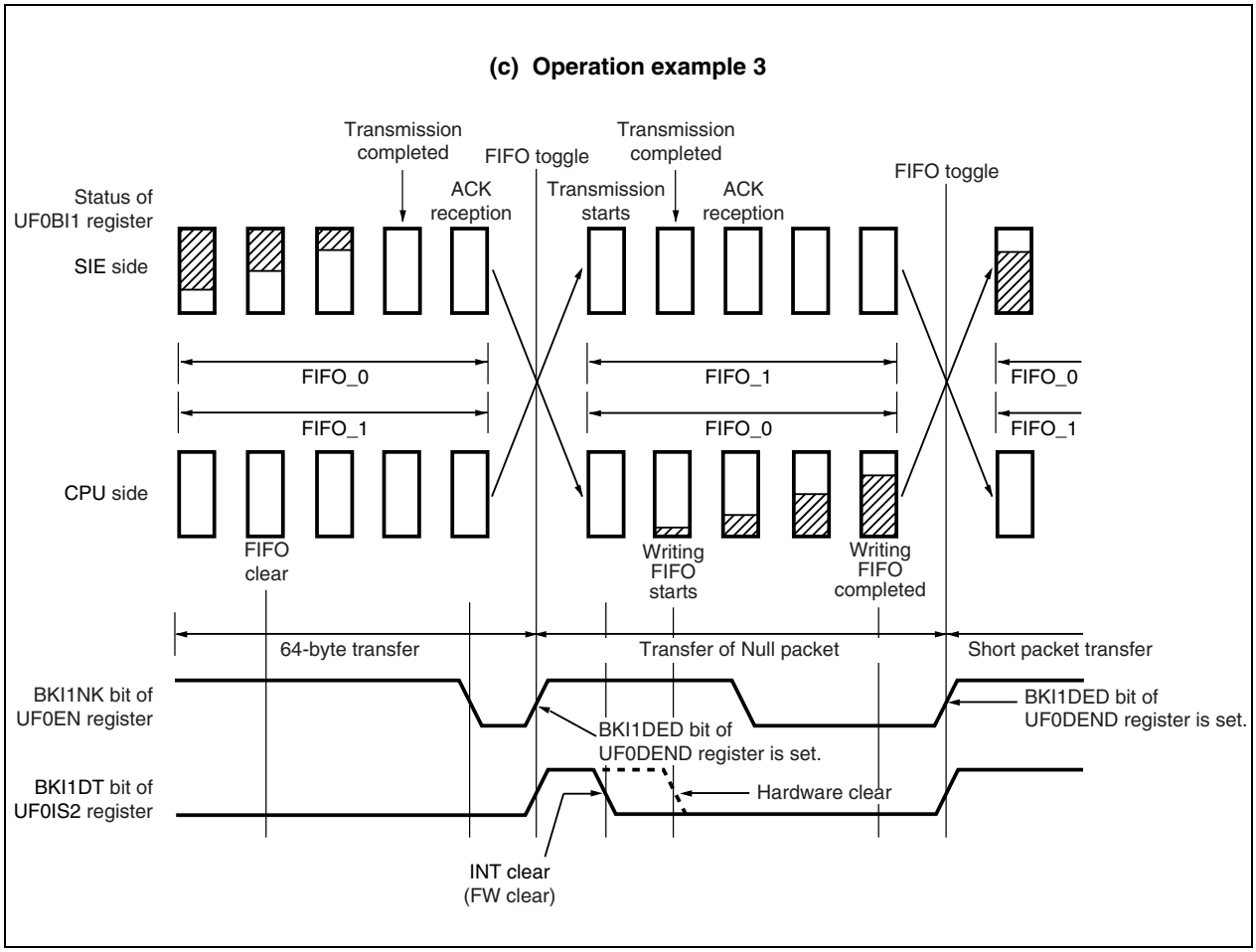


Figure 13-9. Operation of UF0BI1 Register (3/3)



(10) UF0 bulk-in 2 register (UF0BI2)

The UF0BI2 register is a 64-byte × 2 FIFO that stores data for Endpoint3. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when no data is in the FIFO on the SIE side (counter value = 0) and when the FIFO on the CPU side is correctly written (FIFO full or BKI2DED bit = 1).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint3 only when the BKI2NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, FW can transmit data to the host only by writing the data to the UF0BI2 register sequentially. A short packet is transmitted when data is written to the UF0BI2 register and the BKI2DED bit of the UF0DEND register is set to 1 (BKIN2 bit of UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0BI2 register is cleared and the BKI2DED bit of the UF0DEND register is set to 1 (BKIN2 bit of the UF0EPS0 register = 1 (data exists)). When the data is transmitted correctly, a FIFO toggle operation occurs. The BKI2DT bit of the UF0IS2 register is set to 1, and an interrupt request is generated for the CPU.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BI2	BKI27	BKI26	BKI25	BKI24	BKI23	BKI22	BKI21	BKI20	F0589H	Undefined

Bit position	Bit name	Function
7 - 0	BKI27 - BKI20	These bits store data for Endpoint3.

The operation of the UF0BI2 register is illustrated below.

Figure 13-10. Operation of UF0BI2 Register (1/3)

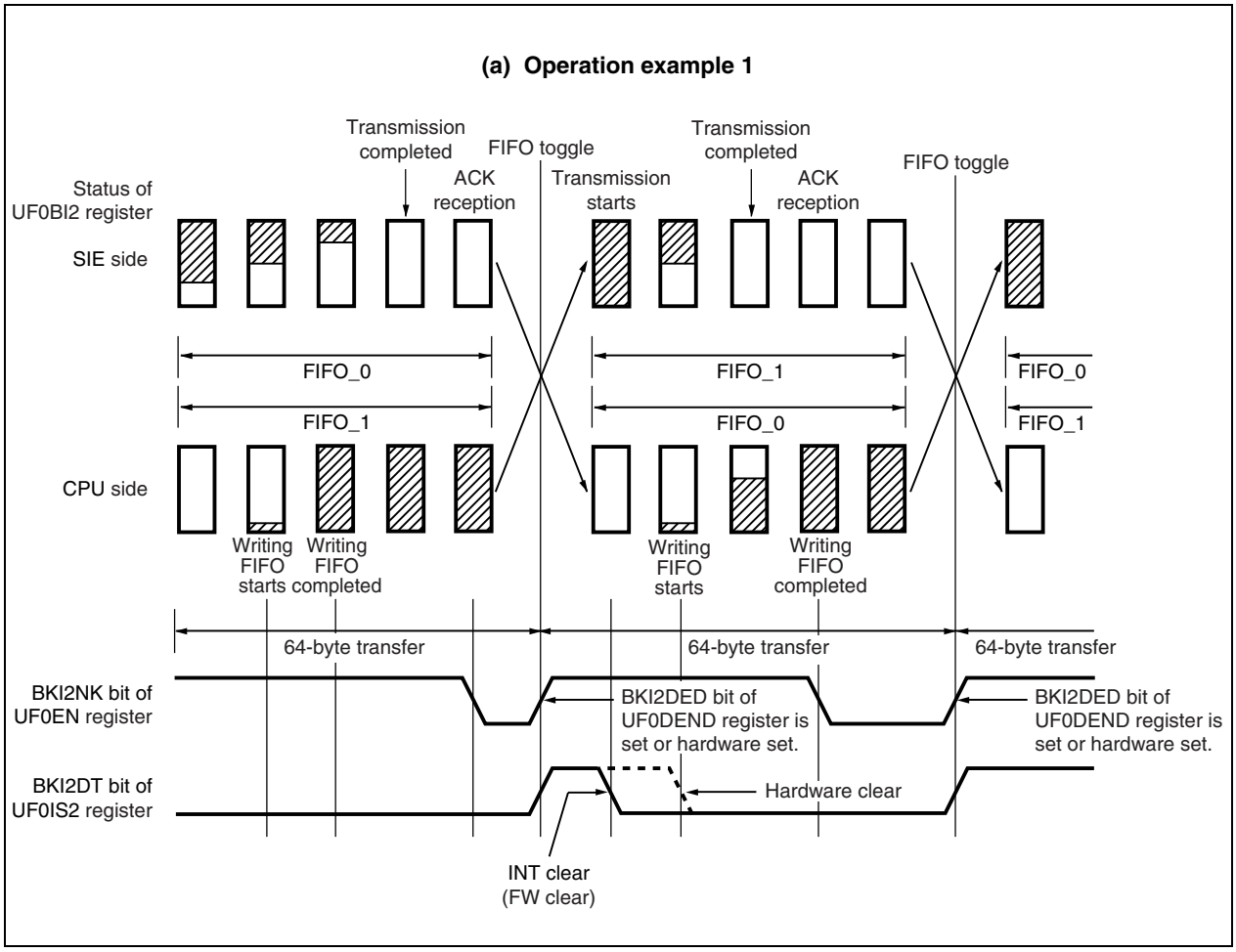


Figure 13-10. Operation of UF0BI2 Register (2/3)

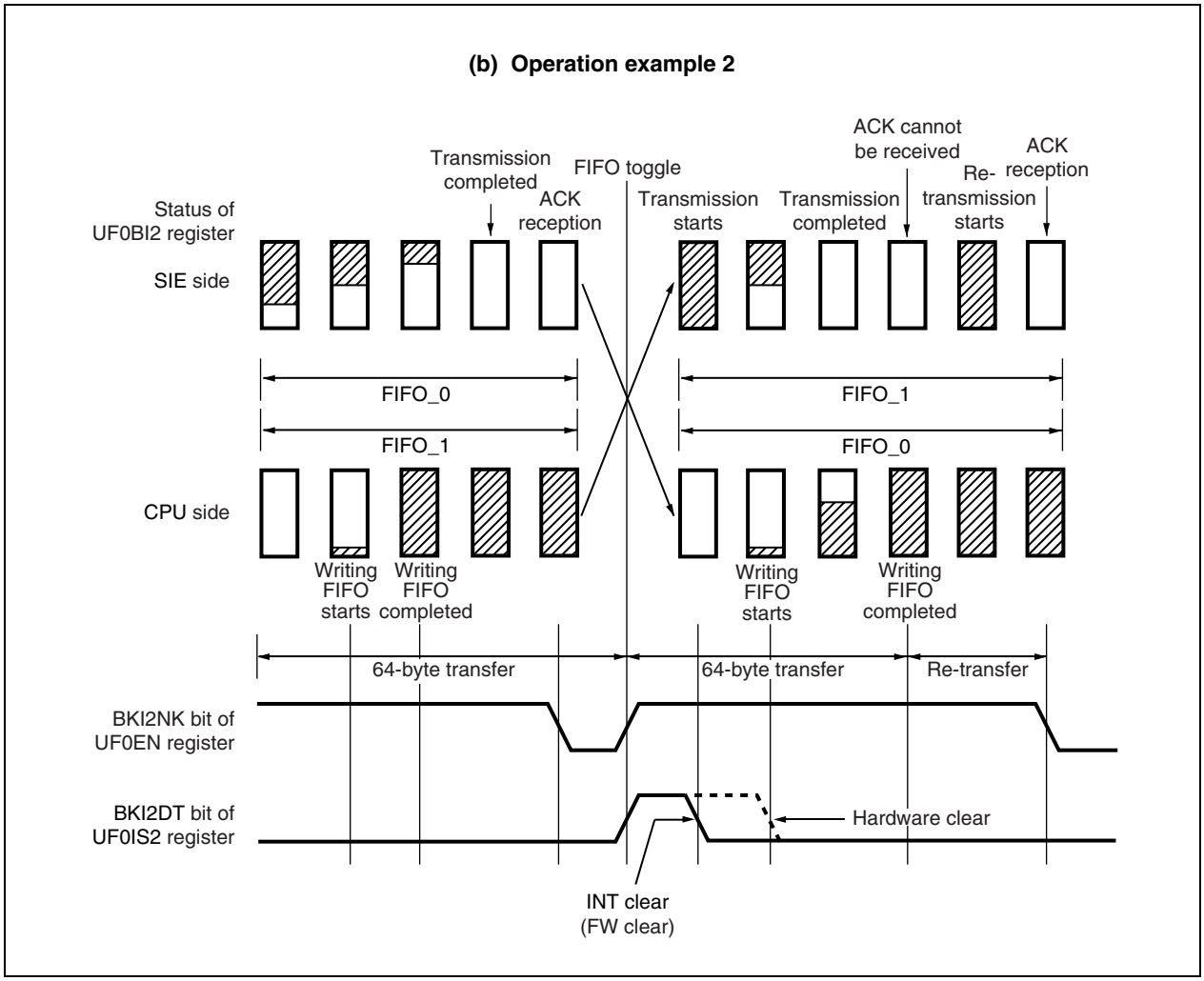
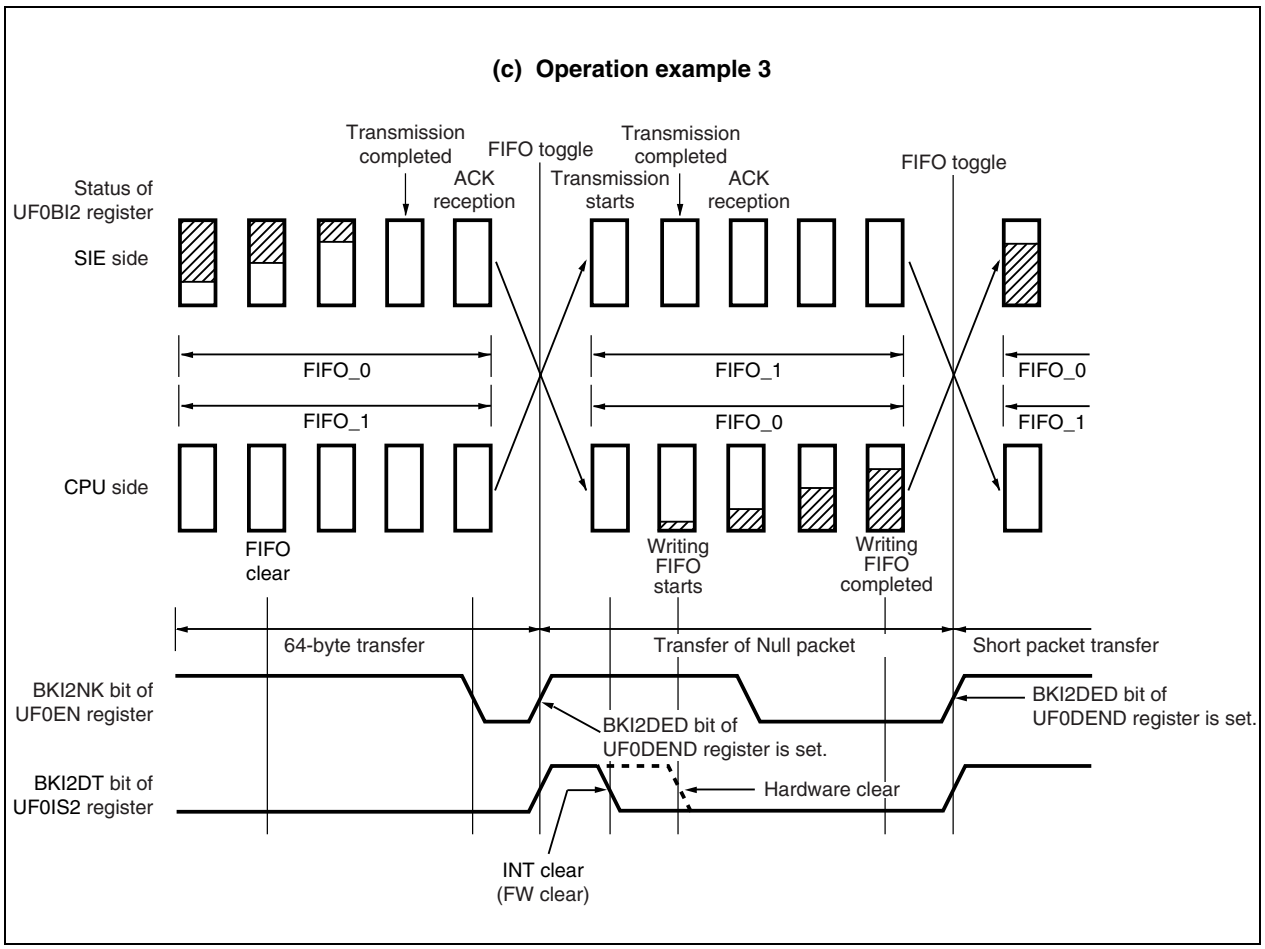


Figure 13-10. Operation of UF0BI2 Register (3/3)



(11) UF0 interrupt 1 register (UF0INT1)

The UF0INT1 register is a 64-byte FIFO that stores data for Endpoint7 (to be passed to SIE).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

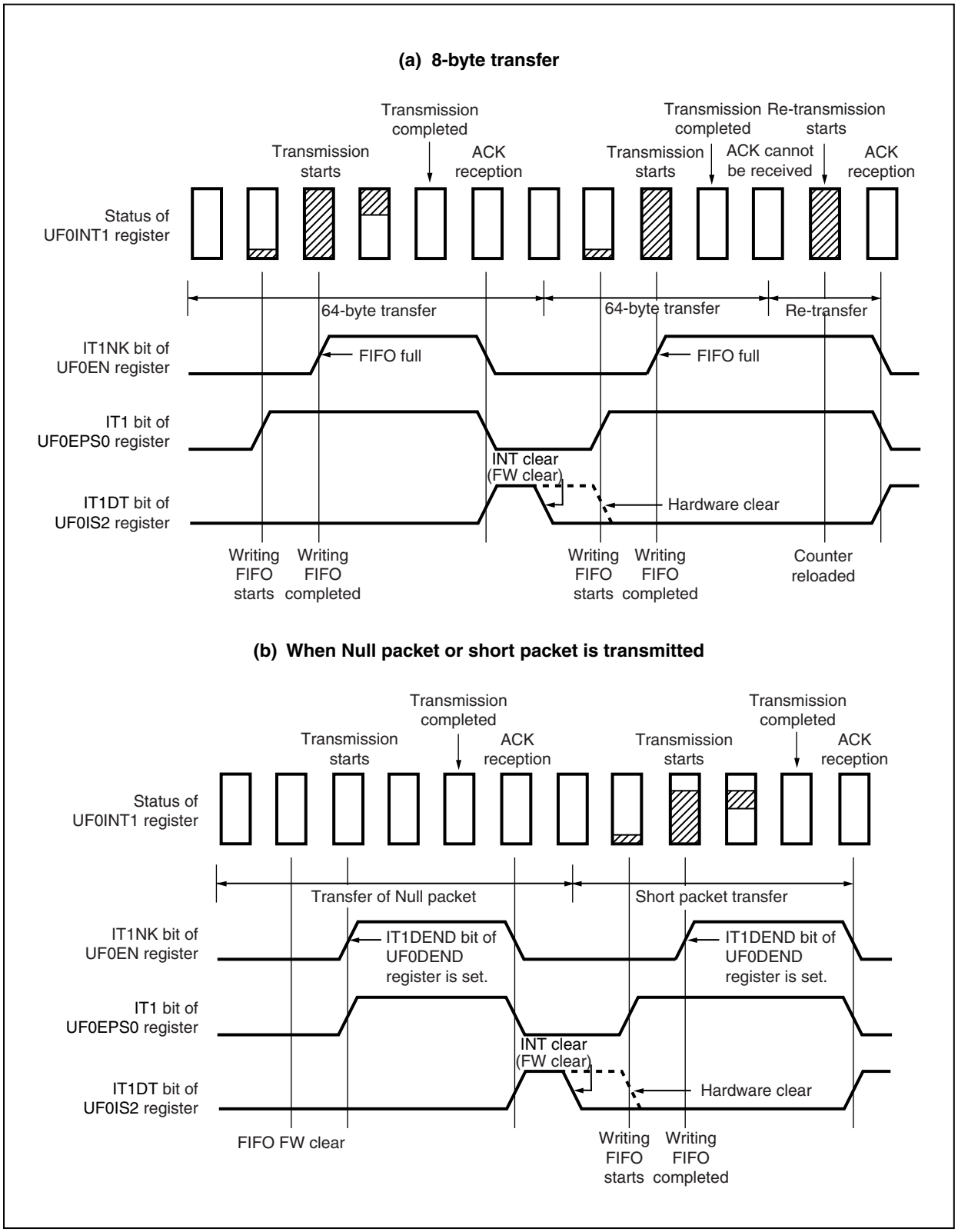
The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint7 only when the IT1NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). When the data is transmitted and the host correctly receives it, the IT1NK bit of the UF0EN register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the UF0INT1 register and the IT1DEND bit of the UF0DEND register is set to 1 (IT1 bit of the UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0INT1 register is cleared and the IT1DEND bit of the UF0DEND register is set to 1 (IT1 bit of the UF0EPS0 register = 1 (data exists)).

	7	6	5	4	3	2	1	0	Address	After reset
UF0INT1	IT17	IT16	IT15	IT14	IT13	IT12	IT11	IT10	F058AH	Undefined

Bit position	Bit name	Function
7 - 0	IT17 to IT10	These bits store data for Endpoint7.

The operation of the UF0INT1 register is illustrated below.

Figure 13-11. Operation of UF0INT1 Register



(12) UF0 interrupt 2 register (UF0INT2)

The UF0INT2 register is a 64-byte FIFO that stores data for Endpoint8 (to be passed to SIE).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

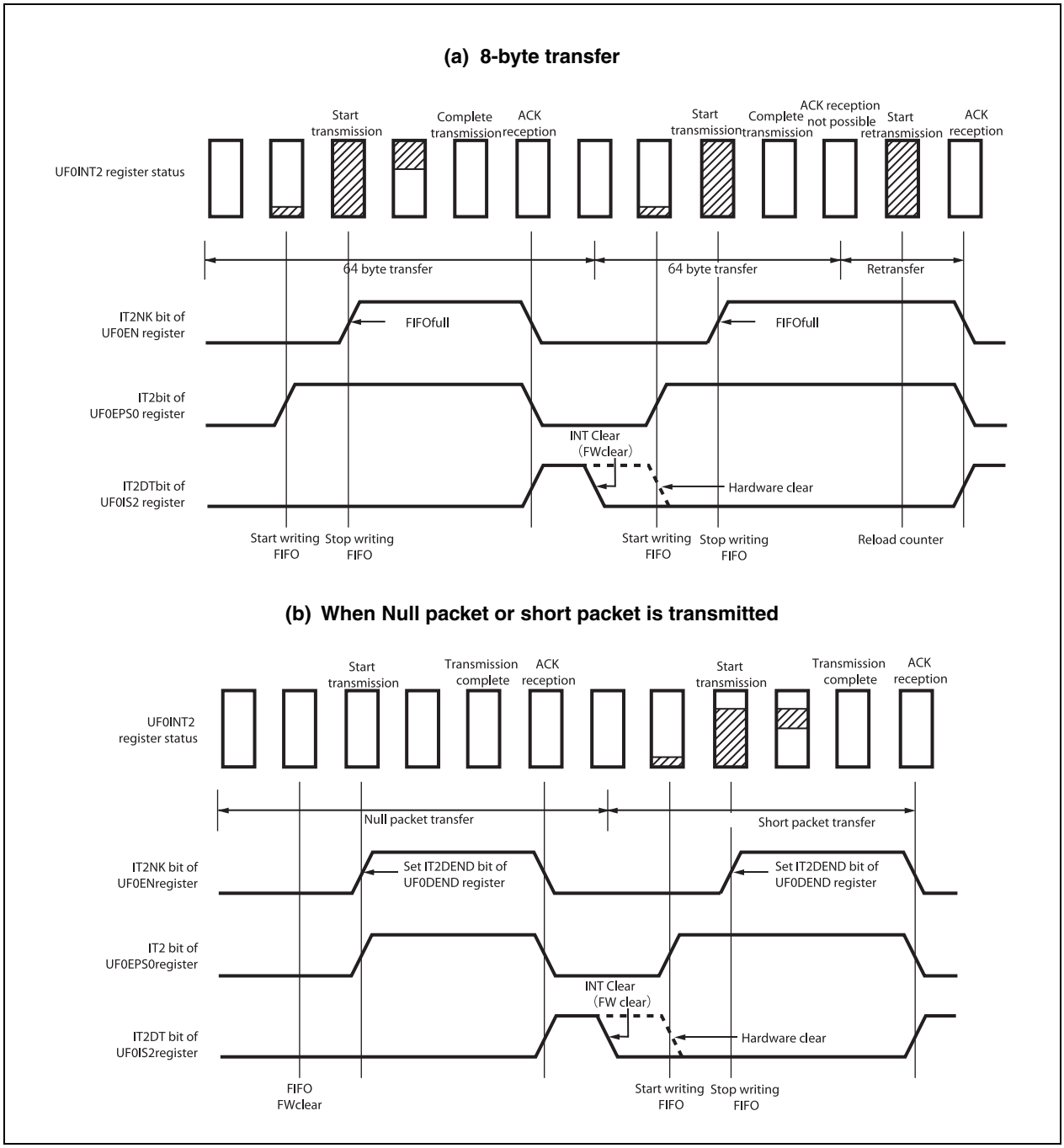
The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint8 only when the IT2NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). When the data is transmitted and the host correctly receives it, the IT2NK bit of the UF0EN register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the UF0INT2 register and the IT2DEND bit of the UF0DEND register is set to 1 (IT2 bit of the UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0INT2 register is cleared and the IT2DEND bit of the UF0DEND register is set to 1 (IT2 bit of the UF0EPS0 register = 1 (data exists)).

	7	6	5	4	3	2	1	0	Address	Initial value
UF0INT2	IT27	IT26	IT25	IT24	IT23	IT22	IT21	IT20	F058BH	Undefined

Bit position	Bit name	Function
7 - 0	IT27 - IT20	These bits store data for Endpoint8.

The operation of the UF0INT2 register is illustrated below.

Figure 13-12. Operation of UF0INT2 Register



13.6.5 EPC request data registers

(1) UF0 device status register L (UF0DSTL)

This register stores the value that is to be returned in response to the GET_STATUS Device request.

This register can be read or written in 8-bit units.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Device request.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DSTL	0	0	0	0	0	0	RMWK	SFPW	F05A2H	00H

Bit position	Bit name	Function
1	RMWK	This bit specifies whether the remote wakeup function of the device is used. 1: Enabled 0: Disabled If the device supports a remote wakeup function, this bit is set to 1 by hardware when the SET_FEATURE Device request has been received, and is cleared to 0 by hardware when the CLEAR_FEATURE Device request has been received. If the device does not support a remote wakeup function, make sure that the SET_FEATURE Device request is not issued from the host.
0	SFPW	This bit indicates whether the device is self-powered or bus-powered. 1: Self-powered 0: Bus-powered

(2) UF0 EP0 status register L (UF0E0SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint0 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in USBF, the E0HALT bit is set to 1 by FW. A write access to this register is ignored while a USB-side access to Endpoint0 is being received.

When the E0HALT bit is set to 1 by FW, it is not reflected until the next SETUP token is received if the control transfer immediately before is for the SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, GET_STATUS Endpoint0 request, or an FW-processed request.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint0 request. If Endpoint0 has stalled, the UF0E0W and UF0E0R registers are cleared, and the EP0NKW and EP0NKR bits of the UF0E0N register are cleared to 0.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0SL	0	0	0	0	0	0	0	E0HALT	F05A6H	00H

Bit position	Bit name	Function
0	E0HALT	This bit indicates the status of Endpoint0. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint0 request has been received, and cleared to 0 by hardware when the CLEAR_FEATURE Endpoint0 request has been received. DATA PID is initialized to DATA0.

(3) UF0 EP1 status register L (UF0E1SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint1 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint1, the E1HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint1 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint1 request. If Endpoint1 has stalled, the UF0BI1 register is cleared and the BKI1NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint1, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E1SL	0	0	0	0	0	0	0	E1HALT	F05A8H	00H

Bit position	Bit name	Function
0	E1HALT	This bit indicates the status of Endpoint1. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint1 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint1 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint1 is linked has correctly been received. DATA PID is initialized to DATA0.

(4) UF0 EP2 status register L (UF0E2SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint2 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint2, the E2HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint2 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint2 request. If Endpoint2 has stalled, the UF0BO1 register is cleared and the BKO1NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint2, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E2SL	0	0	0	0	0	0	0	E2HALT	F05AAH	00H

Bit position	Bit name	Function
0	E2HALT	This bit indicates the status of Endpoint2. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint2 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint2 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint2 is linked has correctly been received. DATA PID is initialized to DATA0.

(5) UF0 EP3 status register L (UF0E3SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint3 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint3, the E3HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint3 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint3 request. If Endpoint3 has stalled, the UF0BI2 register is cleared and the BKI2NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint3, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E3SL	0	0	0	0	0	0	0	E3HALT	F05ACH	00H

Bit position	Bit name	Function
0	E3HALT	This bit indicates the status of Endpoint3. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint3 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint3 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint3 is linked has correctly been received. DATA PID is initialized to DATA0.

(6) UF0 EP4 status register L (UF0E4SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint4 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint4, the E4HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint4 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint4 request. If Endpoint4 has stalled, the UF0BO2 register is cleared and the BKO2NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint4, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E4SL	0	0	0	0	0	0	0	E4HALT	F05AEH	00H

Bit position	Bit name	Function
0	E4HALT	This bit indicates the status of Endpoint4. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint4 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint4 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint4 is linked has correctly been received. DATA PID is initialized to DATA0.

(7) UF0 EP7 status register L (UF0E7SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint7 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint7, the E7HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint7 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint7 request. If Endpoint7 has stalled, the UF0INT1 register is cleared and the IT1NK bit is cleared to 0.

Because writing this register is always masked when transfer to Endpoint7, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E7SL	0	0	0	0	0	0	0	E7HALT	F05B4H	00H

Bit position	Bit name	Function
0	E7HALT	This bit indicates the status of Endpoint7. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint7 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint7 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint7 is linked has correctly been received. DATA PID is initialized to DATA0.

(8) UF0 EP8 status register L (UF0E8SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint8 request.
This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.
If an error occurs in Endpoint8, the E8HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint8 is being received.
The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint8 request. If Endpoint7 has stalled, the UF0INT2 register is cleared and the IT2NK bit is cleared to 0.
Because writing this register is always masked when transfer to Endpoint8, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	Initial value
UF0E8SL	0	0	0	0	0	0	0	E8HALT	F05B6H	00H

Bit position	Bit name	Function
0	E8HALT	This bit indicates the status of Endpoint8. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint8 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint8 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint8 is linked has correctly been received. DATA PID is initialized to DATA0.

(9) UFO Address register (UF0ADRS)

This register stores the device address.

This register is read-only, in 8-bit units.

The device address sent by the SET_ADDRESS request is analyzed and the resultant value is automatically written to this register. If the SET_ADDRESS request is processed by FW, the value of this register is reflected as the device address when the SUCCESS signal is received in the status stage.

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ADRS	0	ADRS6	ADRS5	ADRS4	ADRS3	ADRS2	ADRS1	ADRS0	F05C0H	00H

Bit position	Bit name	Function
6 - 0	ADRS6 - ADRS0	These bits hold the device address of SIE.

(10) UF0 configuration register (UF0CNF)

This register stores the value that is to be returned in response to the GET_CONFIGURATION request.

This register is read-only, in 8-bit units.

When the SET_CONFIGURATION request is received, its wValue is automatically written to this register.

When a change of the value of this register from 00H to other than 00H is detected, the CONF bit of the UF0MODS register is set to 1. If the SET_CONFIGURATION request is processed by FW, the status of this register is immediately reflected on the UF0MODS register as soon as data has been written to this register (CONF bit = 1 before completion of the status stage).

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CNF	0	0	0	0	0	0	CONF1	CONF0	F05C1H	00H

Bit position	Bit name	Function
1, 0	CONF1, CONF0	These bits hold the data to be returned in response to the GET_CONFIGURATION request.

(11) UF0 interface 0 register (UF0IF0)

This register stores the value that is to be returned in response to the GET_INTERFACE wIndex = 0 request.
This register is read-only, in 8-bit units.
When the SET_INTERFACE request is received, its wValue is automatically written to this register.
If the SET_INTERFACE request is processed by FW, wIndex and wValue are decoded, and the setting of endpoint is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IF0	0	0	0	0	0	IF02	IF01	IF00	F05C2H	00H

Bit position	Bit name	Function
2 - 0	IF02 - IF00	These bits hold the data to be returned in response to GET_INTERFACE wIndex = 0 request.

(12) UF0 interface 1 - 4 registers (UF0IF1 - UF0IF4)

These registers store the value that is to be returned in response to the GET_INTERFACE wIndex = n request (n = 1 to 4).
These registers are read-only, in 8-bit units.
When the SET_INTERFACE request is received, its wValue is automatically written to these registers.
These registers are invalidated according to the setting of the UF0AIFN and UF0AAS registers.
If the SET_INTERFACE request is processed by FW, wIndex and wValue are decoded, and the setting of endpoint is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Caution Do not execute a write access to this register. If written, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IF1	0	0	0	0	0	IF12	IF11	IF10	F05C3H	00H
UF0IF2	0	0	0	0	0	IF22	IF21	IF20	F05C4H	00H
UF0IF3	0	0	0	0	0	IF32	IF31	IF30	F05C5H	00H
UF0IF4	0	0	0	0	0	IF42	IF41	IF40	F05C6H	00H

Bit position	Bit name	Function
2 - 0	IFn2 - IFn0	These bits hold the data to be returned in response to GET_INTERFACE wIndex = n request.

Remark n = 1 - 4

(13) UF0 descriptor length register (UF0DSCL)

This register stores the length of the value that is to be returned in response to the GET_DESCRIPTOR Configuration request. The value of this register is the number of bytes of all the descriptors set by the UF0CIEn register minus 1 (n = 0 to 255). The total descriptor length that is to be returned in response to the GET_DESCRIPTOR Configuration request is determined according to the value of this register.

This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Processing of wLength is automatically controlled. If this register is set to 00H, it means that the descriptor to be returned is 1 byte long. If the register is set to FFH, a descriptor length of 256 bytes is returned. When a descriptor exceeding 256 bytes in length is used, set the CDCGDST bit of the UF0MODC register to 1 and process the GET_DESCRIPTOR request by FW (at this time, the CDCGD bit of the UF0MODS register is also set to 1).

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DSCL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0	F05D0H	00H

Bit position	Bit name	Function
7 - 0	DPL7 - DPL0	These bits set the value of the number of bytes of all the descriptors to be returned in response to the GET_DESCRIPTOR Configuration request minus 1.

(14) UF0 device descriptor registers 0 - 17 (UF0DD0 - UF0DD17)

These registers store the value to be returned in response to the GET_DESCRIPTOR Device request.

These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Cautions 1. To rewrite these registers, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the set value.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DDn (n = 0 to 17)									See **Table 13-5**.	Undefined

Table 13-5. Mapping and Data of UF0 Device Descriptor Registers

Symbol	Address	Field Name	Contents
UF0DD0	F05D1H	bLength	Size of this descriptor
UF0DD1	F05D2H	bDescriptorType	Device descriptor type
UF0DD2	F05D3H	bcdUSB	Value below decimal point of Rev. number of USB specification
UF0DD3	F05D4H		Value above decimal point of Rev. number of USB specification
UF0DD4	F05D5H	bDeviceClass	Class code
UF0DD5	F05D6H	bDeviceSubClass	Subclass code
UF0DD6	F05D7H	bDeviceProtocol	Protocol code
UF0DD7	F05D8H	bMaxPacketSize0	Maximum packet size of Endpoint0
UF0DD8	F05D9H	idVendor	Lower value of vendor ID
UF0DD9	F05DAH		Higher value of vendor ID
UF0DD10	F05DBH	idProduct	Lower value of product ID
UF0DD11	F05DCH		Higher value of product ID
UF0DD12	F05DDH	bcdDevice	Lower value of device release number
UF0DD13	F05DEH		Higher value of device release number
UF0DD14	F05DFH	iManufacturer	Index of string descriptor describing manufacturer
UF0DD15	F05E0H	iProduct	Index of string descriptor describing product
UF0DD16	F05E1H	ISerialNumber	Index of string descriptor describing device serial number
UF0DD17	F05E2H	BNumConfigurations	Number of settable configurations

(15) UF0 configuration/interface/endpoint descriptor registers 0 - 255 (UF0CIE0 - UF0CIE255)

These registers store the value to be returned in response to the GET_DESCRIPTOR Configuration request.

These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Descriptor information of up to 256 bytes can be stored in these registers. Store each descriptor in the order of Configuration, Interface, and Endpoint (see **Table 13-6**). If there are two or more Interfaces, repeatedly store the data following the Interface descriptor.

Table 13-6. Mapping of UF0CIEn Register

Address	Descriptor Stored
F05E3H	Configuration descriptor (9 bytes)
F05F5H	Interface descriptor (9 bytes)
F0607H	Endpoint1 descriptor (7 bytes)
F0615H	Endpoint2 descriptor (7 bytes)
F0623H	Endpoint3 descriptor (7 bytes)
:	:
F02xxH	Interface descriptor (9 bytes)
F02xxH + 9	Endpoint1 descriptor (7 bytes)
F02xxH + 16	Endpoint2 descriptor (7 bytes)
F02xxH + 23	Endpoint3 descriptor (7 bytes)
:	:

The range of the valid data that can be set to these registers varies according to the setting of the UF0DSCL register. In addition to the descriptors listed in Table 13-7, descriptors peculiar to classes and vendors can also be stored.

If all the values are fixed, they can be stored in ROM.

Cautions 1. To rewrite these registers, set the EP0NKA bit to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access and a write access.

2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the set value.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CIEn (n = 0 to 255)									F05E3H to F06E2H	Undefined

Table 13-7. Data of UF0CIEn Register

(a) Configuration descriptor (9 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	wTotalLength	Lower value of the total number of bytes of Configuration, all Interface, and all Endpoint descriptors
3		Higher value of the total number of bytes of Configuration, all Interface, and all Endpoint descriptors
4	bNumInterface	Number of Interfaces
5	bConfigurationValue	Value to select this Configuration
6	iConfiguration	Index of string descriptor describing this Configuration
7	bmAttributes	Features of this Configuration (self-powered, without remote wakeup)
8	MaxPower	Maximum power consumption of this Configuration (unit: mA)[Note]

Note Shown in 2 mA units. (example: 50 = 100 mA)

(b) Interface descriptor (9 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bInterfaceNumber	Value of this Interface
3	bAlternateSetting	Value to select alternative setting of Interface
4	bNumEndpoints	Number of usable Endpoints
5	bInterfaceClass	Class code
6	bInterfaceSubClass	Subclass code
7	bInterfaceProtocol	Protocol code
8	Interface	Index of string descriptor describing this Interface

(c) Endpoint descriptor (7 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bEndpointAddress	Address/transfer direction of this Endpoint
3	bmAttributes	Transfer type
4	wMaxPaketSize	Lower value of maximum number of transfer data
5		Higher value of maximum number of transfer data
6	bInterval	Transfer interval

13.6.6 EPC Peripheral control register

(1) USB function 0 buffer control register (UF0BC)

This is the register which performs enable control, floating control for USB function input buffer.
Read, write is possible in 8bit unit.

	7	6	5	4	3	2	1	0	Address	Initial value
UF0BC	ENO2EN	0	0	0	0	0	IEN1EN	IOR1En	F059EH	00H

Bit position	Bit name	Function
7	ENO2EN	Controls buffer use of USBPUC pin. 1:Buffer enable 0:buffer disable
1	IEN1EN	Controls buffer use of USB. 1:Buffer enable 0:buffer disable Caution Clear this bit to (0) when USB is not in use. If this bit is set to (1), then 3 mA (TYP.)current flows by default irrespective of whether USB is in use or not.
0	IOR1EN	Control floating target of the USB buffer. 1: Floating target disabled. 0: Floating target enabled. When cable is not connected (When data input is floating), recognition of Bus Reset, Suspend, Resume due to negative value is prevented. When this bit is set to (1) control floating devices using VBUS signal (Cable connection recognition) .

13.7 STALL Handshake or No Handshake

Errors of USBF are defined to be handled as follows.

Transfer Type	Transaction	Target Packet	Error Type	Function Response	Processing
Control transfer/ bulk transfer/ interrupt transfer	IN/OUT/SETUP	Token	Endpoint not supported	No response	None
			Endpoint transfer direction mismatch	No response	None
			CRC error	No response	None
			Bit stuffing error	No response	None
Control transfer/ bulk transfer	OUT/SETUP	Data	Timeout	No response	None
			PID check error	No response	None
			Unsupported PID (other than Data PID)	No response	None
			CRC error	No response	Discard received data
			Bit stuffing error	No response	Discard received data
	OUT	Data	Data PID mismatch	ACK	Discard received data
Control transfer (SETUP stage)	SETUP	Data	Overrun	No response	Discard received data
Control transfer (data stage)	OUT	Data	Overrun	No response[Note 1]	Set SNDSTL bit of UF0SDS register to 1 and discard received data
Control transfer (status stage)	OUT	Data	Overrun	ACK or no response[Note 2]	Set SNDSTL bit of UF0SDS register to 1 and discard received data
Bulk transfer	OUT	Data	Overrun	No response[Note 1]	Set EnHALT bit of UF0EnSL register (n = 0-4, 7, 8) to 1
Control transfer/ bulk transfer/ interrupt transfer	IN	Handshake	PID check error	–	Hold transferred data and re-transfer data[Note 3]
			Unsupported PID (other than ACK PID)	–	Hold transferred data and re-transfer data[Note 3]
			Timeout	–	Hold transferred data and re-transfer data[Note 3]

Notes 1. A STALL response is made to re-transfer by the host.

2. An ACK response is made if the transfer data is of less than MaxPacketSize and the data received in the status stage is discarded. If MaxPacketSize is exceeded, no response is made, the SNDSTL bit of the UF0SDS register is set to 1, and the received data is discarded.

3. If an OUT transaction indicating a change from the data stage to the status stage is received during control transfer, an error is not handled and it is assumed that reception has been correctly completed.

Cautions 1. It is judged by the Alternative Setting number currently set whether the target Endpoint is valid or invalid.

2. For the response to the request included in control transfer to/from Endpoint0, see 13.5 Requests.

13.8 Register Values in Specific Status

Table 13-8. Register Values in Specific Status (1/2)

Register Name	After CPU Reset ($\overline{\text{RESET}}$)	After Bus Reset
UF0E0N register	00H	Value is held.
UF0E0NA register	00H	Value is held.
UF0EN register	00H	Value is held.
UF0ENM register	00H	Value is held.
UF0SDS register	00H	Value is held.
UF0CLR register	00H	Value is held.
UF0SET register	00H	Value is held.
UF0EPS0 register	00H	Value is held.
UF0EPS1 register	00H	Value is held.
UF0EPS2 register	00H	Value is held.
UF0IS0 register	00H	Value is held.
UF0IS1 register	00H	Value is held.
UF0IS2 register	00H	Value is held.
UF0IS3 register	00H	Value is held.
UF0IS4 register	00H	Value is held.
UF0IM0 register	00H	Value is held.
UF0IM1 register	00H	Value is held.
UF0IM2 register	00H	Value is held.
UF0IM3 register	00H	Value is held.
UF0IM4 register	00H	Value is held.
UF0IC0 register	FFH	Value is held.
UF0IC1 register	FFH	Value is held.
UF0IC2 register	FFH	Value is held.
UF0IC3 register	FFH	Value is held.
UF0IC4 register	FFH	Value is held.
UF0FIC0 register	00H	Value is held.
UF0FIC1 register	00H	Value is held.
UF0DEND register	00H	Value is held.
UF0GPR register	00H	Value is held.
UF0MODC register	00H	Value is held.
UF0MODS register	00H	Bit 2 (CONF): Cleared (0), Other bits: Value is held.
UF0AIFN register	00H	Value is held.
UF0AAS register	00H	Value is held.
UF0ASS register	00H	00H
UF0E1IM register	00H	Value is held.
UF0E2IM register	00H	Value is held.

Table 13-8. Register Values in Specific Status (2/2)

Register Name	After CPU Reset ($\overline{\text{RESET}}$)	After Bus Reset
UF0E3IM register	00H	Value is held.
UF0E4IM register	00H	Value is held.
UF0E7IM register	00H	Value is held.
UF0E8IM register	00H	Value is held.
UF0E0R register	Undefined[Note 1]	Value is held.
UF0E0L register	00H	Value is held.
UF0E0ST register	00H	00H
UF0E0W register	Undefined[Note 1]	Value is held.
UF0BO1 register	Undefined[Note 1]	Value is held.
UF0BO1L register	00H	Value is held.
UF0BO2 register	Undefined[Note 1]	Value is held.
UF0BO2L register	00H	Value is held.
UF0BI1 register	Undefined[Note 1]	Value is held.
UF0BI2 register	Undefined[Note 1]	Value is held.
UF0INT1 register	Undefined	Value is held.
UF0INT2 register	Undefined	Value is held.
UF0DSTL register	00H	00H
UF0E0SL register	00H	00H
UF0E1SL register	00H	00H
UF0E2SL register	00H	00H
UF0E3SL register	00H	00H
UF0E4SL register	00H	00H
UF0E7SL register	00H	00H
UF0E8SL register	00H	00H
UF0ADRS register	00H	00H
UF0CNF register	00H	00H
UF0IF0 register	00H	00H
UF0IF1 register	00H	00H
UF0IF2 register	00H	00H
UF0IF3 register	00H	00H
UF0IF4 register	00H	00H
UF0DSCL register	00H	Value is held.
UF0DDn register (n = 0 to 17)	**Note 2**	**Note 2**
UF0CIEn register (n = 0 to 255)	**Note 2**	**Note 2**

Notes 1. This register can be cleared to 0 by the $\overline{\text{RESET}}$ signal because its write pointer, counter, and read pointer are cleared to 0 when the $\overline{\text{RESET}}$ signal becomes active, in the same manner as clearing by the UF0FICn register, as the register is controlled by FIFO.

2. This register cannot be cleared to 0. Because data can be written to it by FW, however, any value can be written to the register (before doing so, however, be sure to set the EP0NKA bit of the UF0E0NA register to 1).

13.9 FW Processing

The following FW processing is performed.

- Setting processing on device side for the SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests during enumeration processing
- Analysis and processing of XXXXStandard, XXXXClass, and XXXXVendor requests not subject to automatic processing
- Reading data following bulk-transferred OUT token from receive buffer
- Writing data to be returned in response to bulk-transferred IN token
- Writing data to be returned in response to interrupt-transferred token

The following table lists the requests supported by FW.

Table 13-9. FW-Supported Standard Requests

Request	Reception Side	Processing/ Frequency	Explanation
CLEAR_FEATURE	Interface	Automatic STALL response	It is considered that this request does not come to Interface because there is no function selector value, though it is reserved for bmRequestType. When this request is received, the hardware makes an automatic STALL response.
SET_FEATURE	Interface	Automatic STALL response	It is considered that this request does not come to Interface because there is no function selector value, though it is reserved for bmRequestType. When this request is received, the hardware makes an automatic STALL response.
GET_DESCRIPTOR	String	FW	Returns the string descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and writes the data to be returned to the host, to the UF0E0W register.
SET_DESCRIPTOR	Device	FW	Rewrites the device descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and the writes the data for the next control transfer (OUT) to the UF0DDn register (n = 0 to 17).
SET_DESCRIPTOR	Configuration	FW	Rewrites the configuration descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and the writes the data for the next control transfer (OUT) to the UF0CIEn register (n = 0 to 255).
SET_DESCRIPTOR	String	FW	Rewrites the string descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and loads the data for the next control transfer (OUT).
Other	NA	FW	When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the request from the CPUDEC interrupt request, and performs the necessary processing.

13.9.1 Initialization processing

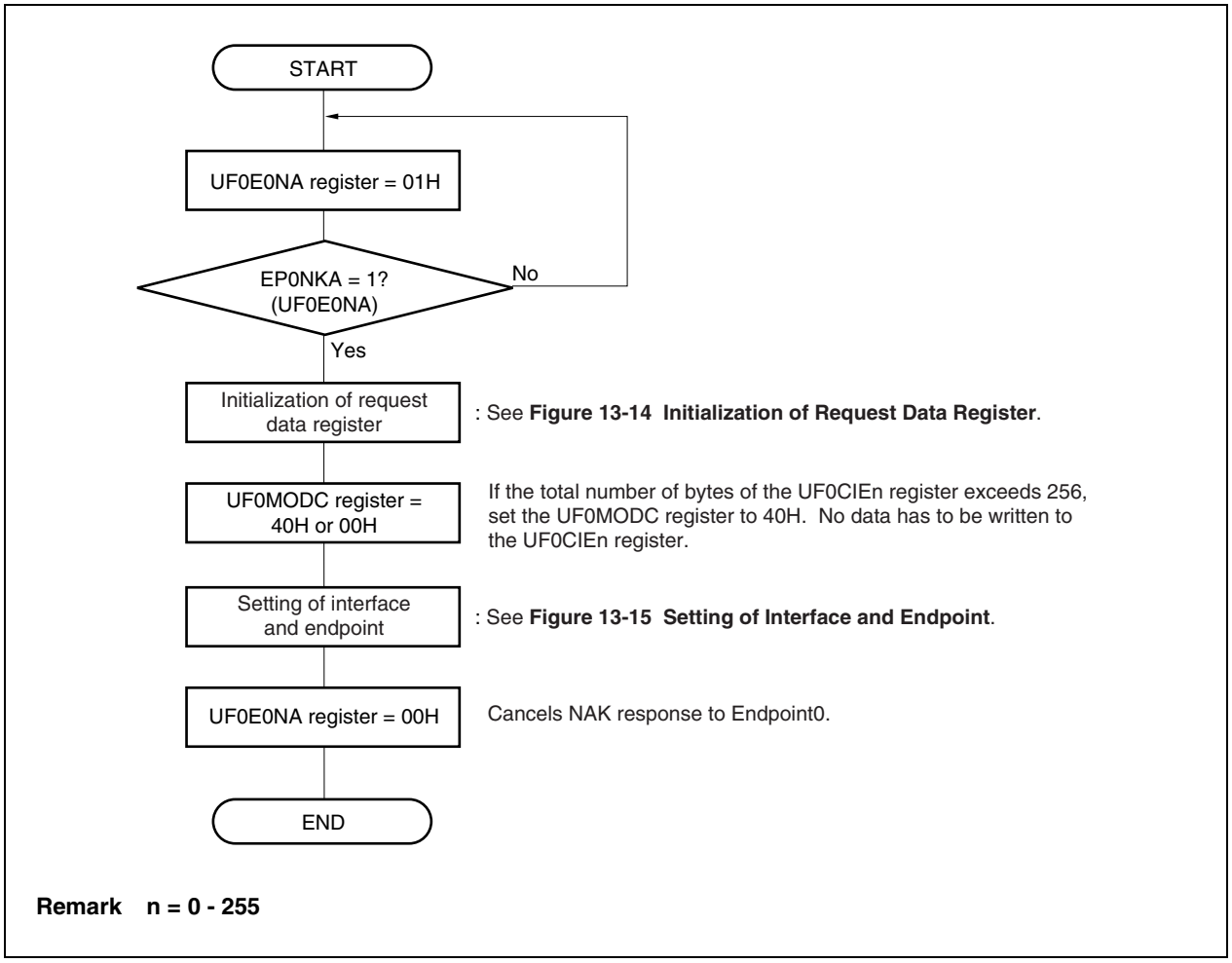
Initialization processing is executed in the following two ways.

- Initialization of request data register
- Setting of interrupt

When a request data register is initialized, data for the GET_XXXX request to which a value is to be automatically returned is written and an endpoint is allocated to an interface. In the interrupt settings, the interrupt sources that do not have to be checked can be masked by using the UF0IMn register (n = 0- 4).

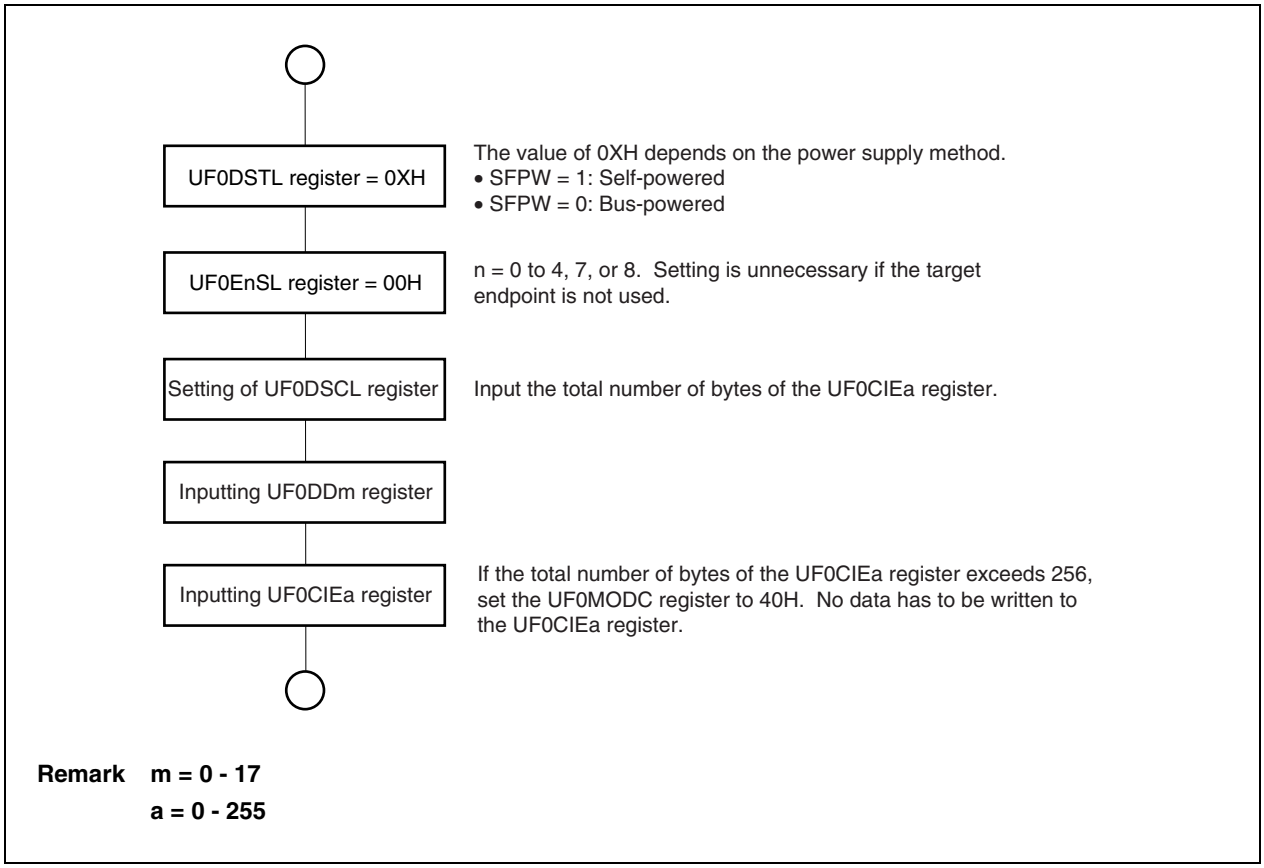
The following flowcharts illustrate the above processing.

Figure 13-13. Initializing Request Data Register



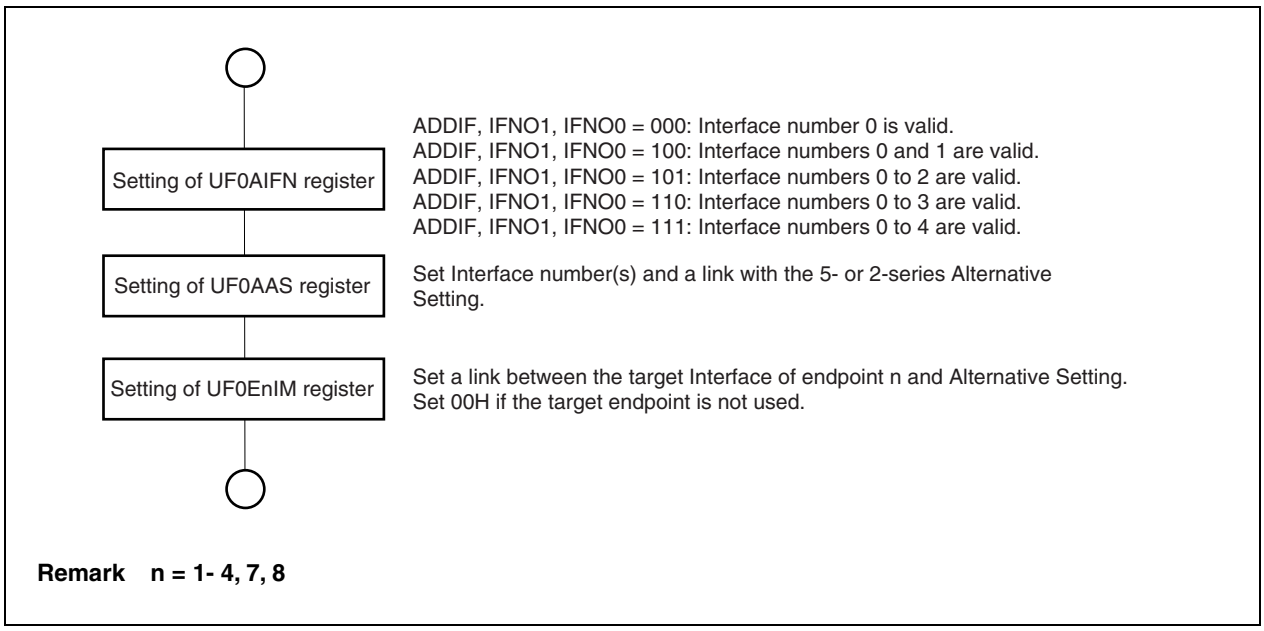
Remark n = 0 - 255

Figure 13-14. Initialization Settings of Request Data Register



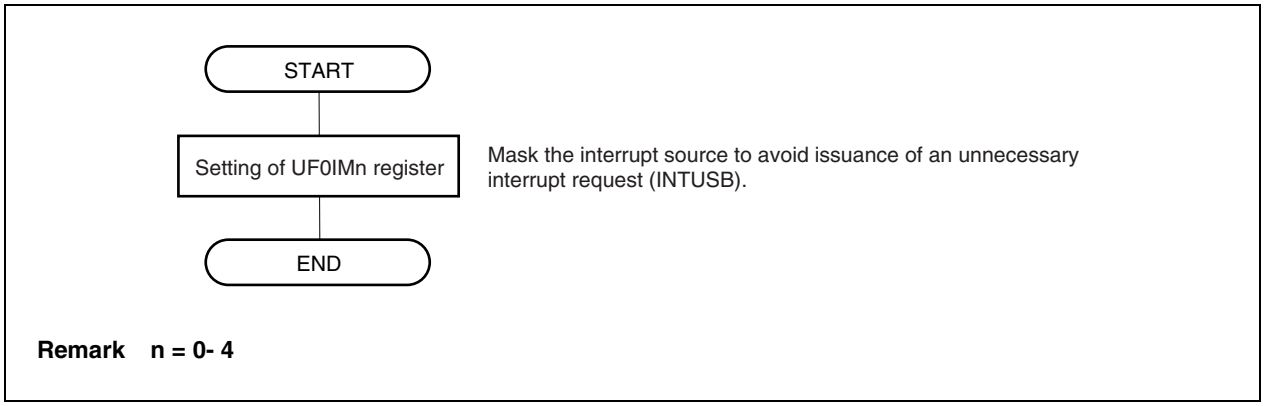
Remark m = 0 - 17
a = 0 - 255

Figure 13-15. Setting of Interface and Endpoint



Remark n = 1- 4, 7, 8

Figure 13-16. Setting of Interrupt

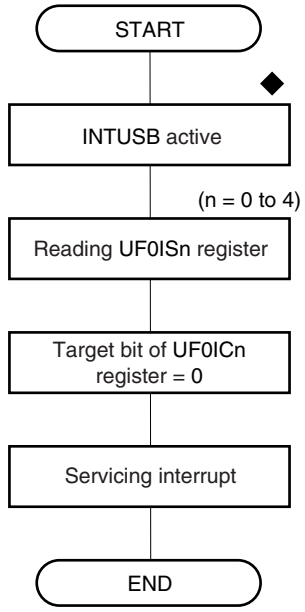


Remark n = 0- 4

13.9.2 Interrupt servicing

The following flowchart illustrates how an interrupt is serviced.

Figure 13-17. Interrupt Servicing



Remark ♦: Processing by hardware

The following bits of the UF0ISn register are automatically cleared by hardware when a given condition is satisfied (n = 0- 4).

- E0INDT, E0ODT, SUCES, STG, and CPUDEC bits of UF0IS1 register
- BKI2DT, BKI1DT, and IT1DT bits of UF0IS2 register
- BKO2FL, BKO2DT, BKO1FL, and BKO1DT bits of UF0IS3 register

Because clearing an interrupt source by the UF0ICn register is given a lower priority than setting an interrupt source by hardware, the interrupt source may not be cleared depending on the timing (n = 0- 4).

13.9.3 USB main processing

USB main processing involves processing USB transactions. The types of transactions to be processed are as follows.

- Fully automatically processed request for control transfer
- Automatically processed requests for control transfer
  (SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)
- CPUDEC request for control transfer
- Processing for bulk transfer (IN)
- Processing for bulk transfer (OUT)
- Processing for interrupt transfer (IN)

Processing for endpoint n involves writing or reading for data transfer. The flowchart shown below is for PIO.

(1) Fully automatically processed request for control transfer

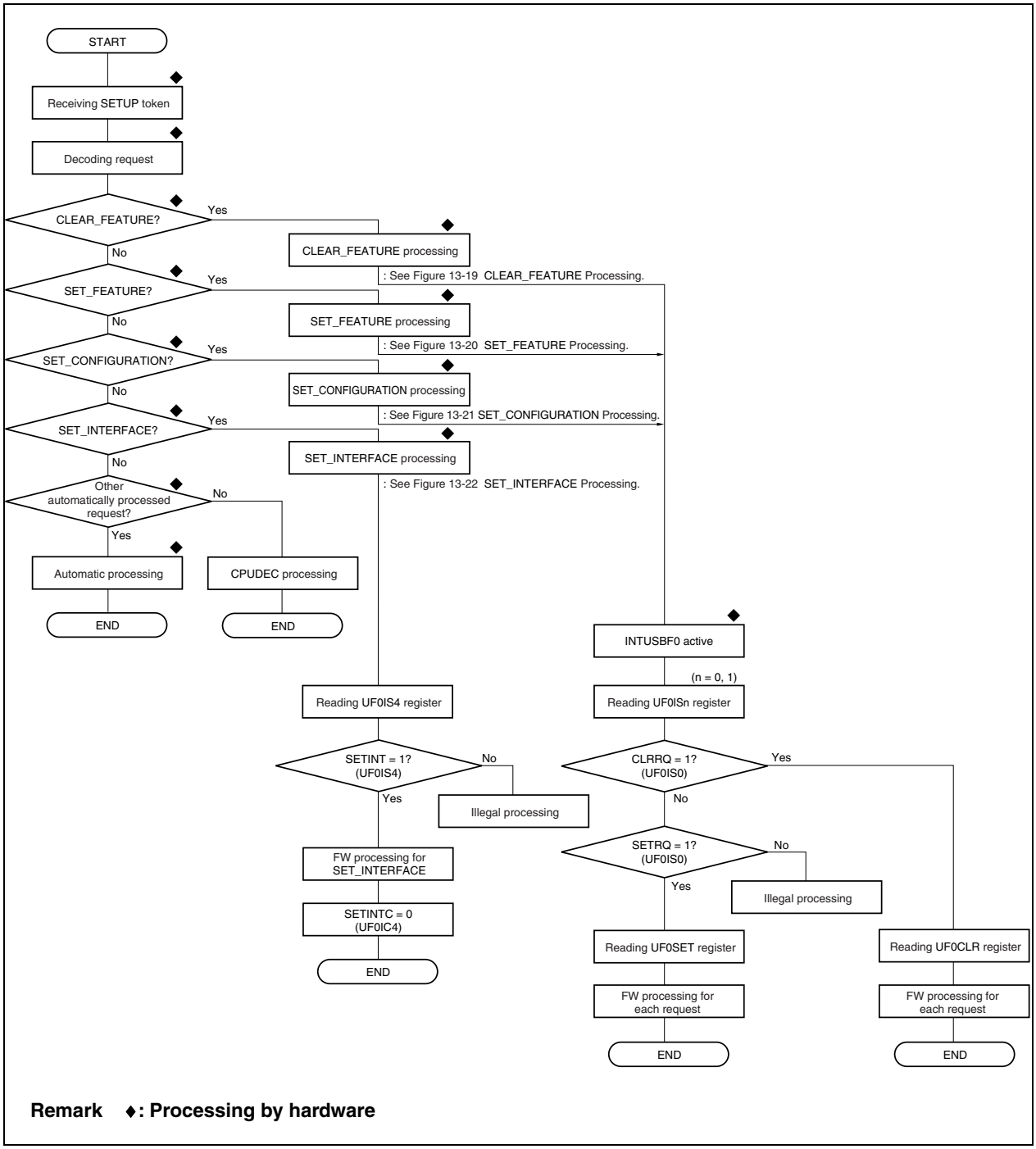
Because the fully automatically processed request for control transfer is executed by hardware, it cannot be referenced by FW. Therefore, FW does not have to perform any special processing for this request.

(2) Automatically processed requests for control transfer
(SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)

Processing to write a register for automatically processed requests for control transfer, such as SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests, is automatically executed by hardware, but an interrupt request is issued for recognition on the device side. This processing may be ignored if there is no special processing to be executed.

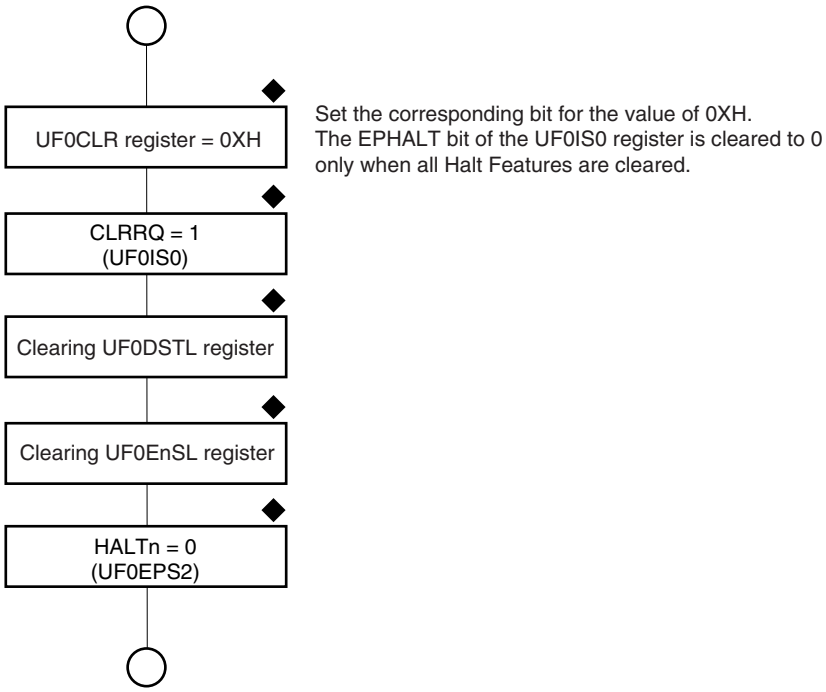
The flowcharts are shown below.

Figure 13-18. Automatically Processed Requests for Control Transfer



Remark ♦: Processing by hardware

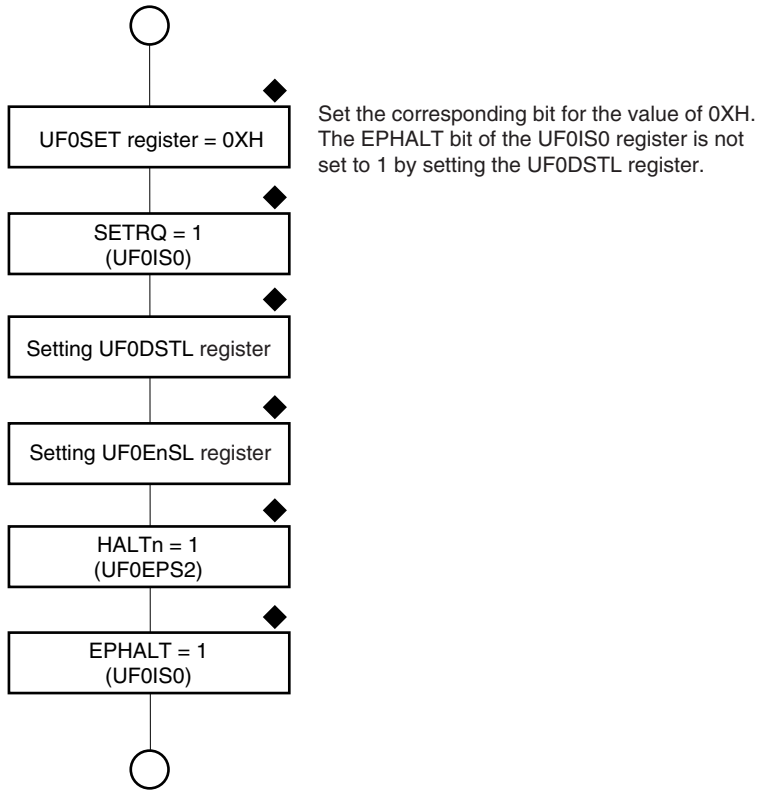
Figure 13-19. CLEAR_FEATURE Processing



Remarks 1. n = 0- 4, 7, 8

2. ♦: Processing by hardware

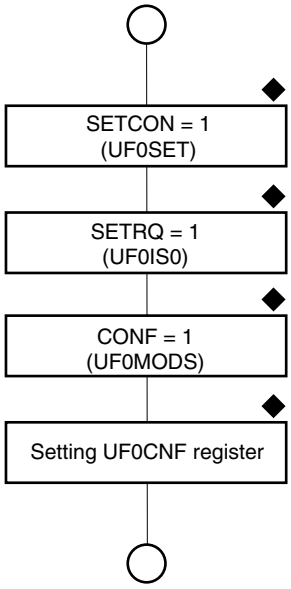
Figure 13-20. SET_FEATURE Processing



Remarks 1. n = 0- 4, 7, 8

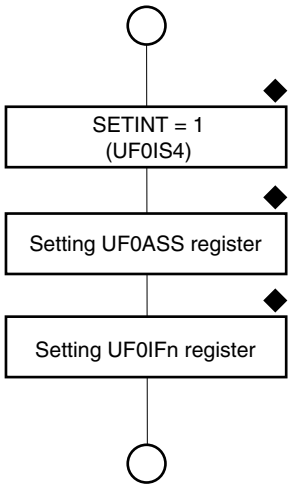
2. ♦: Processing by hardware

Figure 13-21. SET_CONFIGURATION Processing



Remark ♦: Processing by hardware

Figure 13-22. SET_INTERFACE Processing



Remarks 1. n = 0- 4

2. ♦: Processing by hardware

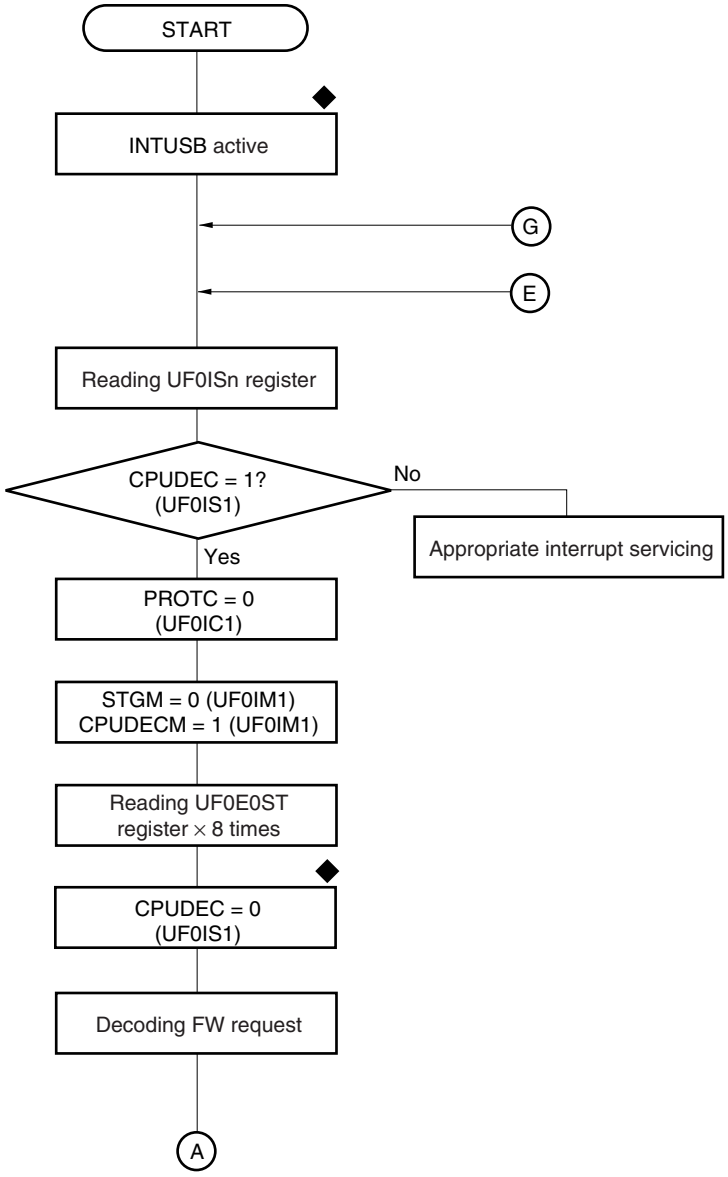
(3) CPUDEC request for control transfer

The CPUDEC request can be classified into three types of processing: control transfer (write), control transfer (read), and control transfer (without data). Control transfer (write) indicates a request that uses the OUT transaction in the data stage (e.g., SET_DESCRIPTOR), and control transfer (read) indicates a request that uses the IN transaction in the data stage (e.g., GET_DESCRIPTOR). Control transfer (without data) indicates a request that has no data stage (e.g., SET_CONFIGURATION).

The flowcharts are shown below.

Figure 13-23. CPUDEC Request for Control Transfer (1/12)

(a) Token phase (1/2)

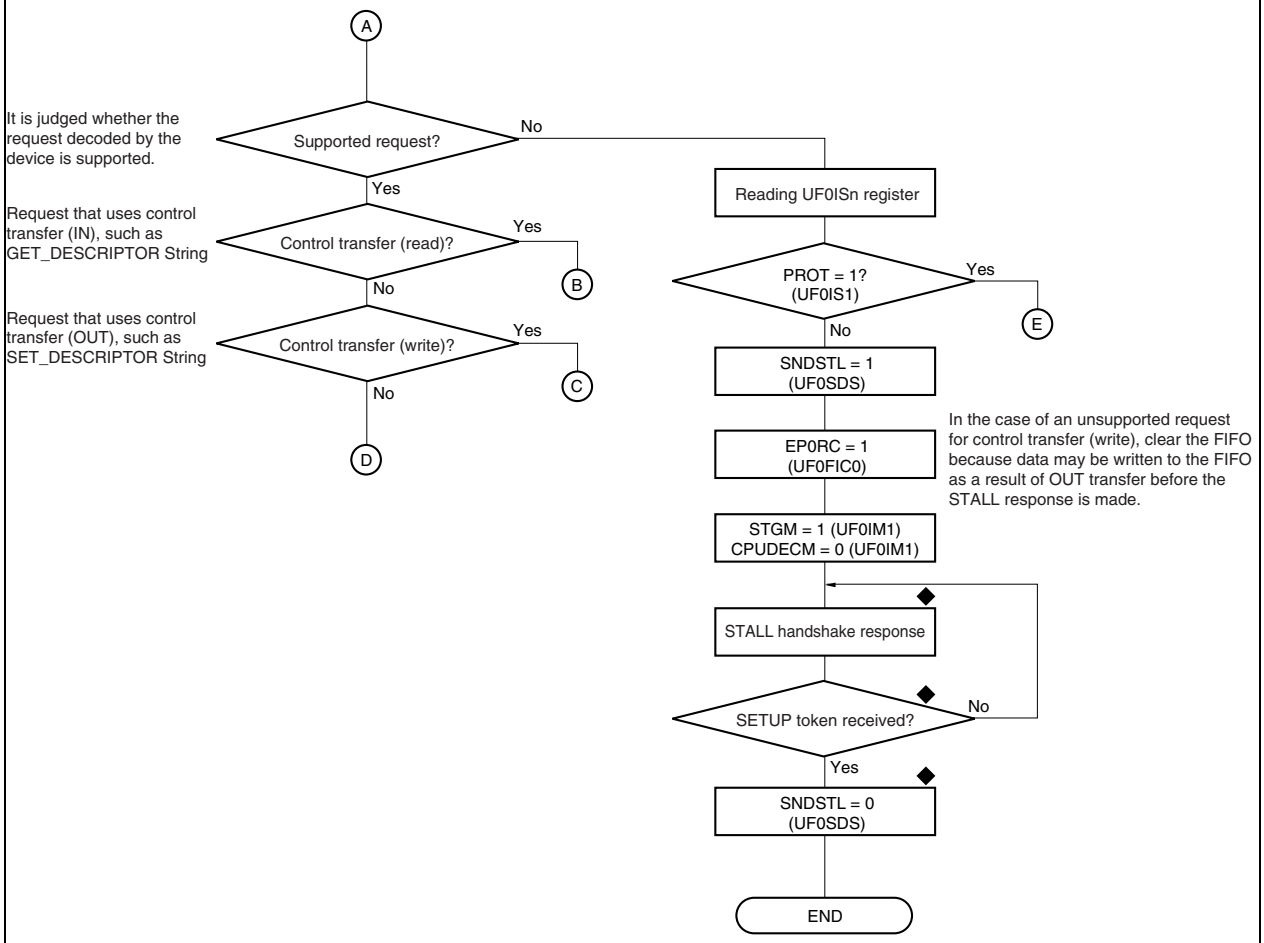


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (2/12)

(a) Token phase (2/2)

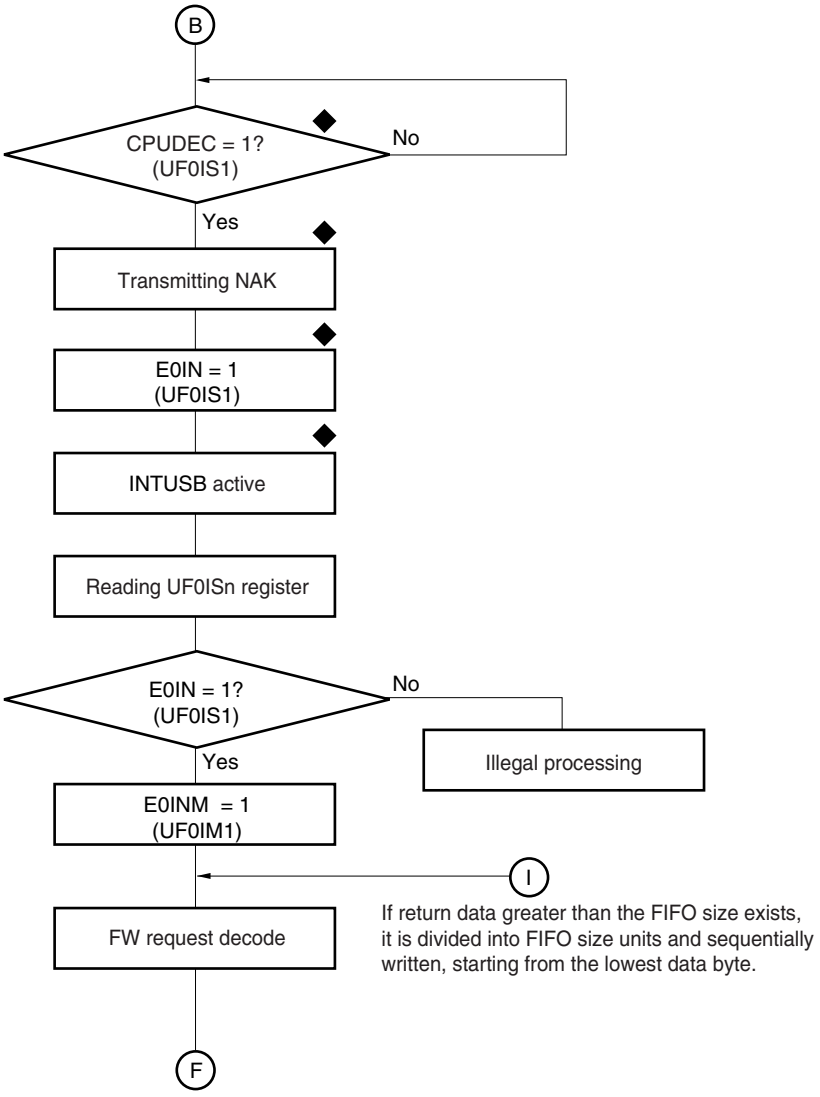


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (3/12)

(b) Control transfer (read) (1/4)

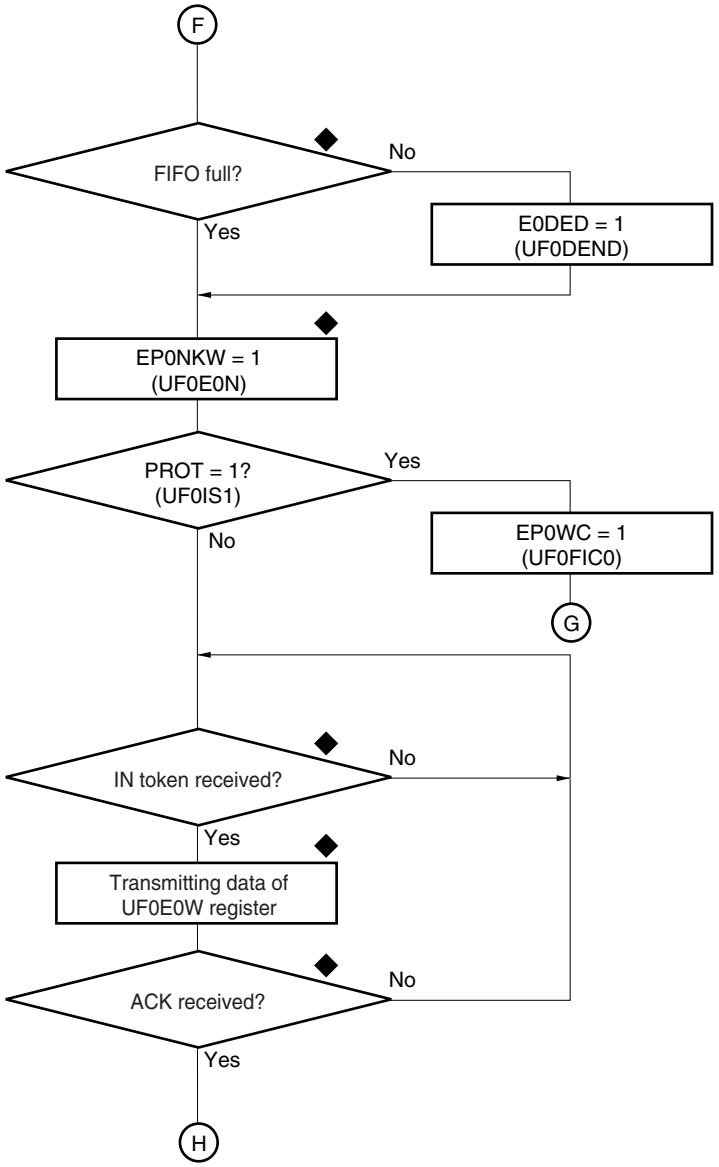


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23 CPUDEC Request for Control Transfer (4/12)

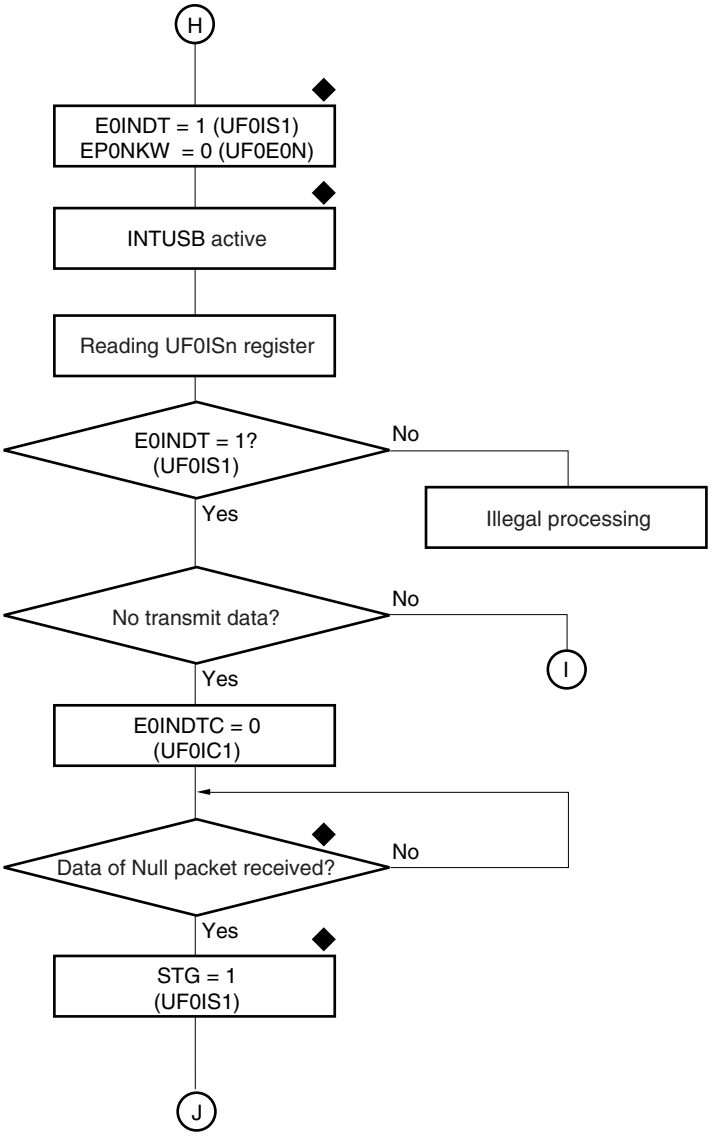
(b) Control transfer (read) (2/4)



Remark ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (5/12)

(b) Control transfer (read) (3/4)

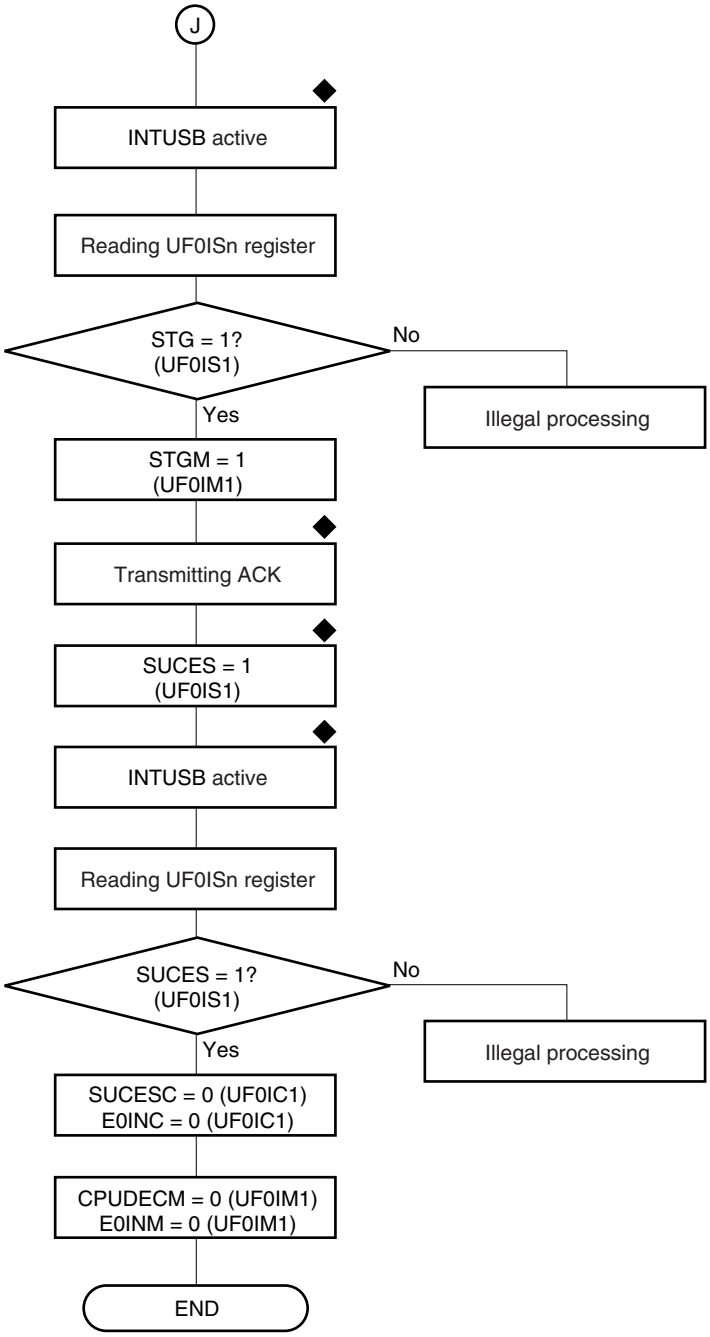


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (6/12)

(b) Control transfer (read) (4/4)

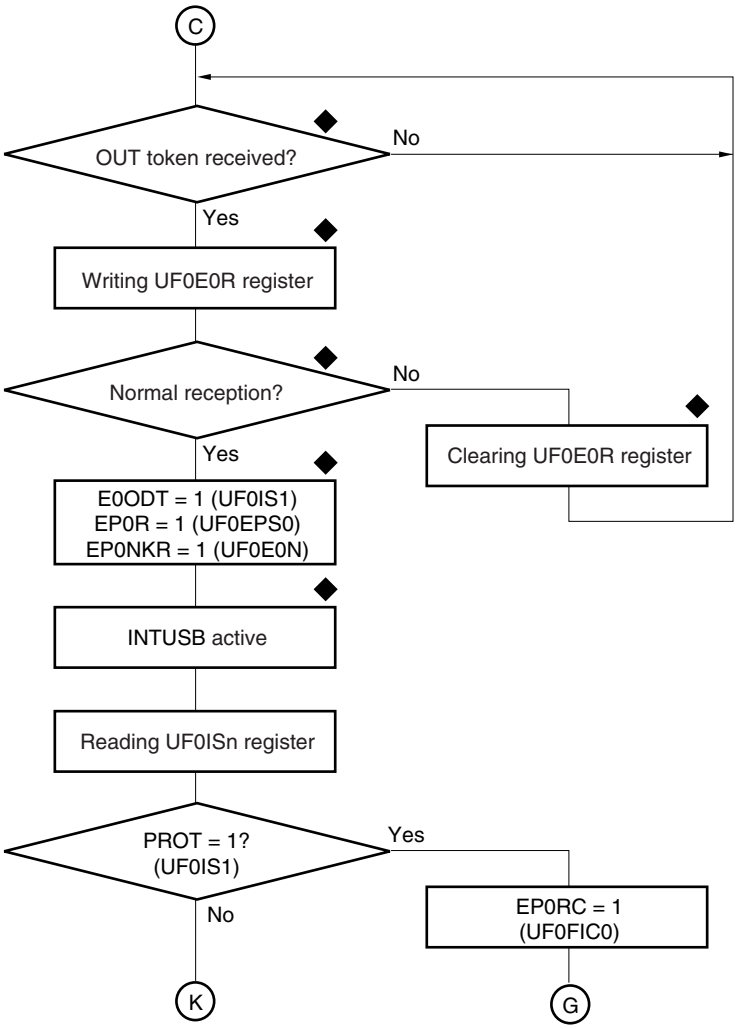


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (7/12)

(c) Control transfer (write) (1/4)

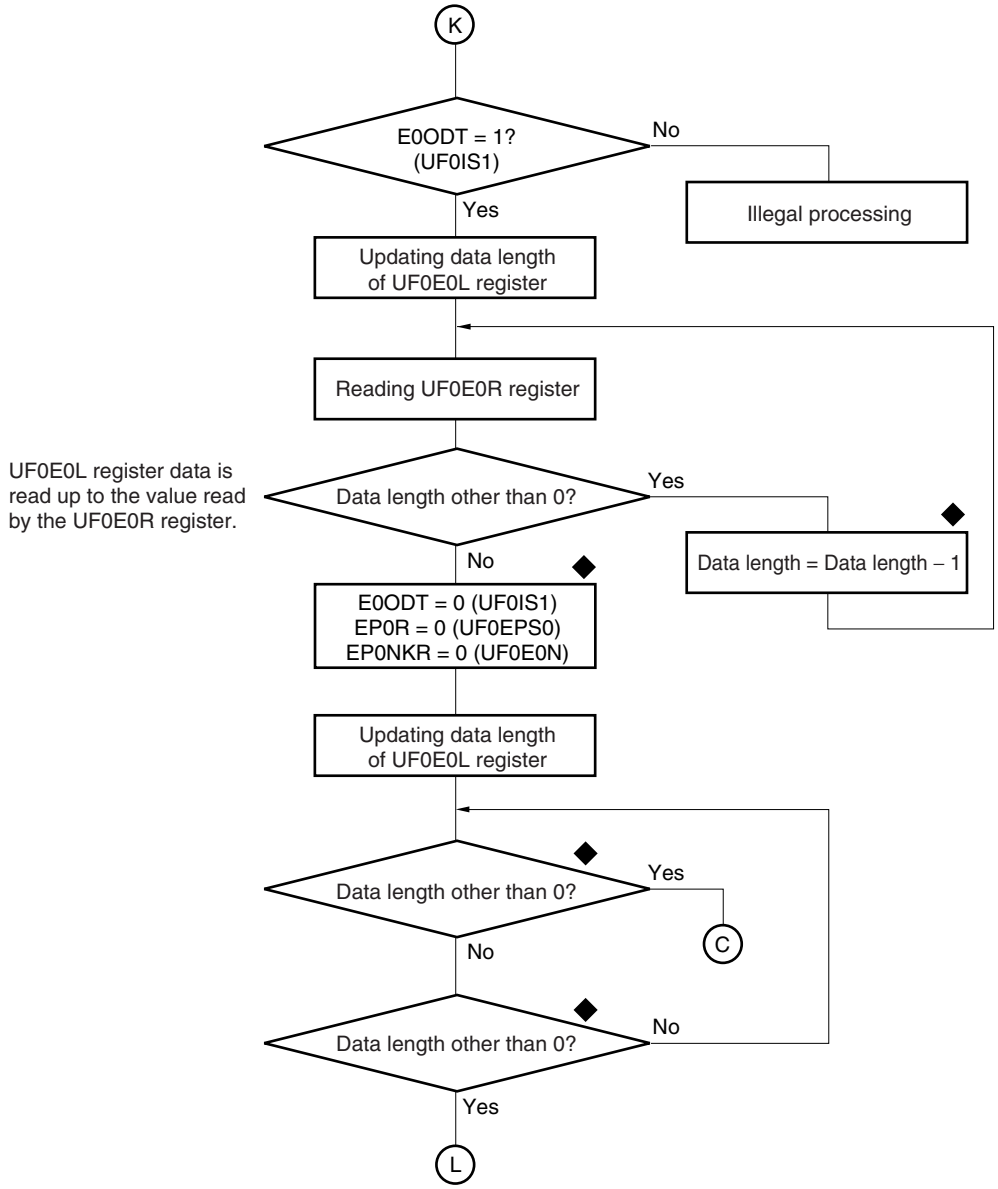


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (8/12)

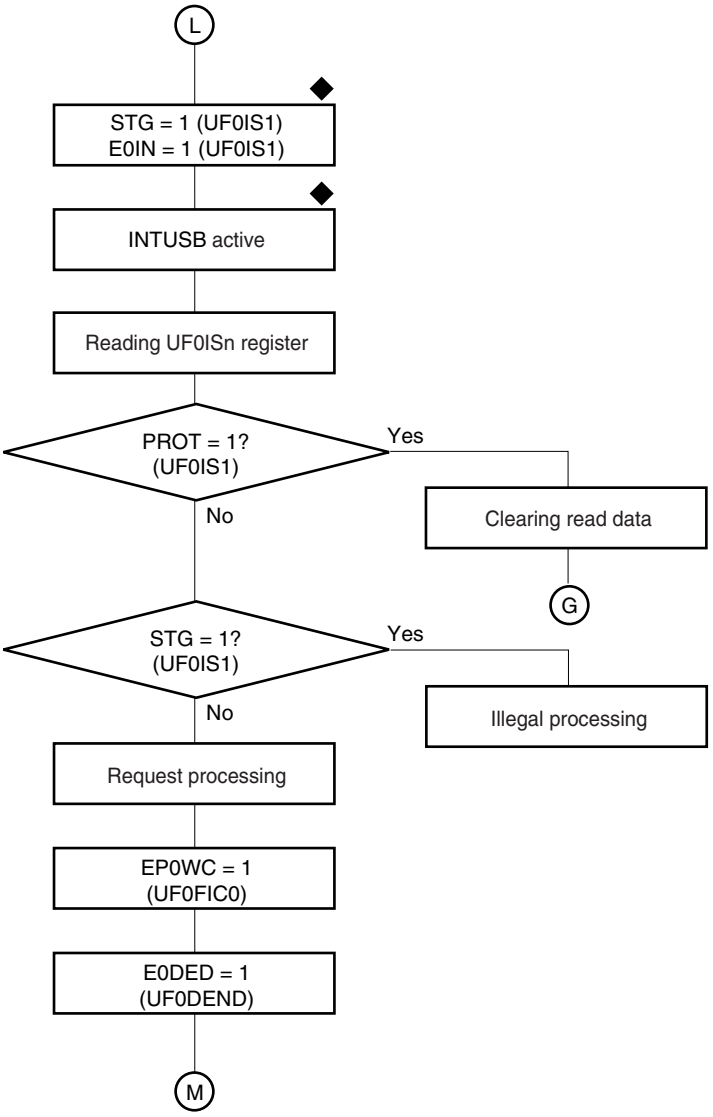
(c) Control transfer (write) (2/4)



Remark ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (9/12)

(c) Control transfer (write) (3/4)

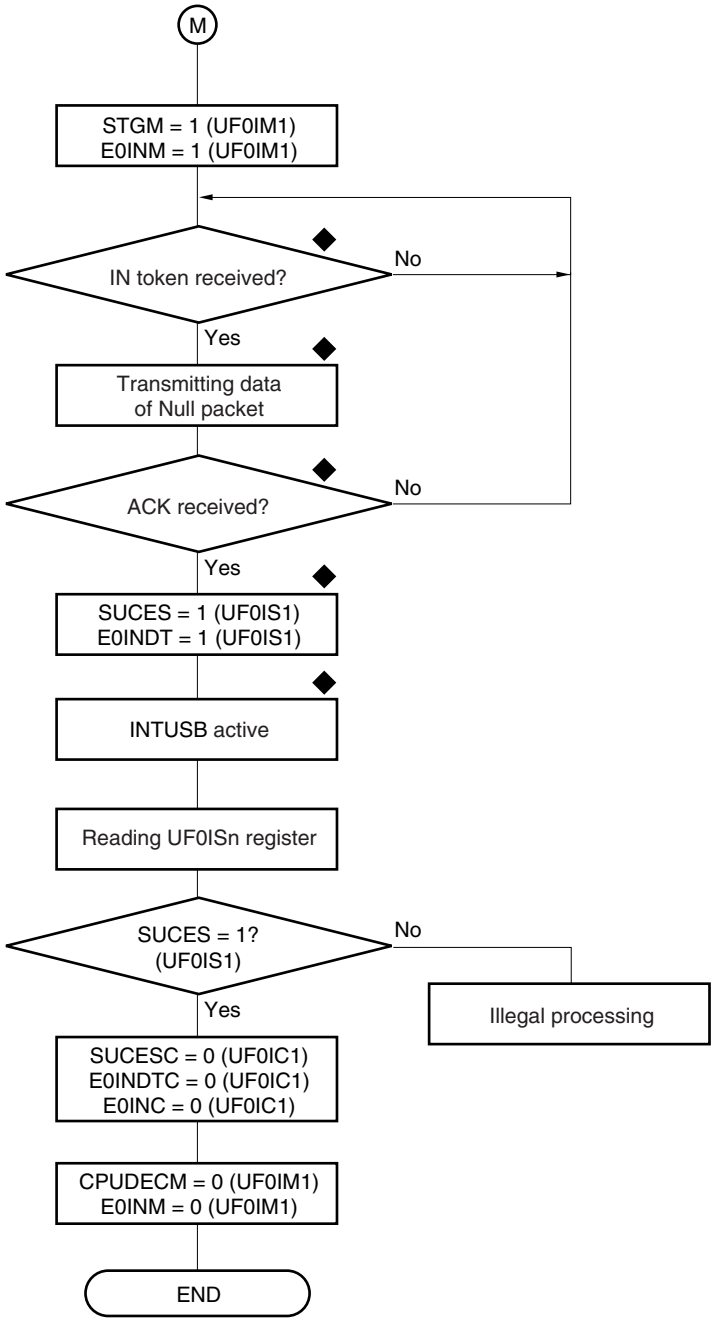


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (10/12)

(c) Control transfer (write) (4/4)

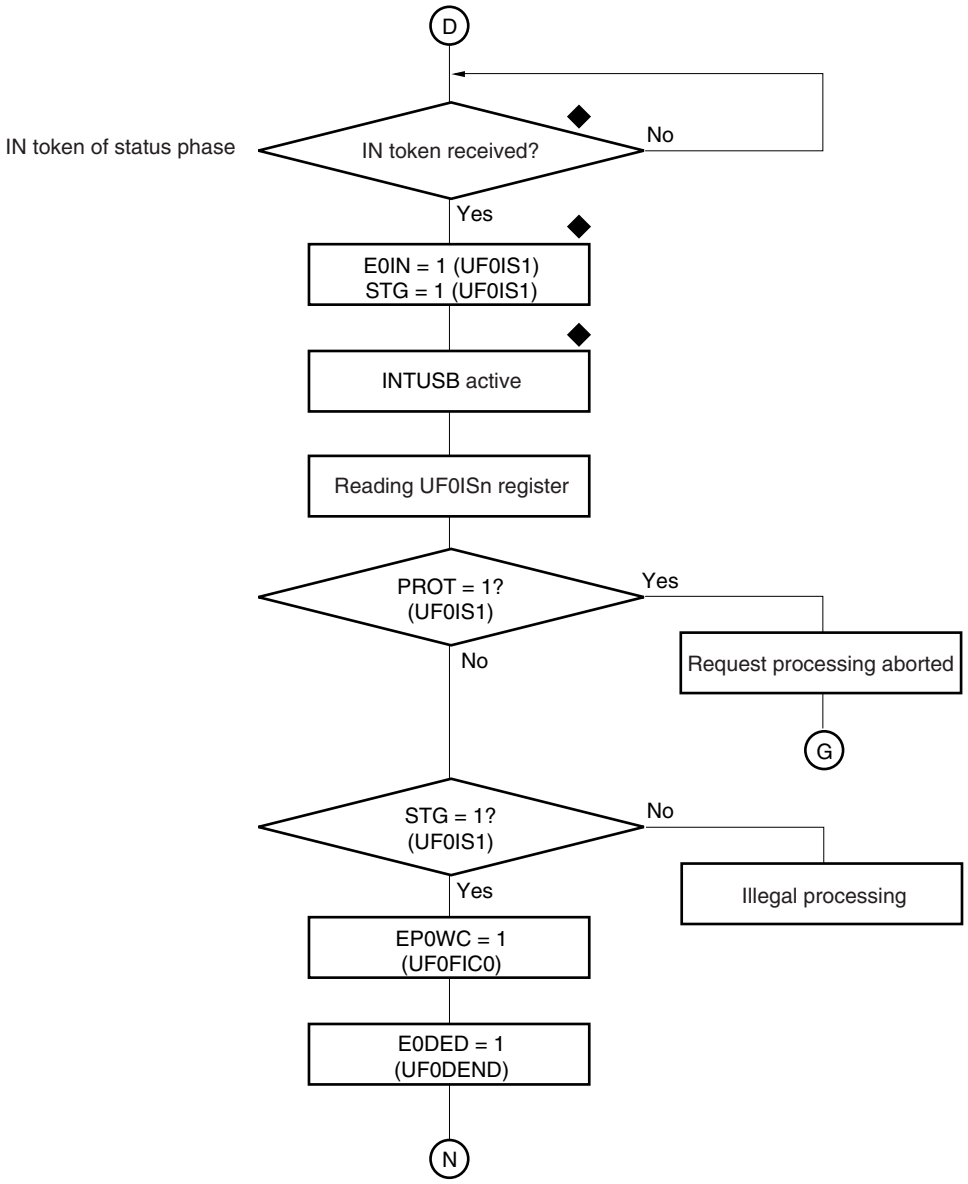


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (11/12)

(d) Control transfer (without data stage) (1/2)

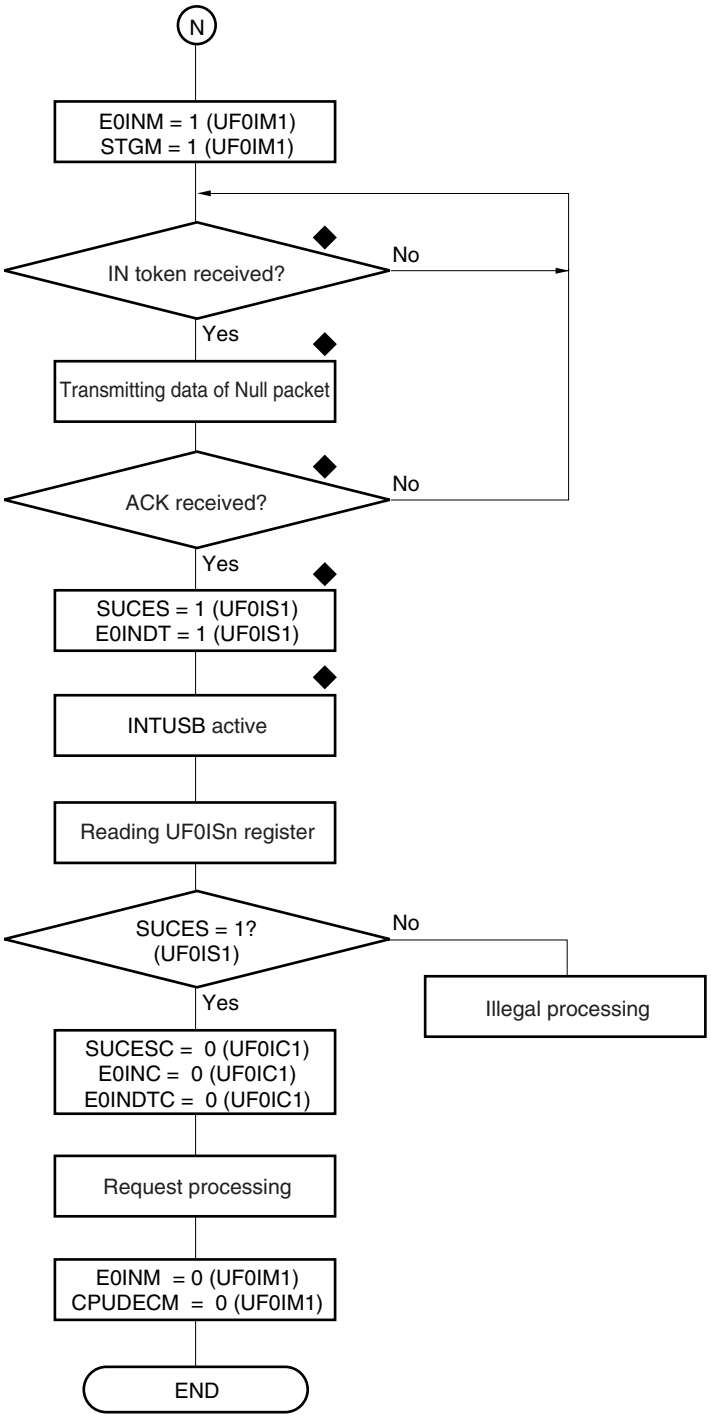


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-23. CPUDEC Request for Control Transfer (12/12)

(d) Control transfer (without data stage) (2/2)



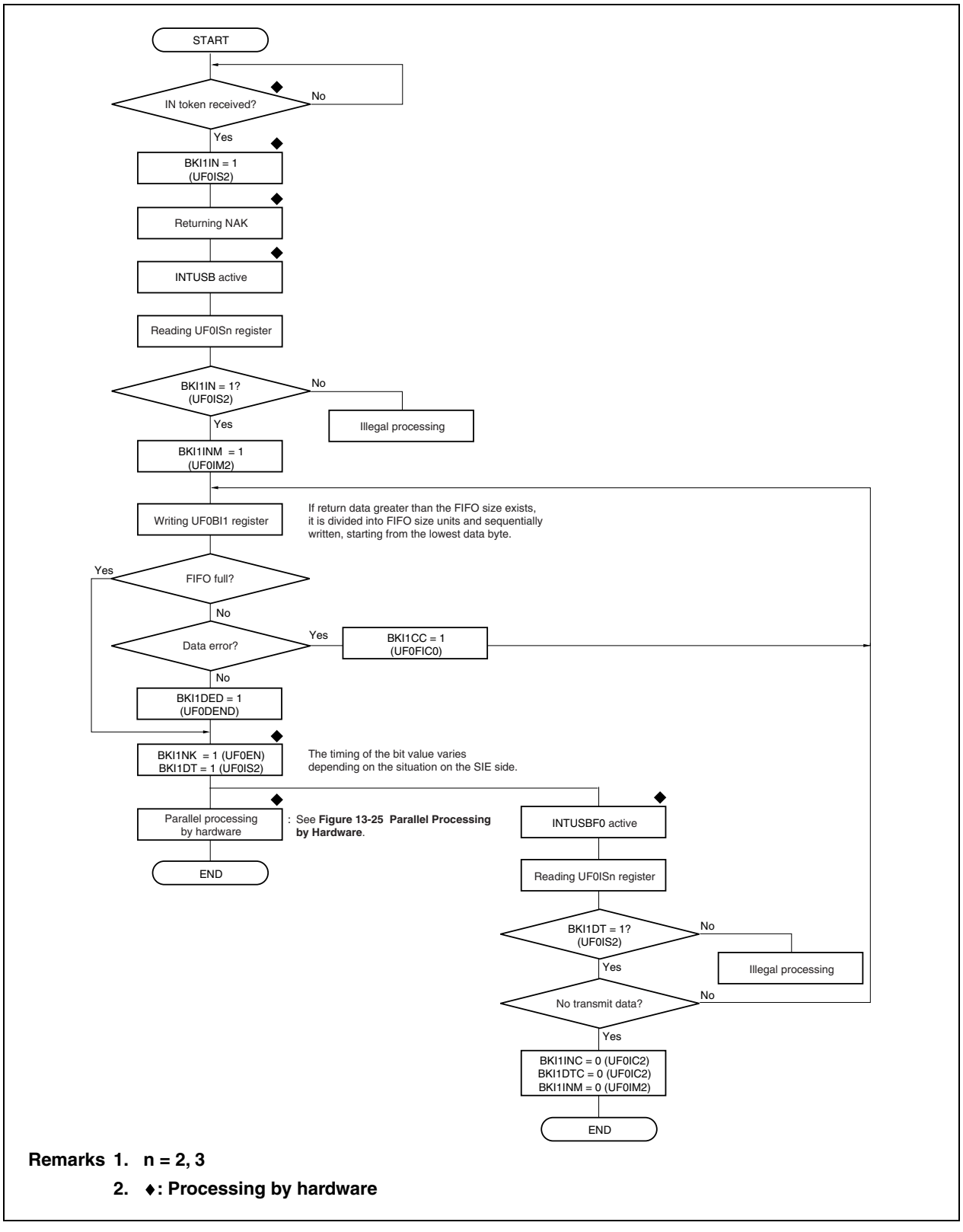
Remarks 1. n = 0, 1

2. ♦: Processing by hardware

(4) Processing for bulk transfer (IN)

Bulk transfer (IN) is allocated to Endpoint1 and Endpoint3. The flowchart shown below illustrates how Endpoint1 is controlled. Endpoint3 can also be controlled in the same sequence. To use this flowchart as the control flow of Endpoint3, therefore, read the bit names of Endpoint1 in the flowchart as those of Endpoint3.

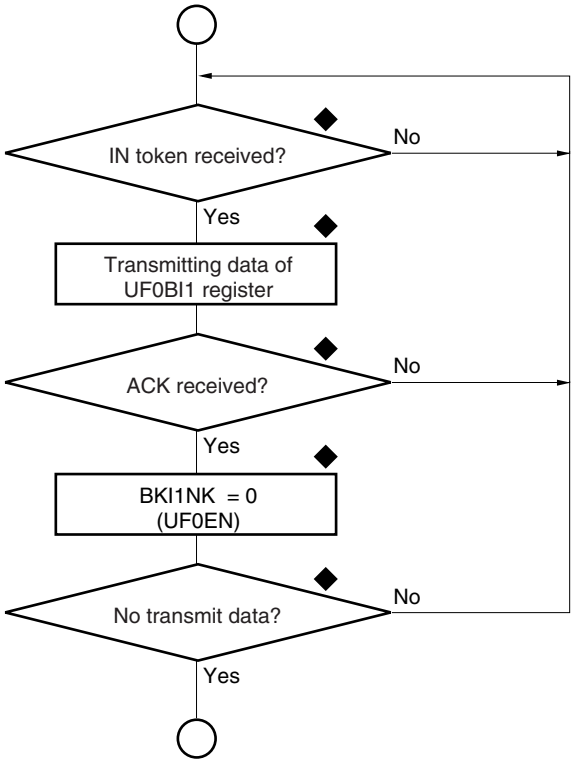
Figure 13-24. Processing for Bulk Transfer (IN) (Endpoint1)



Remarks 1. n = 2, 3

2. ♦: Processing by hardware

Figure 13-25. Parallel Processing by Hardware

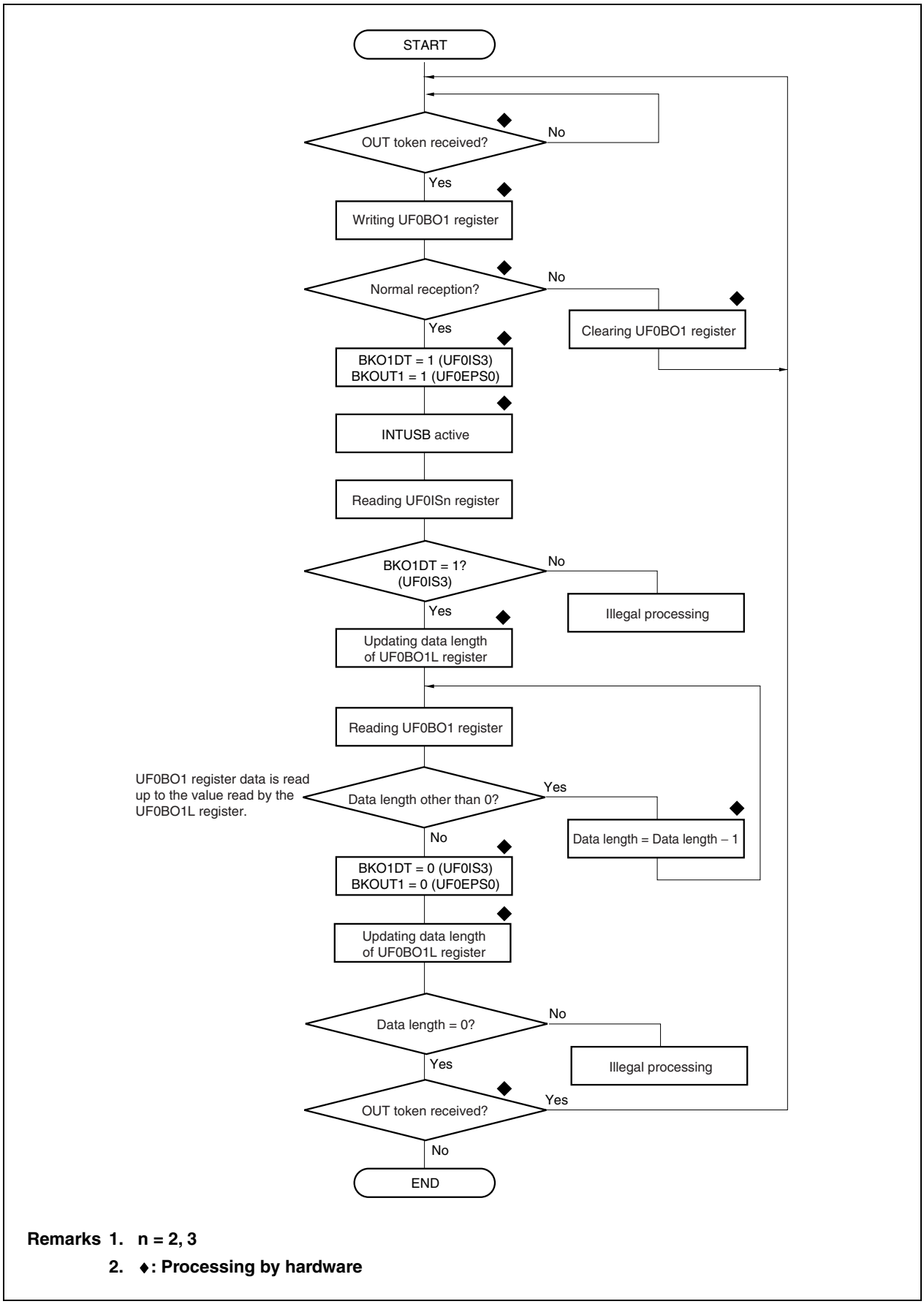


Remark ♦: **Processing by hardware**

(5) Processing for bulk transfer (OUT)

Bulk transfer (OUT) is allocated to Endpoint2 and Endpoint4. The flowchart shown below illustrates how Endpoint2 is controlled. Endpoint4 can also be controlled in the same sequence. To use this flowchart as the control flow of Endpoint4, therefore, read the bit names of Endpoint2 in the flowchart as those of Endpoint4.

Figure 13-26. Normal Processing for Bulk Transfer (OUT) (Endpoint2)

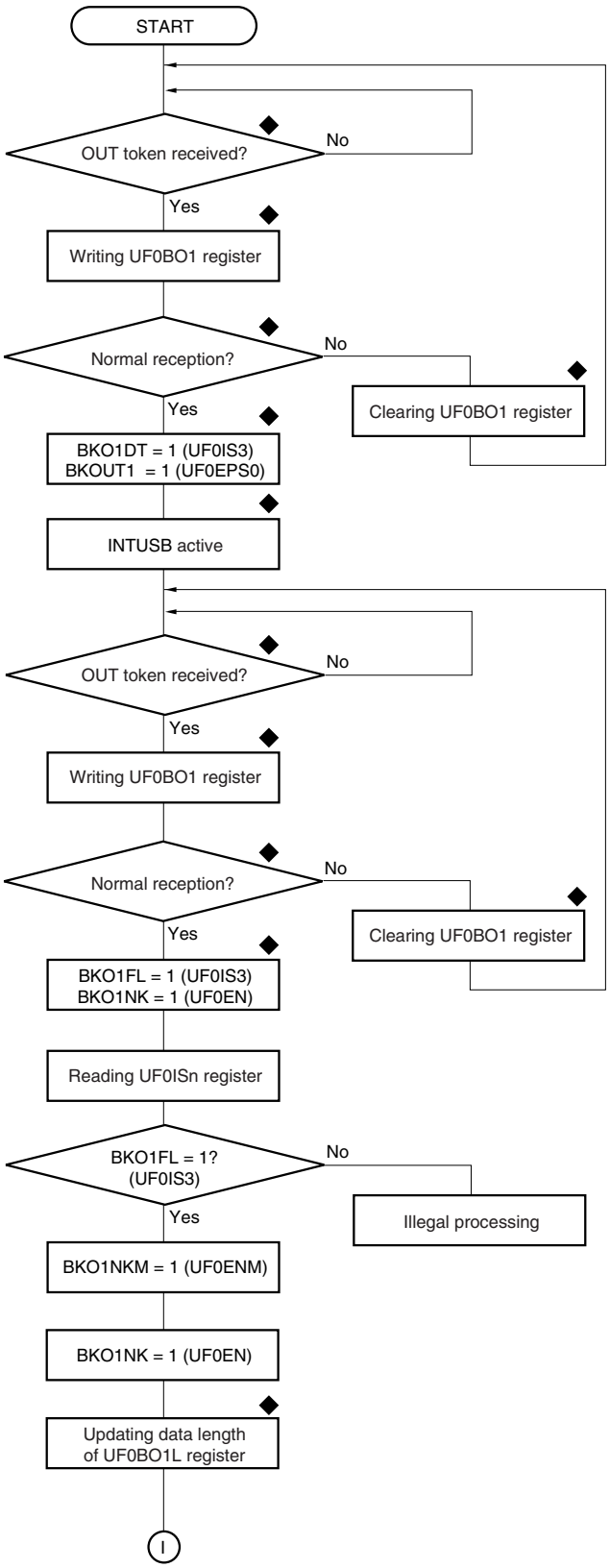


Remarks 1. n = 2, 3

2. ♦: Processing by hardware

During bulk transfer (OUT), more data may be transmitted from the host than expected by the system. Endpoint2 and Endpoint4 for bulk transfer (OUT) of the 78K0R/KC3-L, KE3-L consist of two 64-byte buffers so that NAK responses are suppressed as much as possible and data can be read from the CPU side even while the bus side is being accessed as the transfer rate of the USB bus increases. Consequently, if the host sends more data than expected by the system, up to 128 bytes of extra data may be automatically received in the worst case. In this case, change the control flow from that of the normal processing of Endpoint2 and Endpoint4 to the flow illustrated below when the quantity of data expected by the system has decreased to two packets. This flowchart illustrates how Endpoint2 is controlled. Endpoint4 can also be controlled in the same sequence. To use this flowchart as the control flow of Endpoint4, therefore, read the bit names of Endpoint2 in the flowchart as those of Endpoint4.

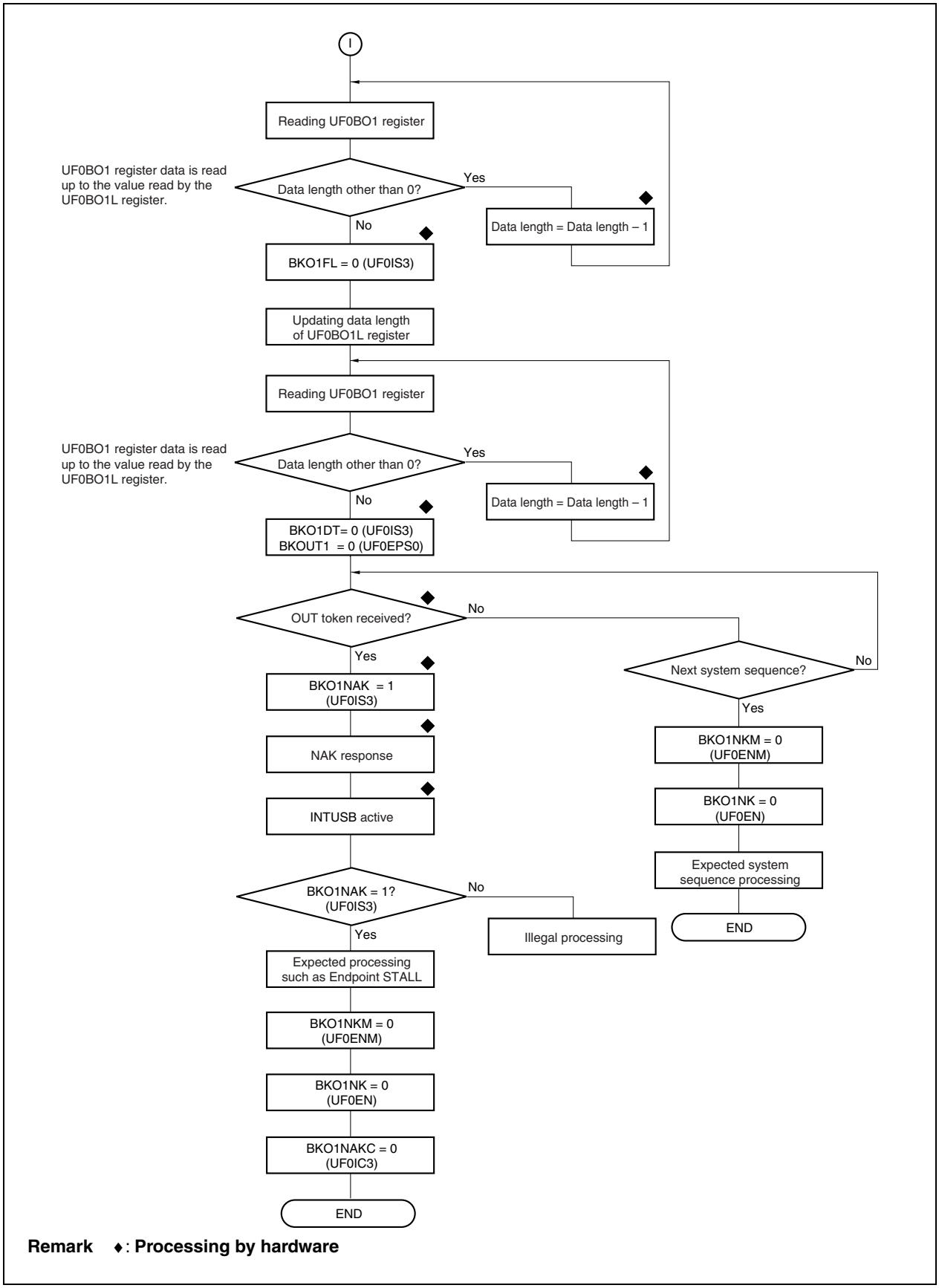
Figure 13-27. Processing If More Data Than Expected by System Is Transmitted (Endpoint2) (1/2)



Remarks 1. n = 2, 3

2. ♦: Processing by hardware

Figure 13-27. Processing If More Data Than Expected by System Is Transmitted (Endpoint2) (2/2)

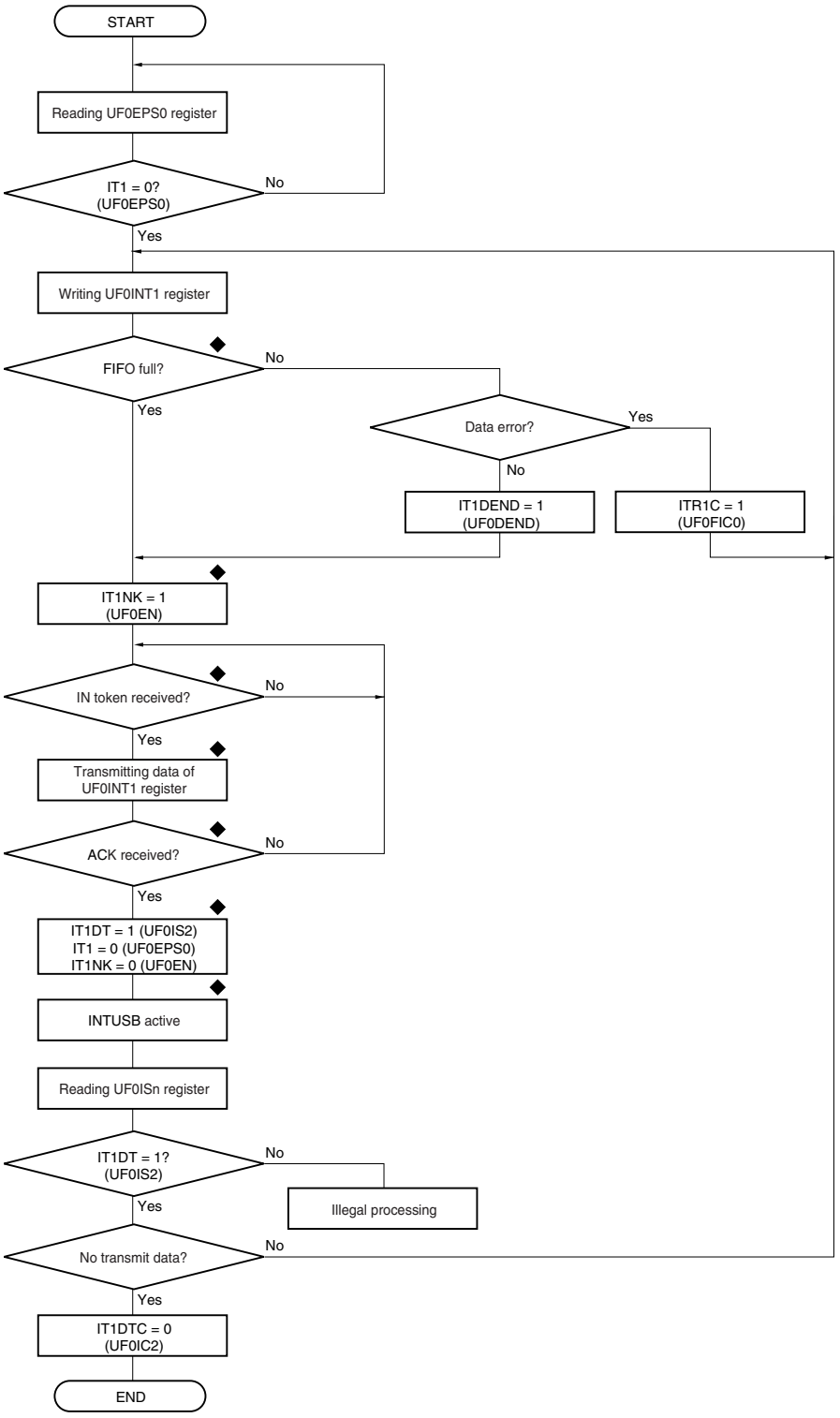


Remark ♦: Processing by hardware

(6) Processing for interrupt transfer (IN)

Interrupt transfer (IN) is allocated to Endpoint7. The flowchart is shown in **Figure 13-28**.

Figure 13-28. Processing for Interrupt Transfer (IN) (Endpoint7)



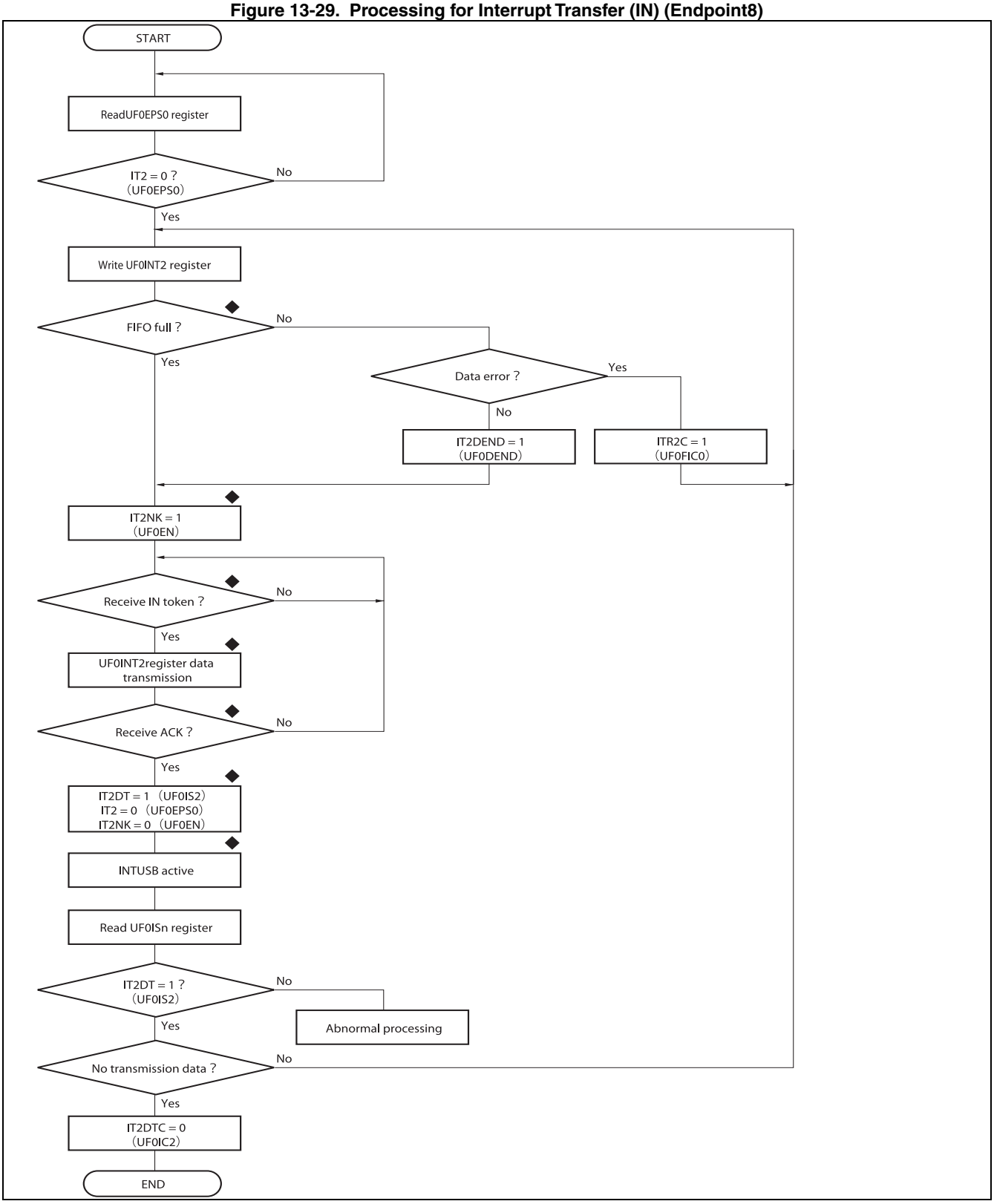
Remarks 1. n = 2, 3

2. ♦: Processing by hardware

(7) Processing for interrupt 2 transfer (IN)

Interrupt 2 transfer (IN) is allocated to Endpoint8. The flowchart is shown in **Figure 13-29**.

Figure 13-29. Processing for Interrupt Transfer (IN) (Endpoint8)



Remarks 1. n = 2, 3

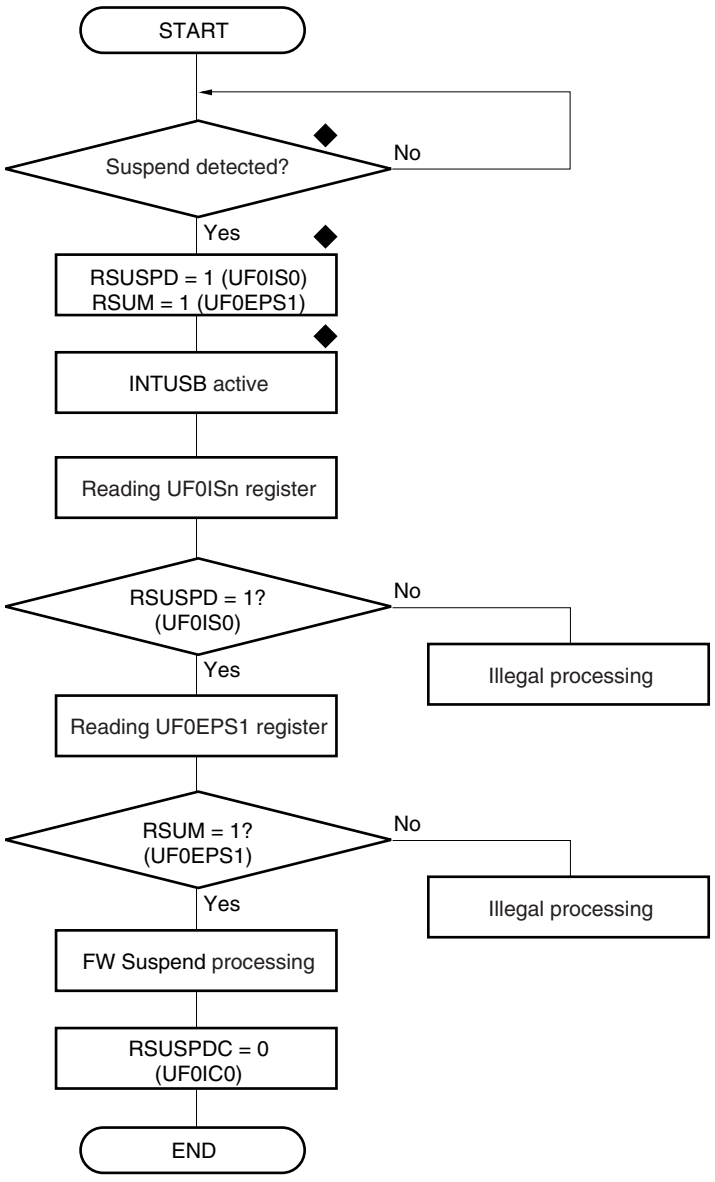
2. ♦: Processing by hardware

13.9.4 Suspend/Resume processing

How Suspend/Resume processing is performed differs depending on the configuration of the system. One example is given below.

Figure 13-30. Example of Suspend/Resume Processing (1/3)

(a) Example of Suspend processing

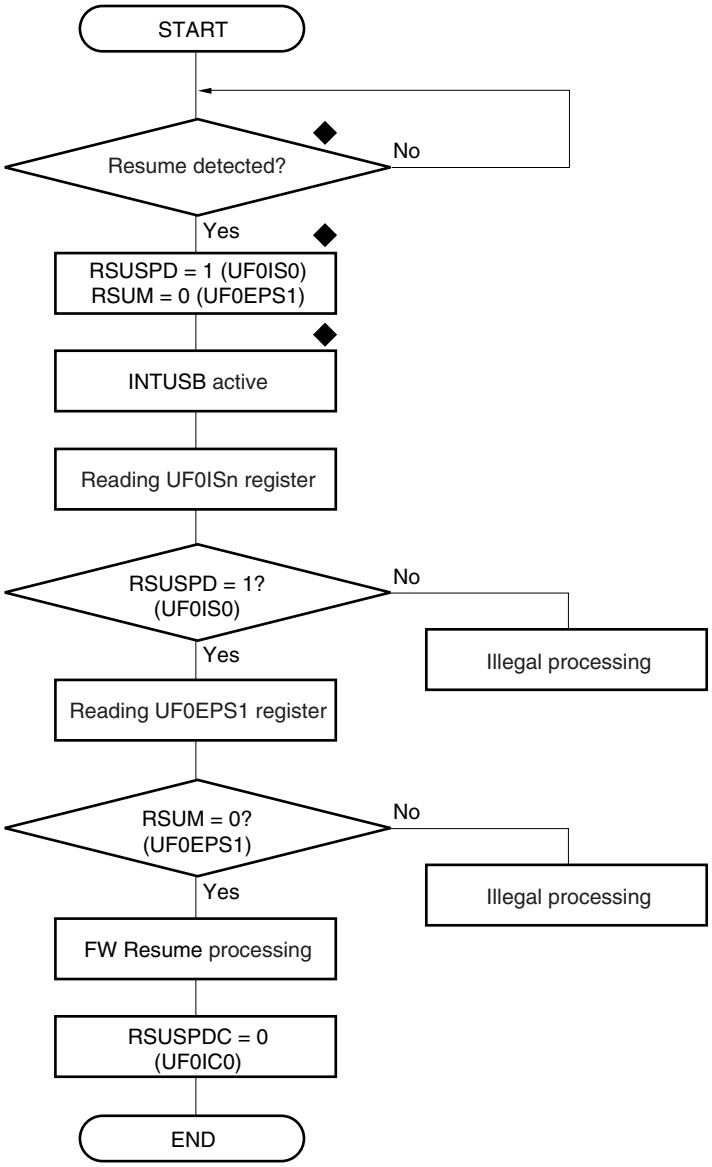


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-30. Example of Suspend/Resume Processing (2/3)

(b) Example of Resume processing

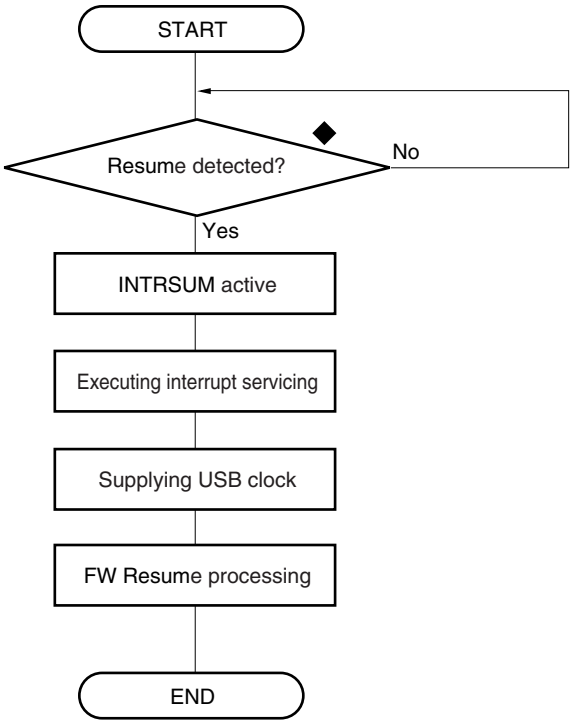


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-30. Example of Suspend/Resume Processing (3/3)

(c) Example of Resume processing (when supply of USB clock to USBF is stopped)



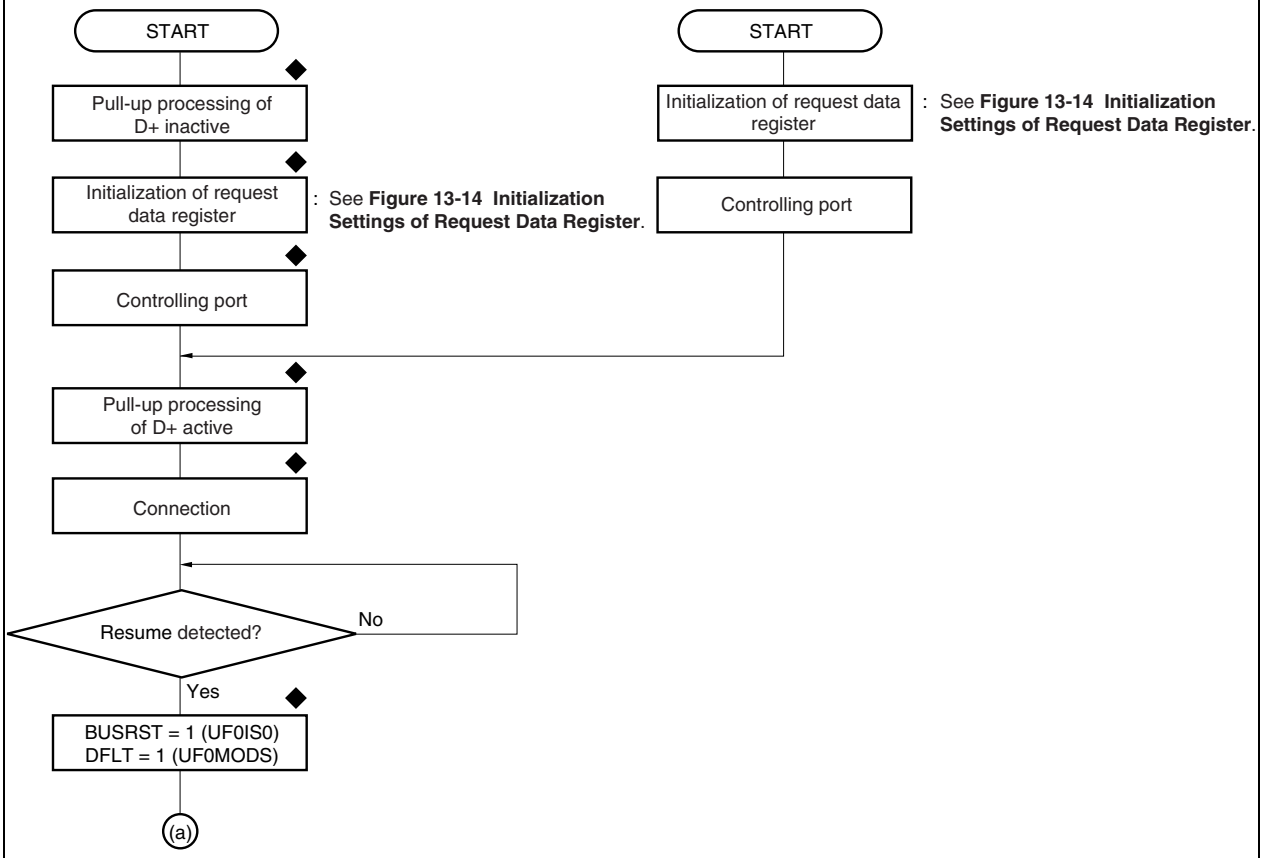
Remark ♦: Processing by hardware

13.9.5 Processing after power application

The processing to be performed after power application differs depending on the configuration of the system. One example is given below.

Figure 13-31. Example of Processing After Power Application/Power Failure (1/3)

(a) Processing after power application (1/2)



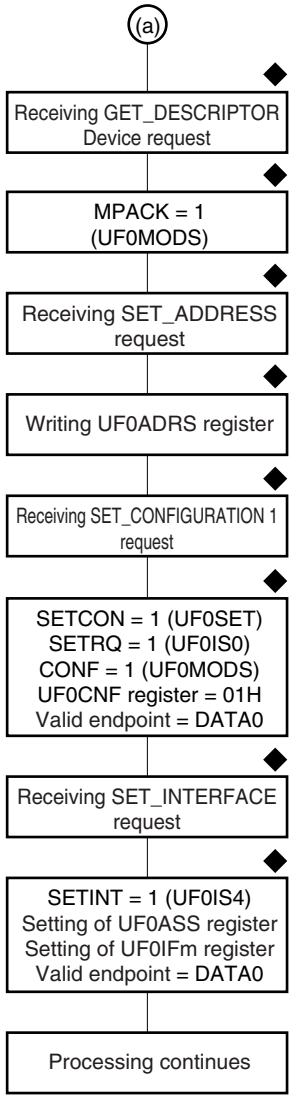
Notes 1. Use one general-purpose port pin for the signal that controls switching of the pull-up resistor of the USB bus.

2. The input mode or control mode of the general-purpose port pin allocated in Note 1 may be selected as the default value. Note the active level of pull-up processing of D+ on power application.

Remark ♦: Processing by hardware

Figure 13-31. Example of Processing After Power Application/Power Failure (2/3)

(a) Processing after power application (2/2)

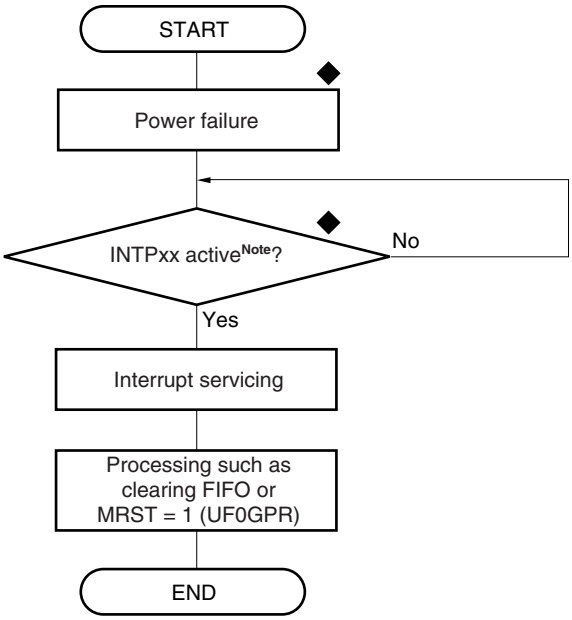


Remarks 1. m = 0 to 4

2. ♦: Processing by hardware

Figure 13-31. Example of Processing After Power Application/Power Failure (3/3)

(b) Processing on power failure



Note INTPxx indicates the external interrupt pins of the 78K0R/KC3-L, KE3-L (INTP00 - INTP2, INTP4 - INTP6, INTP8 - INTP11), and also indicates interrupts input by the external trigger pins (TI0n) of the timer.
Allocate one external interrupt pin to the following applications.

- **Detecting disconnection of the connector in the case of self-powered mode (SFPW bit of UF0DSTL register = 1). In this case, monitor the VDD line of the USB connector, and input the result to the external interrupt pin at the edge. Note that the noise elimination time is that of the interrupt input pin, and that of each timer.**
- **Detecting turning off power from the HUB when the device is mounted on the same board as a HUB chip.**

Remark 1. ♦: Processing by hardware
2. In case of 78K0R/KC3-L
n: Channel number (n= 1-3)
In case of 78K0R/KE3-L
n: Channel number (n= 0-4)

CHAPTER 14 MULTIPLIER/DIVIDER

14.1 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)

14.2 Configuration of Multiplier/Divider

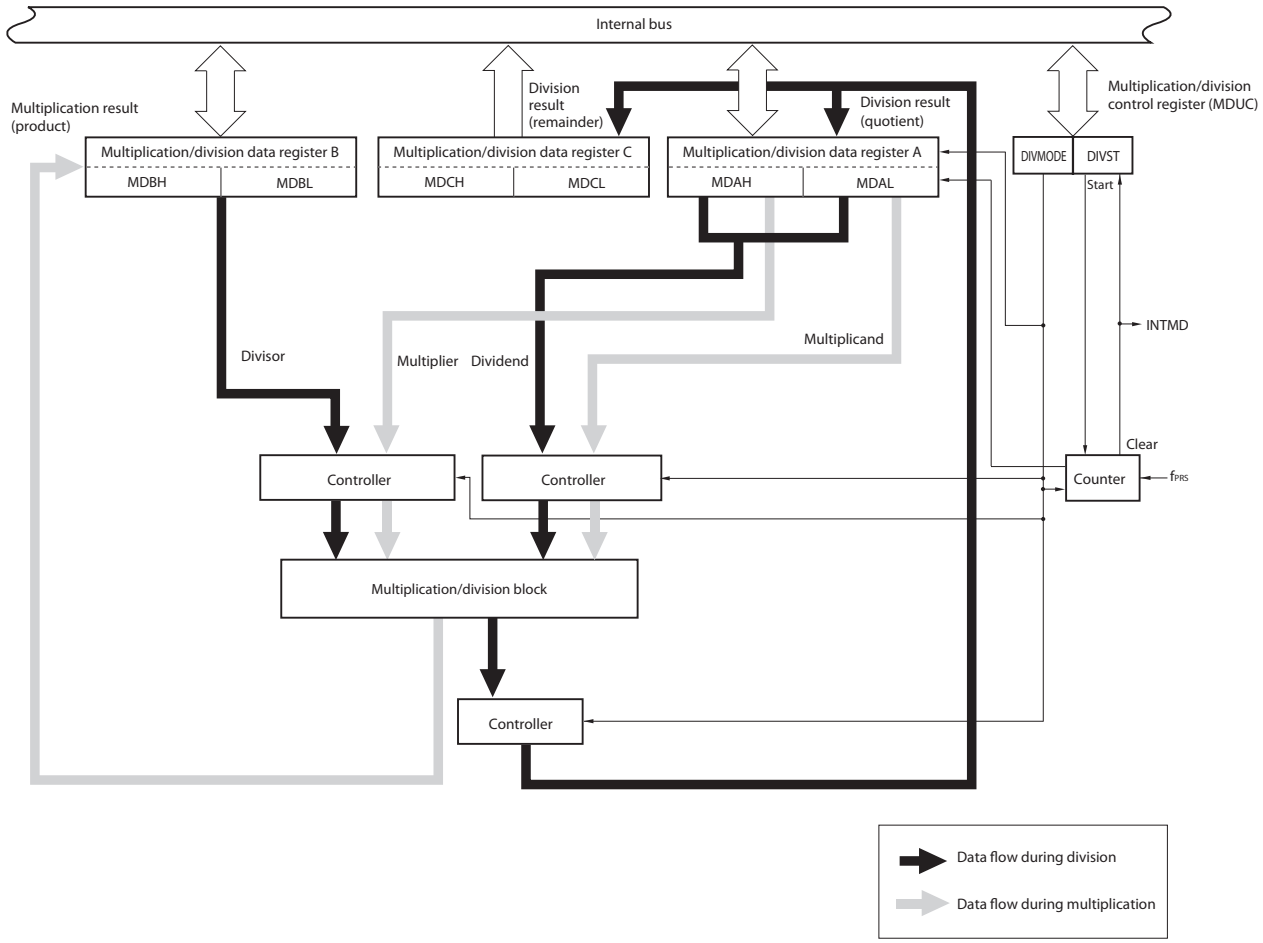
The multiplier/divider consists of the following hardware.

Table 14-1. 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 14-1 shows a block diagram of the multiplier/divider.

Figure 14-1 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 14-2. 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	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Symbol	FFFF1H								FFFF0H							
MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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 14-2. 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 14-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF7H								FFFF6H							
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	FFFF5H								FFFF4H							
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	MDBH 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 14-3. 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 14-4. 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 14-4. 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)]

14.3 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 14-5. 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).

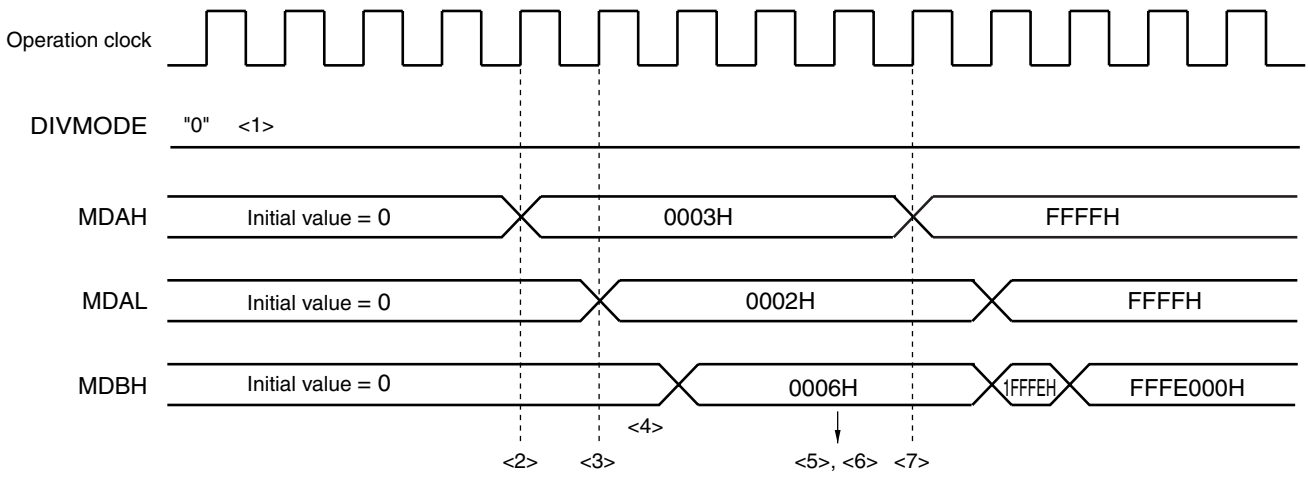
14.4 Operations of Multiplier/Divider

14.4.1 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 **14.4.2 Division operation**.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 14-6.

Figure 14-6. Timing Diagram of Multiplication Operation (0003H × 0002H)



14.4.2 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.

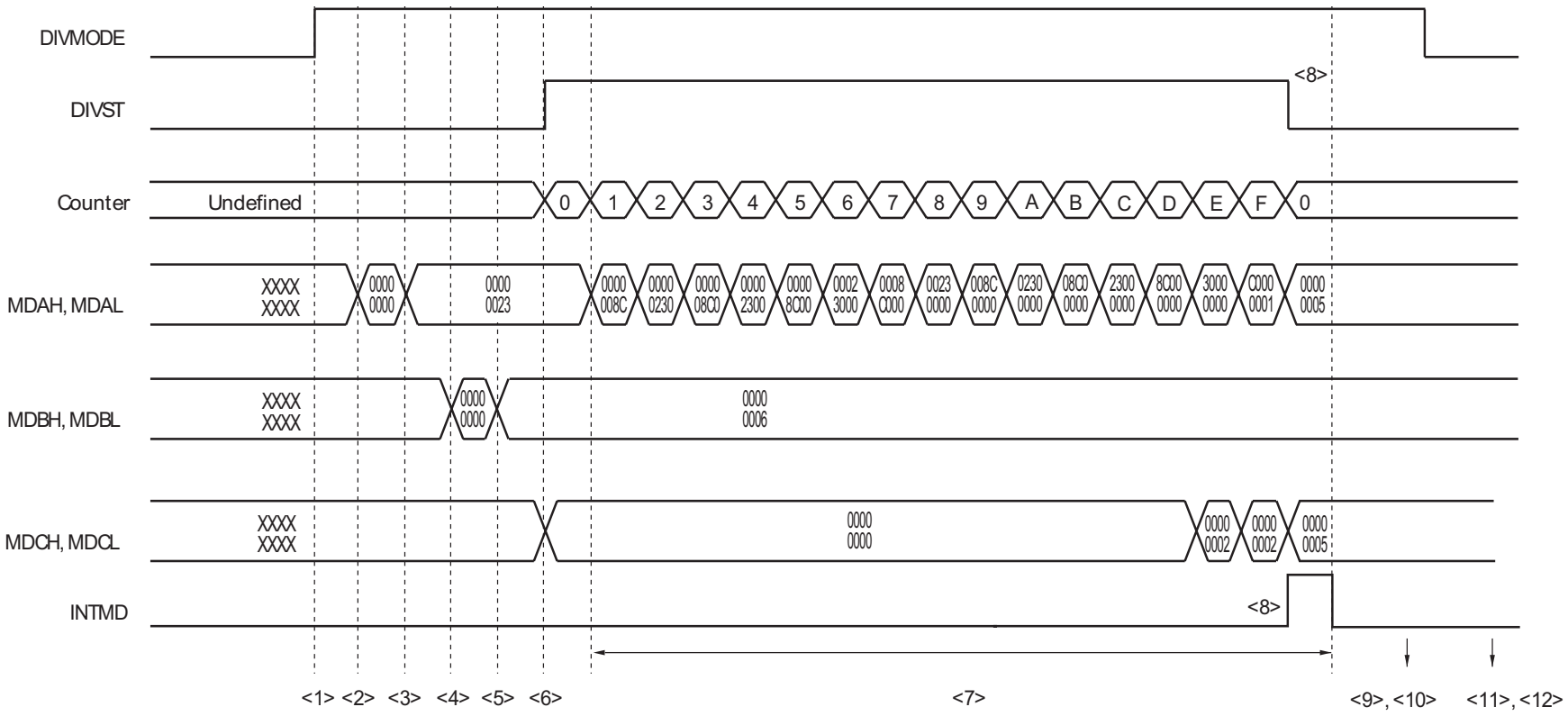
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether DIVST has been cleared

    (The read values of MDBL, MDBH, MDCH, and MDCL during operation processing are not guaranteed.)
- Operation end
  - <8> DIVST is cleared (0) (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 **14.4.1 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 14-7.

Figure 14-7. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 15 DMA CONTROLLER

The 78K0R/KC3-L, KE3-L has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between SFRs of the peripheral hardware supporting DMA 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.

15.1 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 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)
- Transfer target: Between SFR and internal RAM

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

Caution DMA does not support transfer of 2nd SFR and Internal RAM. Because of this, DMA transfer between internal RAM and 2nd SFR mapping register of USB connection is not possible.

15.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 15-1. 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 registers 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. Address FFFFEH, cannot be set due to the PMC register.

Figure 15-1. 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 (FE6FFH - FFEDFH in the case of the μPD78F1022) 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.

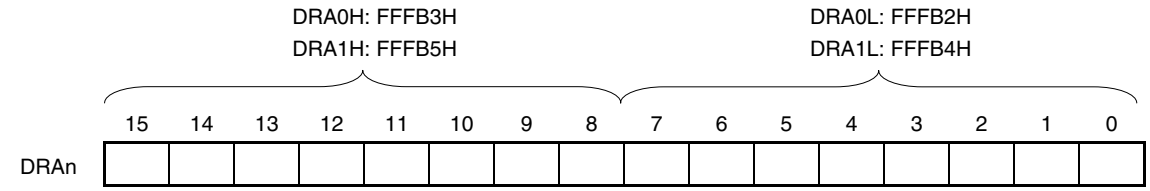
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

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 15-2. 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 15-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W

DBC0H: FFFB7H
DBC1H: FFFB9H

DBC0L: FFFB6H
DBC1L: FFFB8H

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

(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)

15.3 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 - 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 15-4. 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]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is done, once 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[Note2]	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.	

Note 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

2. In case of putting DMA transfer on hold in at the time of combined use of DMA and channel 2, be sure to put DMA of both channels on hold. (DWAIT0 = DWAIT1 = 1)

Remark n: DMA channel number (n = 0, 1)

Figure 15-4. 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	Counter end of timer channel 0 or capture interrupt
0	0	1	1	INTTM01	Counter end of timer channel 1 or capture interrupt
0	1	0	0	INTTM04	Counter end of timer channel 4 or capture interrupt
0	1	0	1	INTTM05	Counter end of timer channel 5 or capture interrupt
0	1	1	0	INTST0/INTCSI00	End of transfer of UART 0 transmission, transmit buffer interrupt / end of transfer of CSI00, transmit buffer interrupt
0	1	1	1	INTSR0	End of transfer of UART 0 reception interrupt
1	0	0	0	INTST1/INTCSI10/INTIIC10	End of transfer of UART 1 transmission, transmit buffer interrupt / end of transfer of CSI10, transmit buffer interrupt / end of transfer of IIC10 interrupt.
1	0	0	1	INTSR1	End of transfer of UART 1 reception interrupt
1	0	1	0	INTST3	End of transfer of UART3 transmit buffer interrupt
1	0	1	1	INTSR3	End of transfer of UART 3 reception 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 15-5. 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.
The DMA controller 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).
The DMA controller 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 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 15.5.7 Forced termination by software).

Remark n: DMA channel number (n = 0, 1)

15.4 Operation of DMA Controller

15.4.1 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, DBCn, and DMCn registers.

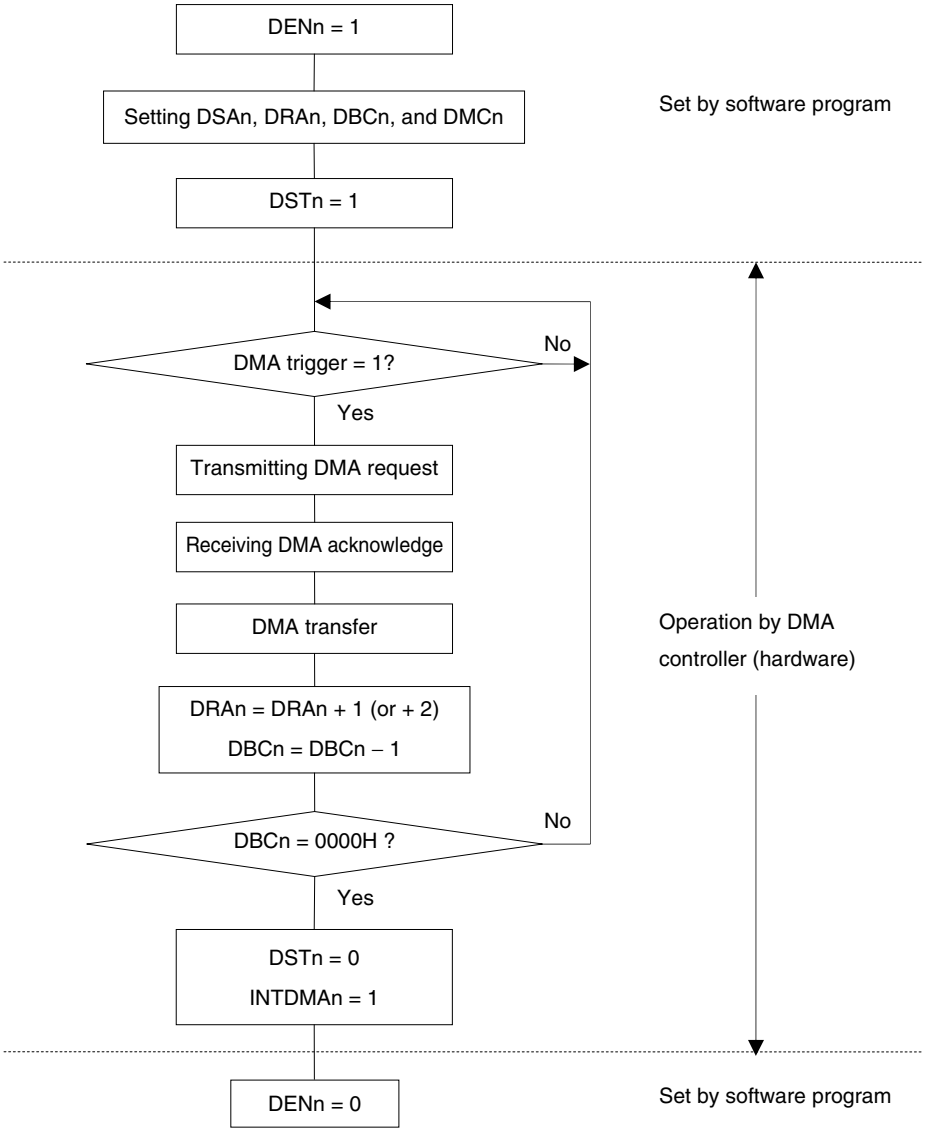
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<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 15-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

15.4.2 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.

15.4.3 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)

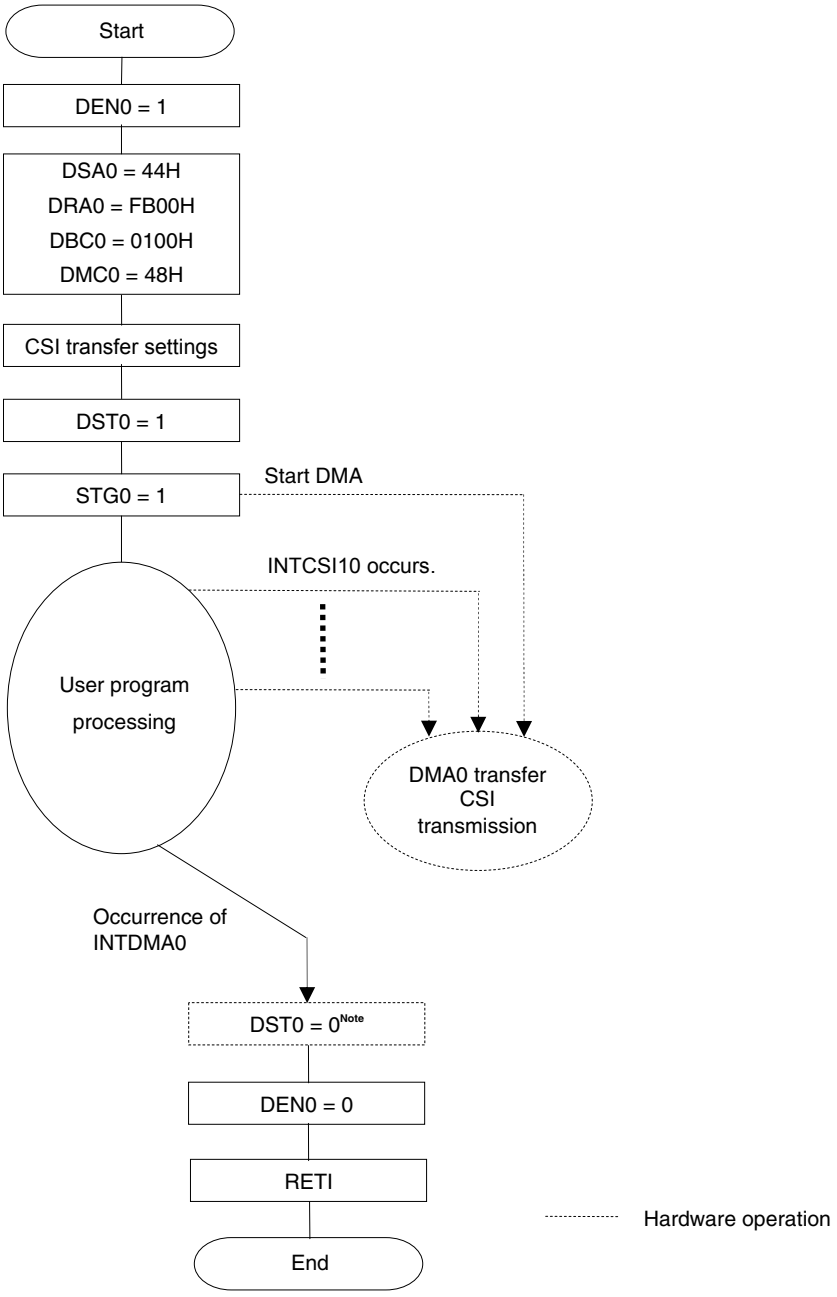
15.5 Example of Setting of DMA Controller

15.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI00 (256 Byte)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
- Interrupt of CSI00 is specified by IFC03 to IFC00 (bits 3 - 0 of the DMC0 register) = 0110B.
- Transfers FF100H to FF1FFH (256 bytes) of RAM to FFF10H of the transmit buffer (SIO00) of CSI.

Figure 15-7. 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 15.5.7 Forced termination by software).

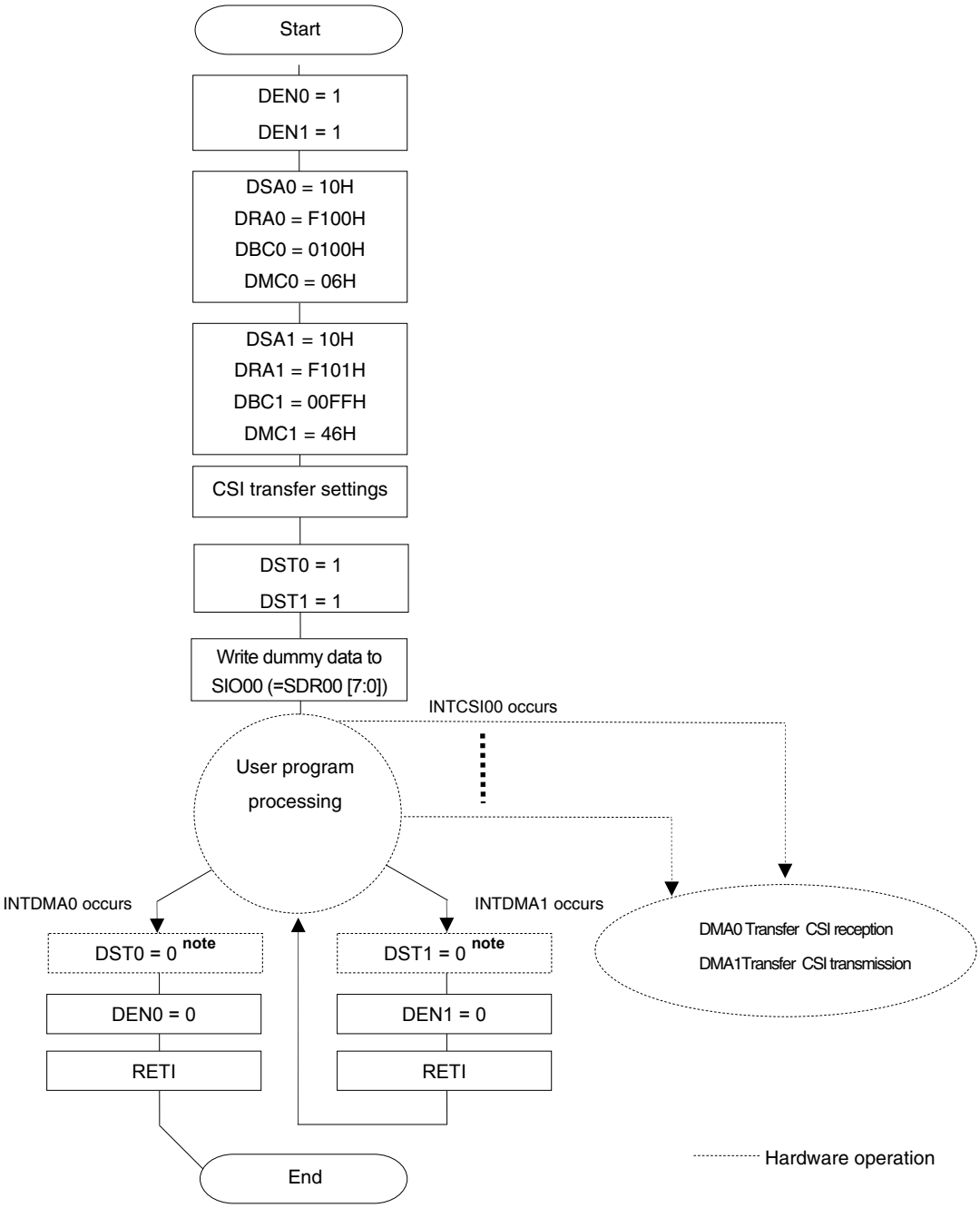
The fist trigger for consecutive transmission is not started by the interrupt of CSI. It starts by a software trigger.
CSI transmission of the second time and onward is automatically executed.
The DMA interrupt (INTDMA0) is generated as soon as the last data of data register has been written to the transmission buffer. At this point, the last data of CSI is being transmitted. To start DMA transfer again, therefore, wait until transfer of CSI is completed.

15.5.2 CSI master reception

The following flow chart shows the setting sample of CSI master reception.

- Reception of CSI00 (256 Byte)
- Use reception data of DMA channel 0 to read and channel 1 to write dummy data.
- Start up source of DMA: INTCSI00
  (When channel 0 and channel 1 is set to similar source of action, first transfer channel 0 and then transfer channel 1).
- CSI00 interrupt is assigned to IFC03-IFC00 = IFC13-IFC10 (Bit 3-0 of DMCn register) =0110B.
- Transfer (Reception) From FFF10H of data register of CSI00 to FF100H-FF1FFH (256 byte) of RAM.
  (For successive transmission/reception), data for initial buffer empty interrupt of reception data is disabled before the reception.
- Transfer dummy data FF101H-FF1FFH of RAM (255 byte) to CSI data register (SIO00)) of FFF10H .
  (Write dummy data to (instruction) for the 1st bit)

Figure 15-8. Example of setting for CSI master reception



Note The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Because writing is enabled for the DENn flag only when DSTn = 0, if a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, set DSTn to 0 and then DENn to 0 (for details, refer to 15.5.7 Forced termination by software).

In case of CSI master reception, CSI interrupt is not present when reception starts, therefore in this example, transmission data is written in software. Data reception transfer occurs automatically after 1st byte.

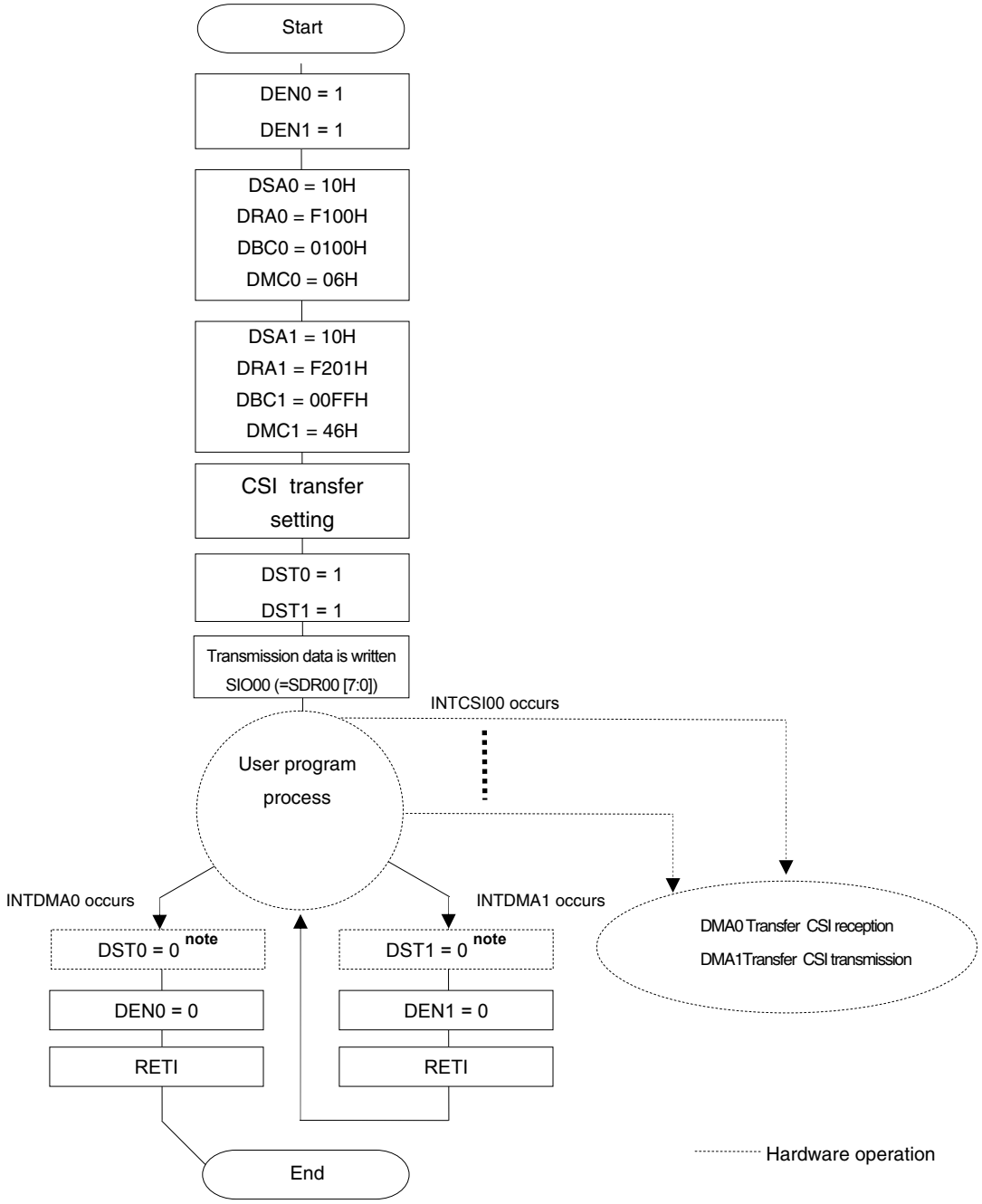
At the time of successive reception mode, initial buffer empty interrupt of reception data becomes disabled before reception of enabled data. At the point when writing of initial transmission data of data register completes, DMA interrupt (INTDMA1) occurs. At the point when reading of initial reception data of data register completes, DMA interrupt (INTDMA0) occurs. Please start the DMA transmission again after ending CSI transfer.

15.5.3 Transmission/reception of CSI

The following flow chart shows the setting example of CSI transmission/reception.

- Transmission/reception of CSI00 (256 Byte)
- Use DMA channel 0 to read reception data, channel 1 to write.
- Start up source of DMA: INTCSI00
  (When channel 0 and channel 1 is set to similar source of action, first channel 0 is transferred and channel 1 is transferred later.)
- CSI00 interrupt is assigned to IFC03-IFC00 = IFC13-IFC10 (Bit 3-0 of DMCn register) =0110B.
- Transfer (Reception) From FFF10H of data register of CSI to FF100H-FF1FFH (256 byte) of RAM.
  (For successive transmission/reception), data for initial buffer empty interrupt of reception data is disabled before reception.
- Transfer (transmission) of FFF10H of CSI data register (SI00) to FF201H-F2FFH of RAM (255 byte).
  (Transmission data of initial 1 bit is written to soft (instruction))

Figure 15-9. Example of setting for CSI transmission reception



Note The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Because writing is enabled for the DENn flag only when DSTn = 0, if a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, set DSTn to 0 and then DENn to 0 (for details, refer to 15.5.7 Forced termination by software).

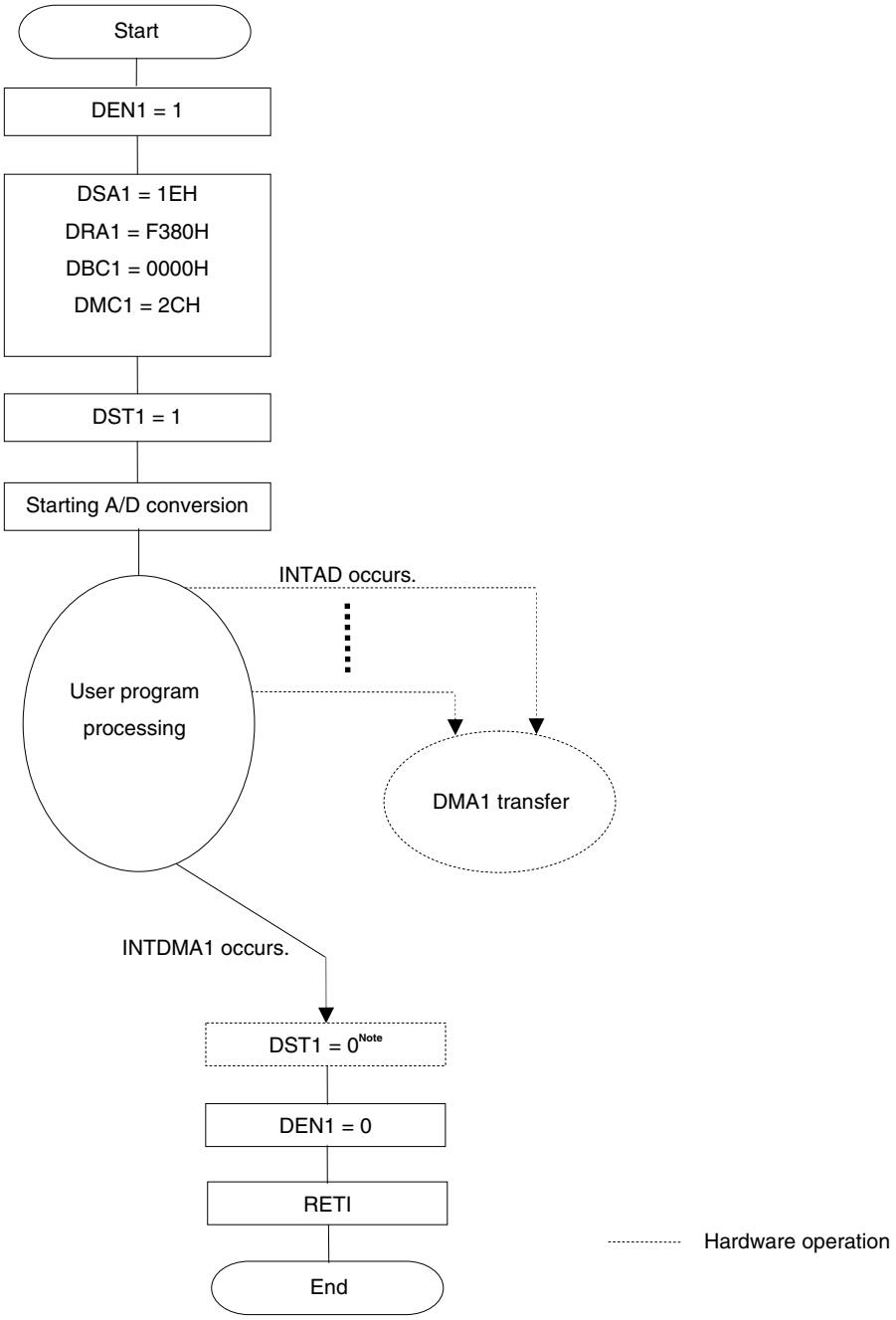
In case of CSI transmission reception, CSI interrupt is not present when 1st byte of transmission data is written. Therefore in this example, transmission reception data is written in software. Data transmission for 2nd byte onwards is transferred automatically.Data reception transfer occurs automatically after 1st byte.
(At the time of successive transmission reception mode, initial buffer empty interrupt of reception data is disabled before reception of enabled data.)
DMA interrupt (INTDMA1) is generated when writing of initial transmission data of data register completes. DMA interrupt (INTDMA0) is generated when reading of initial reception data of data register completes. Please start the DMA transmission after ending the CSI transfer.

15.5.4 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 10-bit A/D conversion result register to 2048 bytes of FF380H – FFB7FH of RAM.

Figure 15-10. Example of Setting of Consecutively Capturing A/D Conversion Results



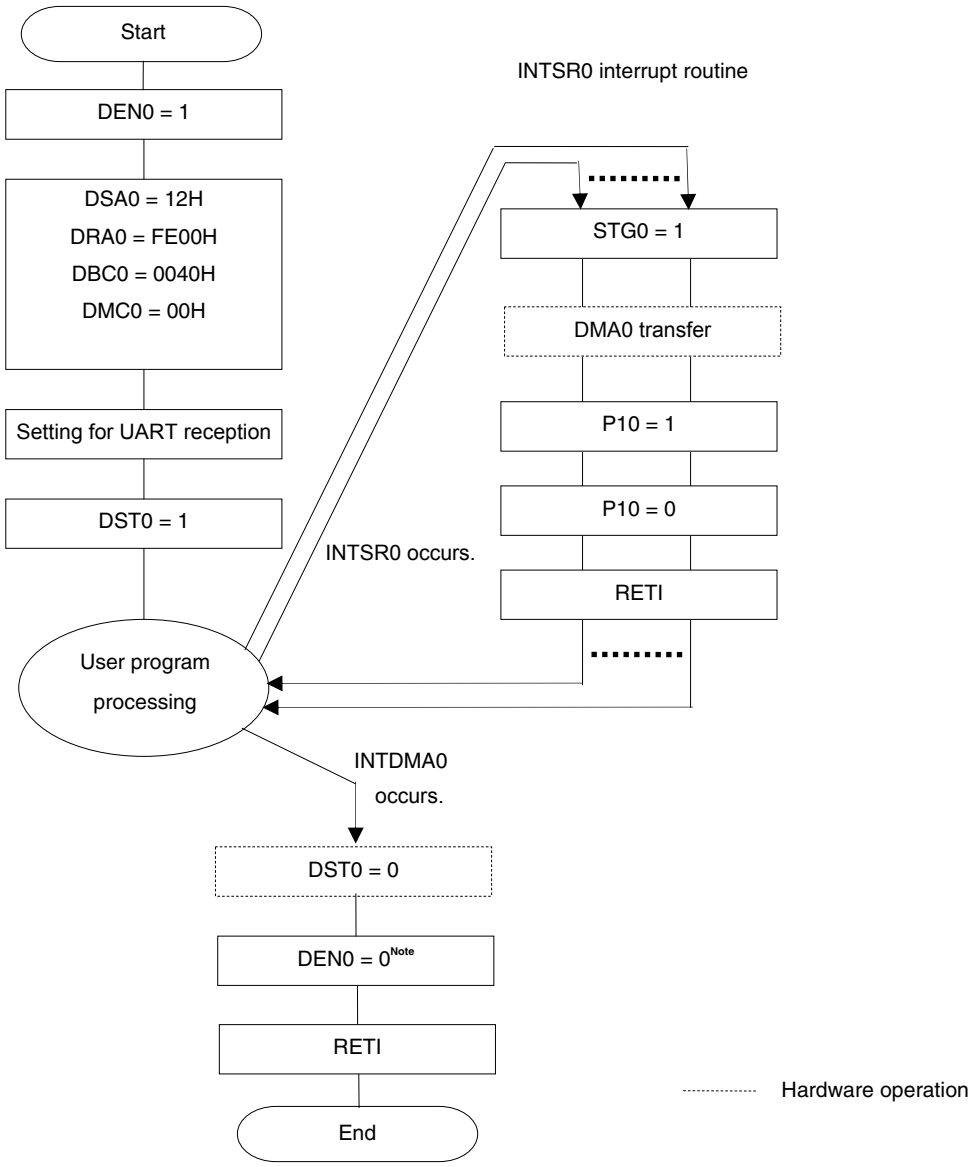
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to 15.5.7 Forced termination by software).

15.5.5 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 15-11. 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 15.5.7 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.

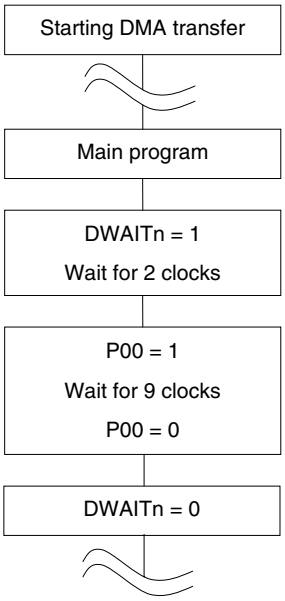
15.5.6 Holding DMA transfer pending by DWAITn

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 of the set system, a DMA transfer can be held pending by setting DWAITn to 1. DMA transfer of transfer trigger generated during hold is executed after hold release. However, as there is only 1 transfer trigger per channel that can be held, even if the transfer trigger for the same channel is generated twice during hold, DMA transfer is executed only once after hold release.

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 DWAITn to 1.

After setting DWAITn to 1, it takes two clocks until a DMA transfer is held pending.

Figure 15-12. Example of Setting for Holding DMA Transfer Pending by DWAITn



Note **In case of putting transfer of DMA on hold in between combined use of DMA and channel 2, be sure to put DMA of both the channels on hold. (DWAIT0 = DWAIT1 = 1). Hold may not occur if DMA transfer is executed when one of the DMA is in hold.**

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

15.5.7 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.

<IN case of using 1 channel for DMA>

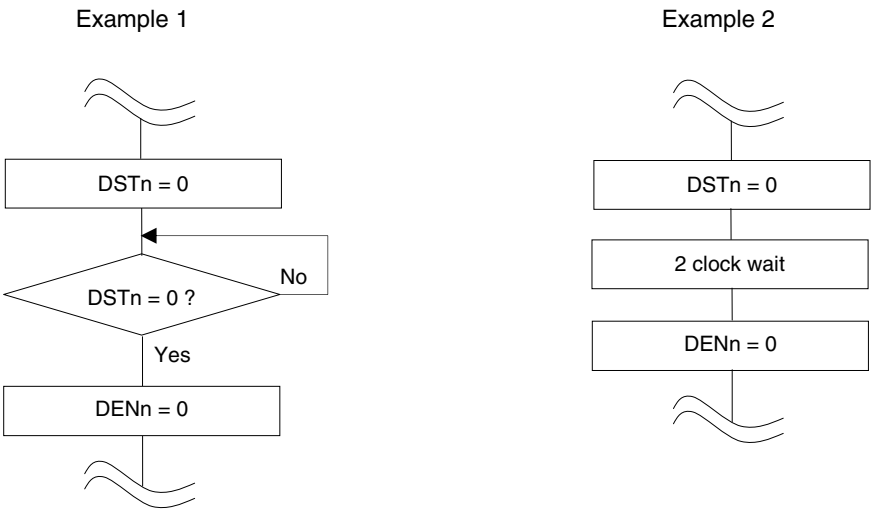
- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that DSTn has actually been cleared to 0, and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

< **IN case of using 2 channel for DMA** >

- Clear the DSTn (0) bit in case of forced termination (DSTn=0) by software while using 2 channel for DMA, after releasing DMA transfer hold by setting set (1) of DWAIT0, DWAIT1 along with channel2. After that, clear the DSTn bit after releasing the hold after clearing (0) DWAIT0, DWAIT1 along with channel 2.

Figure 15-13. Forced Termination of DMA Transfer (1/2)



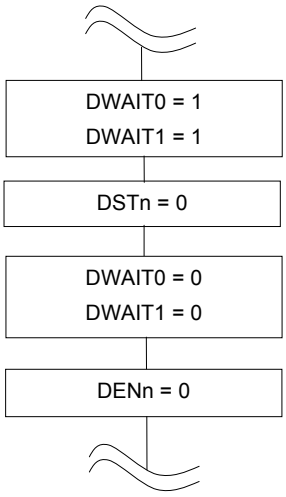
Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

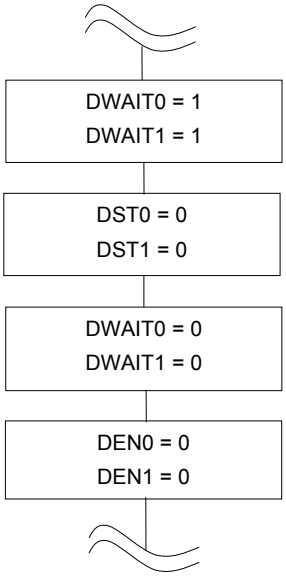
Figure 15-13. Forced Termination of DMA Transfer (1/2)

Example 3

Procedure for Forced Termination of any channel between use along with channel 2



Procedure for Forced Termination of any channel between use along with channel 2



Caution In example 3, wait of 2 clock after DWAITn set (1) is not important. Again as it is clearing more than two clocks, from clear (0) DSTn to clear (0) DEMn, it is not necessary to clear wait 2 after DSTn clear (0).

Remark 1. n: DMA Channel number (n=0,1)
2. 1 clock: 1/fclk (fclk: CPU clock)

15.6 Cautions on Using DMA Controller

(1) Priority of DMA

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 15-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note Execute the instruction from internal RAM, maximum time will become 16 clock.

Caution 1. Excluding the two clocks of DMA transfer

2. At the time of DMA hold release operation (See 15.6 (4)), maximum response time of each case operation time to hold that case becomes plus time.

3. Continuous transfer trigger of maximum response + channel 1 within clock 1, do not set it as it may 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 15-3. 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
- 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.

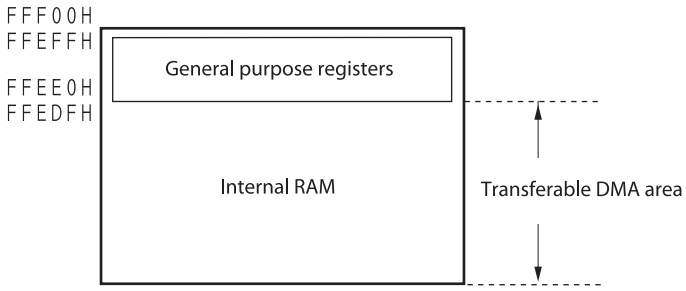
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

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 16 INTERRUPT FUNCTIONS

Interrupt factors change according to the product.

		78K0R/KC3-L (48pin)	78K0R/KE3-L (64pin)
Maskable interrupts	External	7	11
	Internal	36	41

16.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, 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 16-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupts

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

16.2 Interrupt Sources and Configuration

The 78K0R/KC3-L, KE3-L has interrupt sources including maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 16-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 16-1. Interrupt Source List (1/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time)	Internal	0004H	(A)
	1	INTLVI	Low-voltage detection[Note 4]		0006H	
	2	INTP0	Pin input edge detection	External	0008H	(B)
	3	INTP1			000AH	
	4	INTP2			000CH	
	6	INTP4			0010H	
	7	INTP5			0012H	
	8	INTST3	UART3 transmission transfer end or buffer empty interrupt	Internal	0014H	(A)
	9	INTSR3	UART3 reception transfer end		0016H	
	10	INTSRE3	UART3 reception communication error occurrence		0018H	
	11	INTDMA0	End of DMA0 transfer		001AH	
	12	INTDMA1	End of DMA1 transfer		001CH	
	13	INTST0 /INTCSI00	UART0 transmission transfer end or buffer empty interrupt/CSI00 communication end		001EH	
	14	INTSR0	UART0 reception transfer end		0020H	
	15	INTSRE0	UART0 reception communication error occurrence		0022H	
	16	INTST1 /INTCSI10 /INTIIC10	UART1 transmission transfer end / CSI10 communication / IIC10 communication end		0024H	
	17	INTSR1	UART1 reception transfer end		0026H	
	18	INTSRE1	UART1 reception communication error occurrence		0028H	
	19	INTIICA	End of IICA communication		002AH	
	20	INTTM00	End of timer array unit 0 channel 0 count or capture		002CH	
	21	INTTM01	End of timer array unit 0 channel 1 count or capture		002EH	

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 42 indicates the lowest priority.

2. Basic configuration types (A) - (D) correspond to (A) - (D) in Figure 16-1.

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 16-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	22	INTTM02	End of timer channel 2 count or capture		0030H	(A)
	23	INTTM03	End of timer channel 3 count or capture		0032H	
	24	INTAD	End of A/D conversion	Internal	0034H	
	25	INTRTC	Fixed-cycle signal of real-time counter/alarm match detection		0036H	
	26	INTRTCI	Interval signal detection of real-time counter		0038H	
	27	INTKR	Key return signal detection	External	003AH	(C)
	28	INTST2[Note 2] /INTCSI20 /INTIIC20	UART2 transmission transfer end / CSI20 transfer end / IIC20 transfer end	Internal	003CH	(A)
	29	INTP6	Pin input edge detection	External	003EH	(B)
	31	INTTM04	End of timer channel 4 count or capture		0042H	
	32	INTTM05	End of timer channel 5 count or capture		0044H	
	33	INTTM06	End of timer channel 6 count or capture		0046H	
	34	INTTM07	End of timer channel 7 count or capture		0048H	
	35	INTSR2[Note 2]	UART2 reception transfer end		004AH	
	36	INTP8[Note 2]	Pin input edge detection	External	004EH	
	37	INTP9[Note 2]			0050H	
	38	INTP10[Note 2]			0052H	
	39	INTP11[Note 2]			0054H	

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 42 indicates the lowest priority.

2. Only for 78K0R/KE3-L

3. Basic configuration types (A) - (D) correspond to (A) - (D) in Figure 16-1.

Table 16-1. Interrupt Source List (3/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
Maskable	40	INTSRE2[Note 2]	Generate communication error for UART2 reception		005CH	
	41	INTUSB	USB function status		005EH	
	42	INTRSUM	USB Resume signal detection		0060H	
Nonmaskable	–	INTDBG	OCD brake generation	–	0002H	
Software	–	BRK	Execution of BRK instruction	–	007EH	(D)
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–
		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 42 indicates the lowest priority.

2. Only for 78K0R/KE3-L

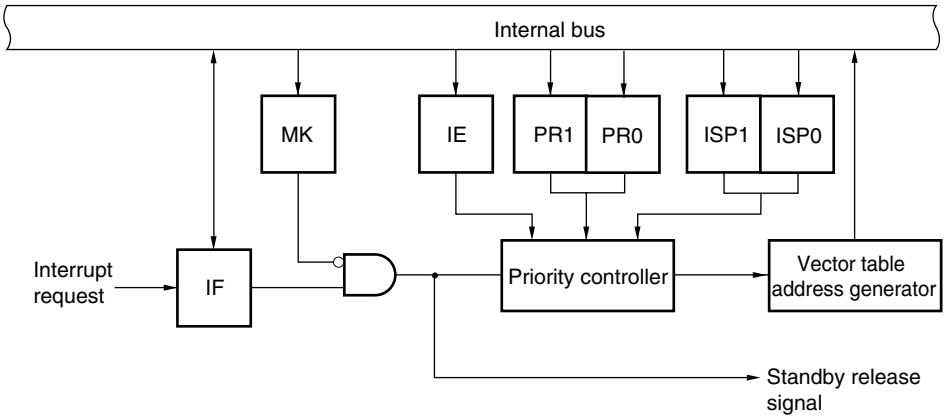
3. Basic configuration types (A) - (D) correspond to (A) - (D) in Figure 16-1.

4. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

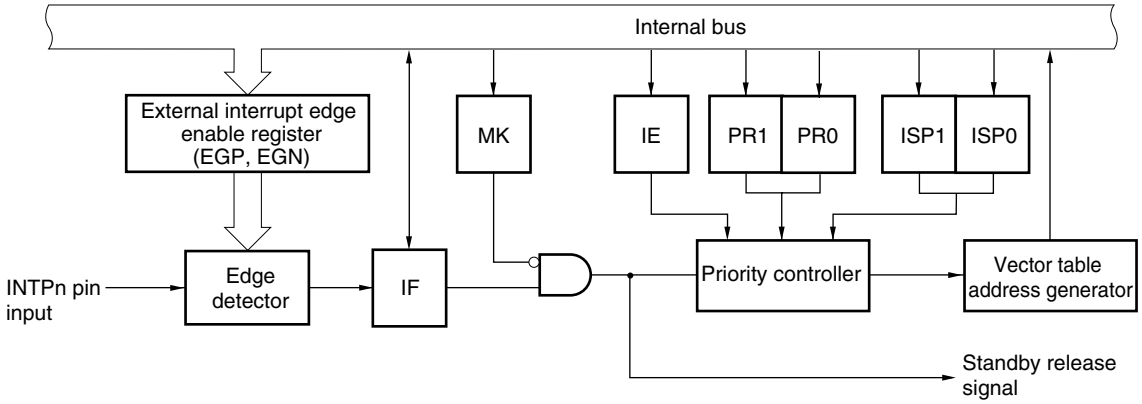
5. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 16-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



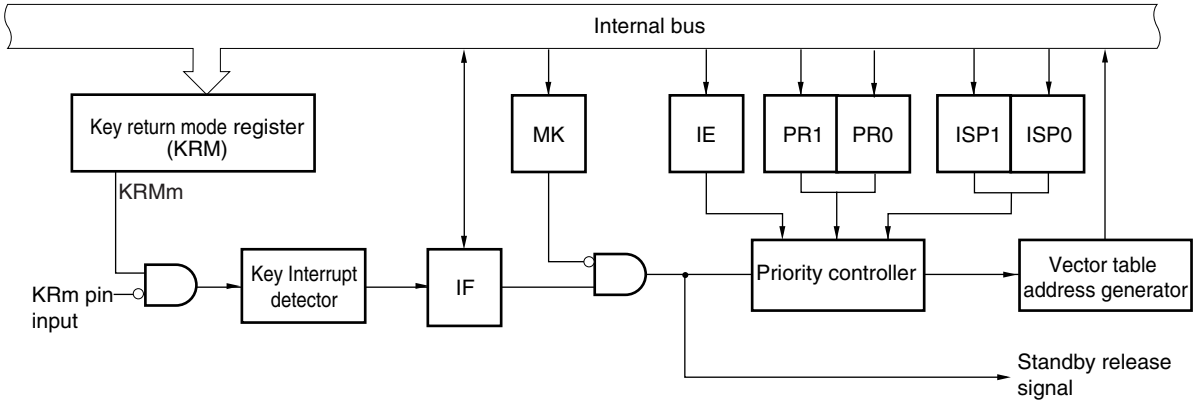
(B) External maskable interrupt (INTPn)



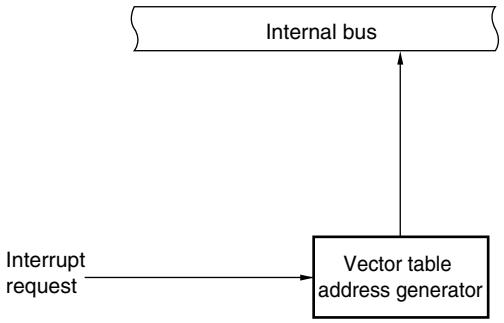
Remarks 1. IF : Interrupt request flag
IE : Interrupt enable flag
ISP0: In-service priority flag 0
ISP1:I n-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1:Priority specification flag 1
2. Only for 78K0R/KC3-L
n = 0 -2, 4-6
Only for 78K0R/KE3-L
n = 0 -2, 4-6, 8-11

Figure 16-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



Remarks 1. **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

2. Only for 78K0R/KC3-L
n = 0 - 3
Only for 78K0R/KE3-L
m = 0 - 7

16.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, 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 Note 1)
- External interrupt falling edge enable registers (EGN0, EGN1 Note 1)
- Program status word (PSW)

Note Only for 78K0R/KE3-L

Table 16-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 16-2. Flags Corresponding to Interrupt Request Sources (1/3)

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	
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	
INTST0Note	STIF0Note		STMK0Note		STPR00, STPR10Note	
INTCSI00Note	CSIIF00Note		CSIMK00Note		CSIPR000, CSIPR100Note	
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 IF1H is set to 1. Bit 5 of MK0H, PR00H, and PR10H supports these three interrupt sources.

Table 16-2. Flags Corresponding to Interrupt Request Sources (2/3)

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[Note 1]	
INTIIC10[Note 1]	IICIF10[Note 1]		IICMK10[Note 1]		IICPR010, IICPR110[Note 1]	
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	
INTKR	KRIF		KRMK		KRPR0, KRPR1	
INTST2[Note 2,3]	STIF2[Note 2,3]		STMK2[Note 2,3]		STPR02, STPR12[Note 2,3]	
INTCSI20[Note 2,3]	CSIIF20[Note 2,3]		CSIMK20[Note 2,3]		CSIPR020, CSIPR120[Note 2,3]	
INTIIC20[Note 2,3]	IICIF20[Note 2,3]		IICMK20[Note 2,3]		IICPR020, IICPR120[Note 2,3]	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	

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. Only for 78K0R/KE3-L

3. 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.

Table 16-2. Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register				Register
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTP8 Note	PIF8		PMK8		PPR08, PPR18	
INTP9 Note	PIF9		PMK9		PPR09, PPR19	
INTP10 Note	PIF10		PMK10		PPR010, PPR110	
INTP11 Note	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H
INTSRE2 Note	SREIF2		SREMK2		SREPR02, SREPR12	
INTUSB	USBIF		USBMK		USBPR0, USBPR1	
INTRSUM	RSUMIF		RSUMMK		RSUMPR0, RSUMPR1	

Note. Only for 78K0R/KE3-L

(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.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	0	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0	STIF0 CSIIF00	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	STIF1 CSIIF10 IICIF10

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	0	PIF6	STIF2 Note CSIIF20 IICIF20	KRIF	RTCIIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10 Note	PIF9 Note	PIF8 Note	0	SRIF2	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	RSUMIF	USBIF	0	SREIF2 Note	0	0	0	PIF11 Note

Note. Only for 78K0R/KE3-L

Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

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.

(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 16-3. 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	0	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0	STMK0 CSIMK00	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	STMK1 CSIMK10 IICMK10

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	0	PMK6	STMK2 Note CSIMK20 IICMK20	KRMK	RTCIMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10 Note	PMK9 Note	PMK8 Note	0	SRMK2	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	0	RSUMMK	USBMK	SREMK2 Note	0	0	0	PMK11 Note

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note. Only for 78K0R/KE3-L

(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 16-4. 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	0	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	0	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00	STPR00 CSIPR000	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10	STPR10 CSIPR100	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	STPR01 CSIPR010 IICPR010

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR1	SREPR11	SRPR11	STPR11 CSIPR110 IICPR110

Figure 16-4. 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	0	PPR06	STPR02 Note CSIPR020 IICPR020	KRPR0	RTCIPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	TMPR113	PPR16	STPR12 Note CSIPR120 IICPR120	KRPR1	RTCIPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010 Note	PPR09 Note	PPR08 Note	0	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110 Note	PPR19 Note	PPR18 Note	PPR17	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	0	RSUMPR0	USBPR0	SREPR02 Note	0	0	0	PPR011 Note

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	0	RSUMPR1	USBPR1	SREPR12 Note	0	0	0	PPR111 Note

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)

Note. Only for 78K0R/KE3-L

(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 16-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1 note) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1 note)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1 Note	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 Note	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0-6,8-11)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 16-3 shows the ports corresponding to EGPn and EGNn.

Table 16-3. Ports Corresponding to EGPn and EGNn

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1[Note]	EGN1	P50	INTP1
EGP2	EGN2	P51	INTP2
EGP3	EGN3	P30	INTP3
EGP4	EGN4	P31	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6
EGP8[Note]	EGN8	P74	INTP8
EGP9[Note]	EGN9	P75	INTP9
EGP10[Note]	EGN10	P76	INTP10
EGP11[Note]	EGN11	P77	INTP11

Note. Only for 78K0R/KE3-L

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 – 6, 8-11

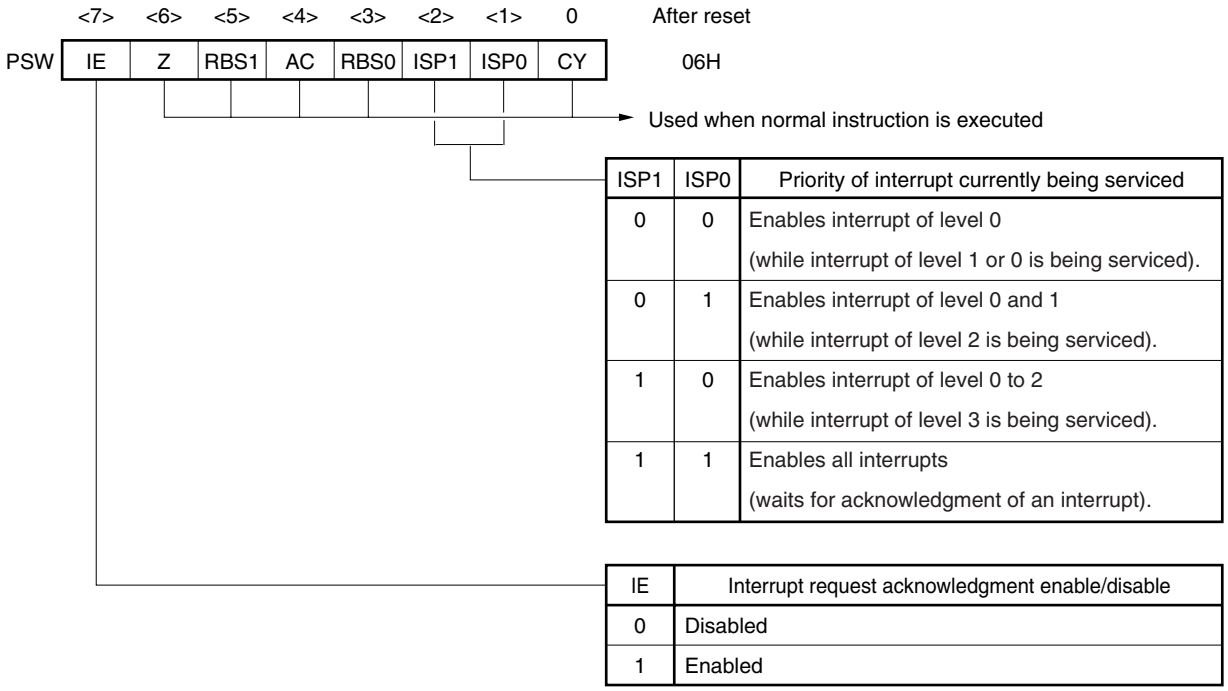
(5) Program status word (PSW)

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 ISP1 and ISP0 flags are set according to the priority specification level of the acknowledged interrupt. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 16-6. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables 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

16.4 Interrupt Servicing Operations

16.4.1 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 16-4 below.

For the interrupt request acknowledgment timing, see **Figures 16-8** and **16-9**.

Table 16-4. 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.

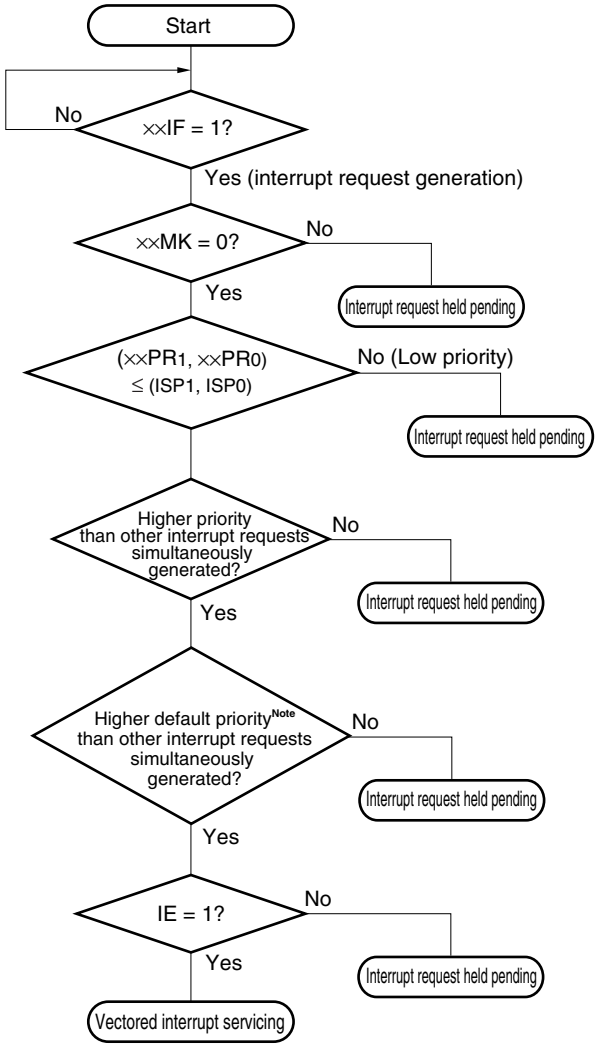
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 16-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

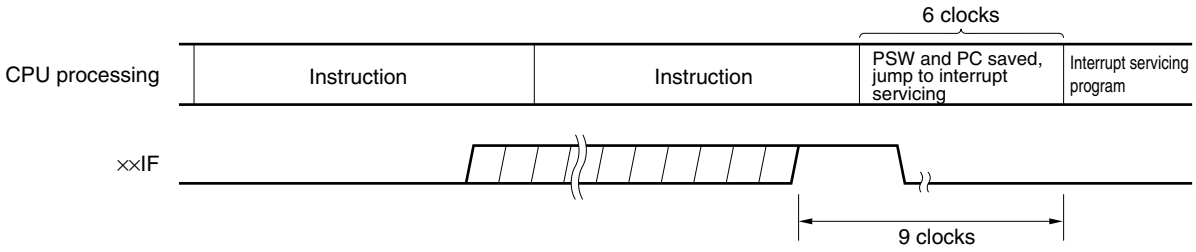
Figure 16-7. Interrupt Request Acknowledgment Processing Algorithm



××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 16-6**)

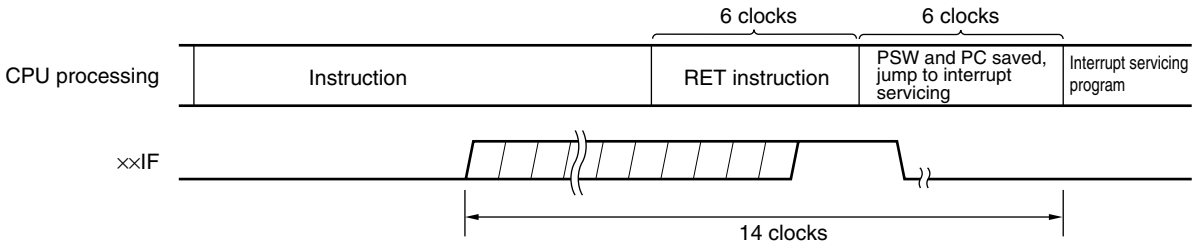
Note For the default priority, refer to Table 16-1 Interrupt Source List.

Figure 16-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 16-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

16.4.2 Software interrupt request acknowledgment

A software interrupt 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.

16.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority 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.

Table 16-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 16-10 shows multiple interrupt servicing examples.

Table 16-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0, ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0, ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1, ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1, ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

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)

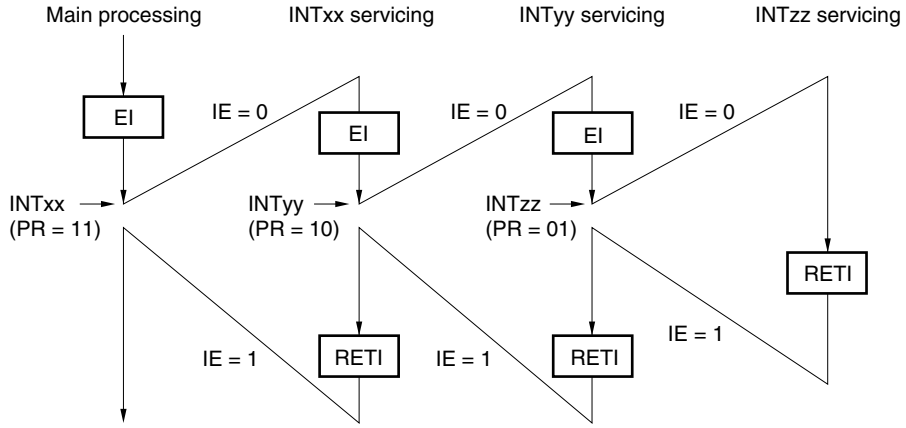
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

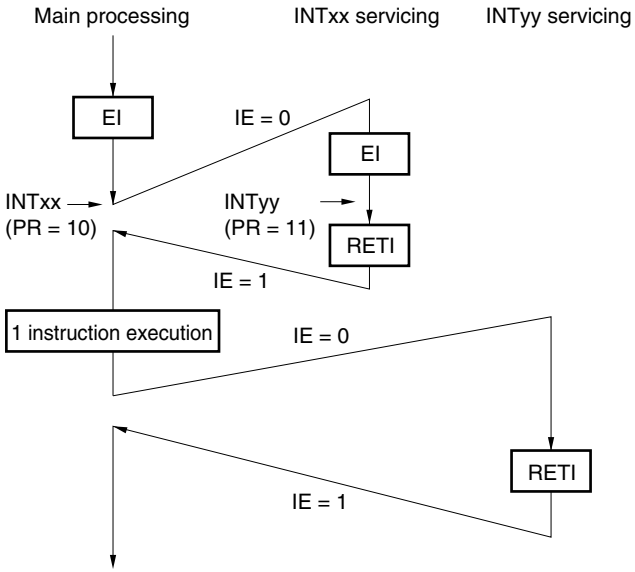
Figure 16-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

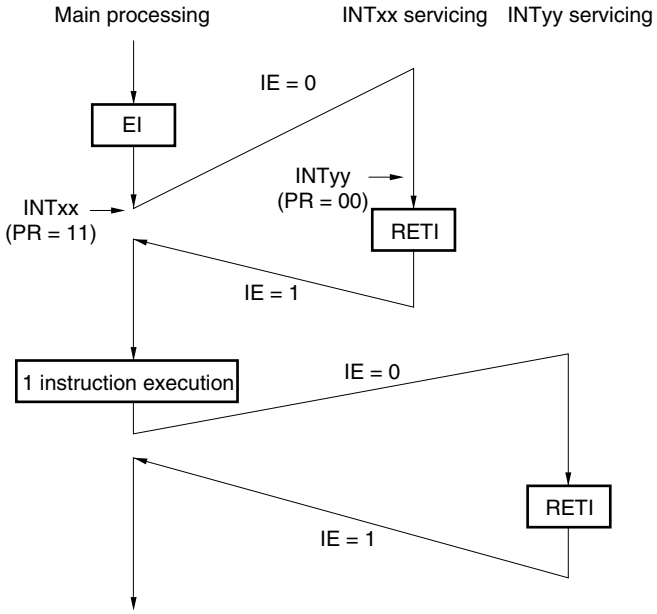


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Figure 16-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

16.4.4 Interrupt request hold

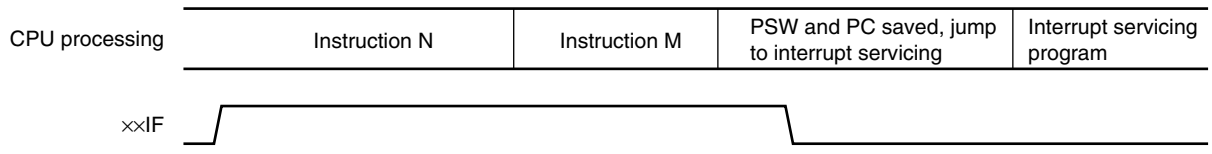
There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- <R> BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- <R> SKH
- <R> SKNH
- 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 16-11 shows the timing at which interrupt requests are held pending.

Figure 16-11. 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 17 KEY INTERRUPT FUNCTION

17.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pin.

Table 17-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRM0	Controls KR0 signal in 1-bit units.
KRM1	Controls KR1 signal in 1-bit units.
KRM2	Controls KR2 signal in 1-bit units.
KRM3	Controls KR3 signal in 1-bit units.
KRM4	Controls KR4 signal in 1-bit units.
KRM5Note	Controls KR5 signal in 1-bit units.
KRM6Note	Controls KR6 signal in 1-bit units.
KRM7Note	Controls KR7 signal in 1-bit units.

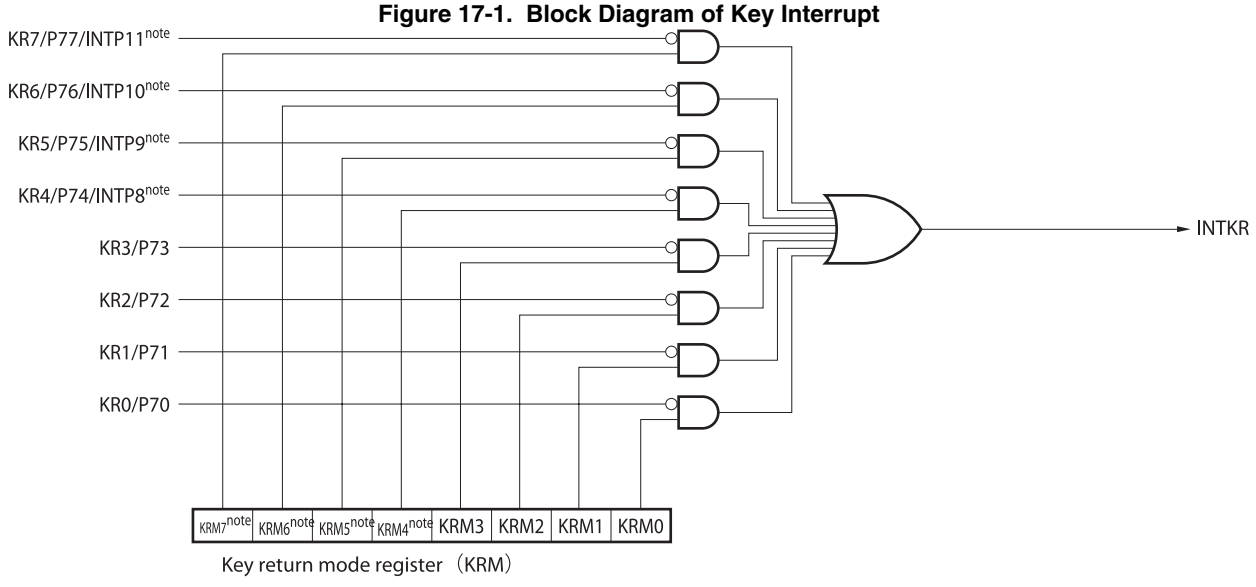
17.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 17-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM) Port mode register 7 (PM 7)

Figure 17-1. Block Diagram of Key Interrupt



Note 78K0R/KE3-L Only

17.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register controls the KRMn bits using the KRn signals, respectively.

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

Reset signal generation clears this register to 00H.

Figure 17-2. Format of Key Return Mode Register (KRM)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. If any of the KRMn bit used is set to 1, set bits 0 to 7 (PU7n) of the corresponding pull-up resistor register 7 (PU7) to 1.

2. An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (250 ns or more).

3. The bits not used in the key interrupt mode can be used as normal ports.

Note 78K0R/KE3-L only

(2) Port mode register (PM7)

In case of pins using P70/KR0, P71/KR1, P72/KR2, P73/KR3, P74/KR4/INTP8Note, P75/KR5/INTP9Note, P76/KR6/INTP10Note, P77/KR7/INTP11Note as key interrupt input please set each pin of PM70-PM77 to1. Output latch of P70-P77 can be set to either 0 or 1.

PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-3. Format of Port Mode Register (PM7)

Address:FFF27H After reset:FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77Note	PM76Note	PM75Note	PM74Note	PM73	PM72	PM71	PM70

PM7n	Selection of input mode P7n pin(n = 0-7)
0	Output mode(Output buffer on)
1	Input mode(Output buffer off)

Note 78K0R/KE3-L only

CHAPTER 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 Standby function

The standby function reduces the operating current of the system, and the following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, 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 23 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.

18.1.2 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 18-1. 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

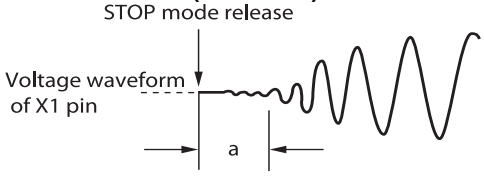
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status			
									f_X = 12 MHz	f_X = 16 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	2^8/fX min	21.33 μs max	16 μs max	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fX min	21.33 μs min	16 μs min	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fX min	42.67 μs min	32 μs min	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fX min	85.33 μs min	64 μs min	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fX min	170.67 μs min	128 μs min	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fX min	682.67 μs min	512 μs min	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fX min	2.73 ms min	2.05 ms min	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fX min	10.92 ms min	8.19 ms min	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fX min	21.85 ms min	16.38 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 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 18-2. 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			
				fx = 16 MHz	fx = 16 MHz	fx = 20 MHz
0	0	0	2^8/fx	21.33 μs	Setting prohibited	Setting prohibited
0	0	1	2^9/fx	42.67 μs	32 μs	25.6 μs
0	1	0	2^{10}/fx	85.33 μs	64 μs	51.2 μs
0	1	1	2^{11}/fx	170.67 μs	128 μs	102.4 μs
1	0	0	2^{13}/fx	682.67 μs	512 μs	409.6 μs
1	0	1	2^{15}/fx	2.73 ms	2.05 ms	1.64 ms
1	1	0	2^{17}/fx	10.92 ms	8.19 ms	6.55 ms
1	1	1	2^{18}/fx	21.85 ms	16.38 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.

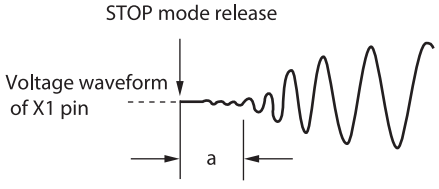
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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 fx: X1 clock oscillation frequency

18.2 Standby Function Operation

18.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, internal high-speed oscillation clock, 20 MHz internal high-speed oscillation clock, USB clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 18-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on 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		
	f_{USB}		Operation stopped (Setting prohibited)	Status before HALT mode was set is retained	
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Operation stopped. However, 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					
Serial array unit (SAU)					
Serial interface (IICA)					
USBF			Operation stopped	Operable	
Multiplier/divider			Operable		
DMA controller					
Power-on-clear function					
Low-voltage detection function					
External interrupt					
Key interrupt function					

Remark f_{IH} : Internal high-speed oscillation clock
f_{IH20} : 20 MHz internal high-speed oscillation clock
f_X : X1 clock
f_{EX} : External main system clock
f_{XT} : XT1 clock
f_{IL} : Internal low-speed oscillation clock
f_{USB} : USB Clock oscillation frequency

Table 18-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on 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}	
	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
	f_{USB}		Operation stopped (Setting prohibited)
CPU			Operation stopped
Flash memory			Operation stopped (wait state in low consumption current mode)
RAM			Operation stopped. However, 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
Serial array unit (SAU)			Operable
Serial interface (IICA)			Cannot operate
USBF			
Multiplier/divider			Operable
DMA controller			
Power-on-clear function			
Low-voltage detection function			
External interrupt			
Key interrupt function			

Remark f_{IH}: Internal high-speed oscillation clock
f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{IL}: Internal low-speed oscillation clock
f_{USB}: USB Clock oscillation frequency

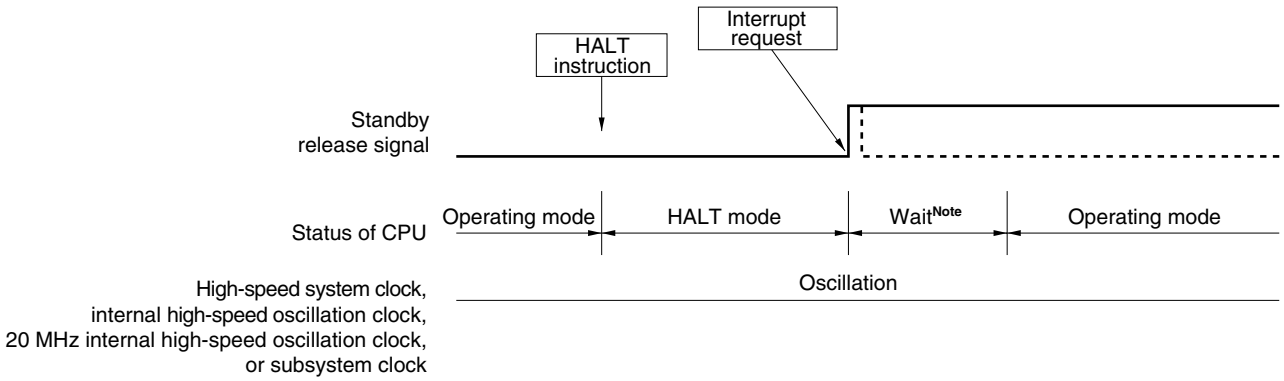
(2) HALT mode release

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 18-3. 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.

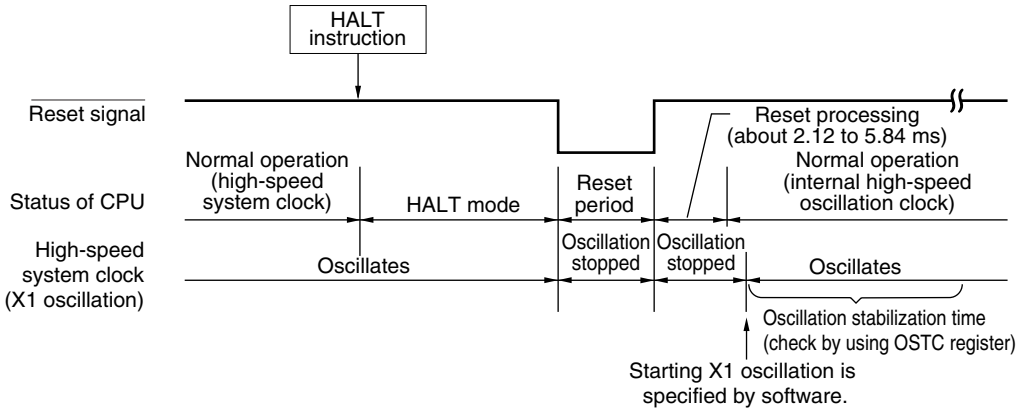
(b) Release by reset signal generation

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 18-4. HALT Mode Release by Reset(1/2)

(1) When high-speed system clock is used as CPU clock

<R>



(2) When internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock is used as CPU clock

<R>

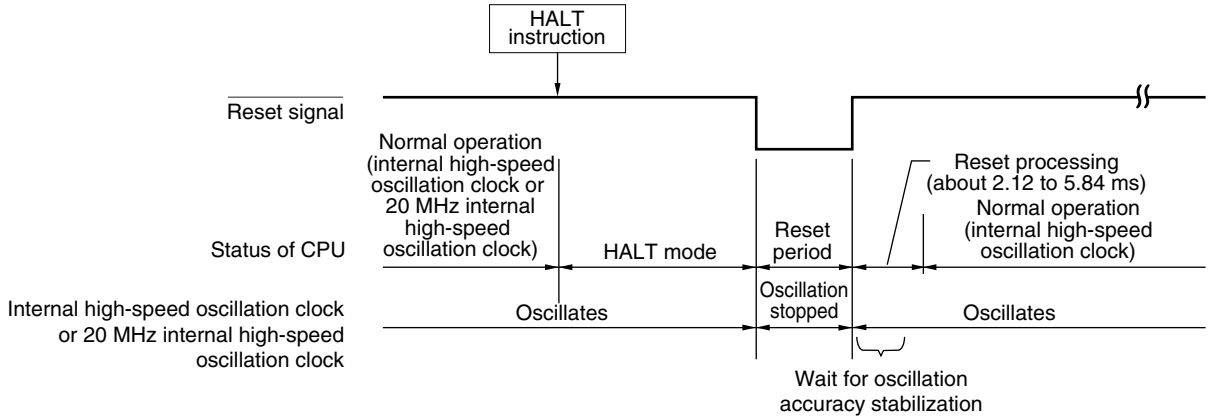
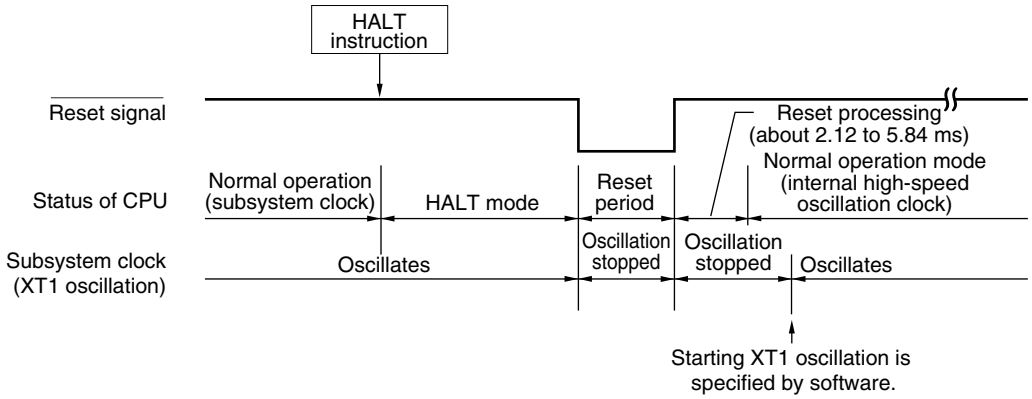


Figure 18-4. HALT Mode Release by Reset(2/2)

(3) When subsystem clock is used as CPU clock

<R>



Remark fx: X1 clock oscillation frequency

18.2.2 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 release 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 released 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 18-2. 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		
	f_{USB}		Operation stopped (Setting prohibited)		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Operation stopped. However, status before HALT 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			Operation disabled		
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 disabled		
Serial array unit (SAU)					
Serial interface (IICA)			Wakeup by address match operable		
USBF			Operation stopped		
Multiplier/divider			Operation disabled		
DMA controller					
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					
Key interrupt function					

Remark f_{IH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{IL}: Internal low-speed oscillation clock
f_{USB}: USB Clock oscillation frequency

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 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

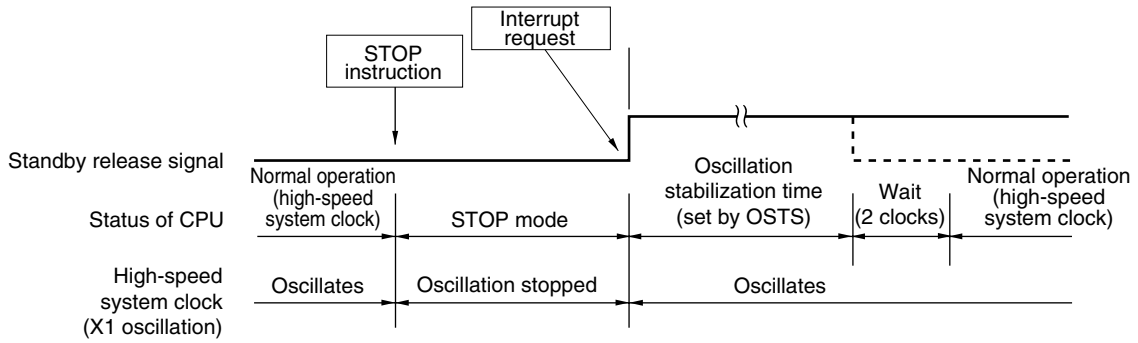
The STOP mode can be released by the following two sources.

(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 18-5. STOP Mode Release by Interrupt Request Generation (1/2)

<R> **(1) When high-speed system clock (X1 oscillation) is used as CPU clock**

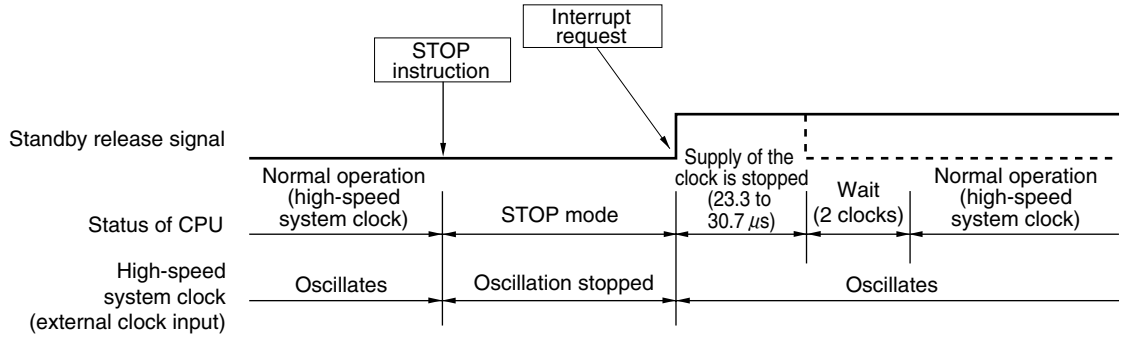


Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 18-5. STOP Mode Release by Interrupt Request Generation (2/2)

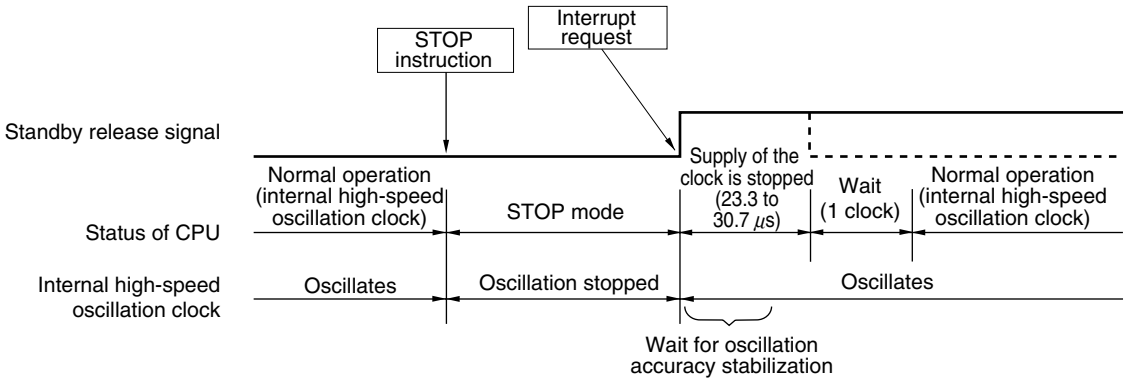
(2) When high-speed system clock (external clock input) is used as CPU clock

<R>



(3) When internal high-speed oscillation clock is used as CPU clock

<R>



Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

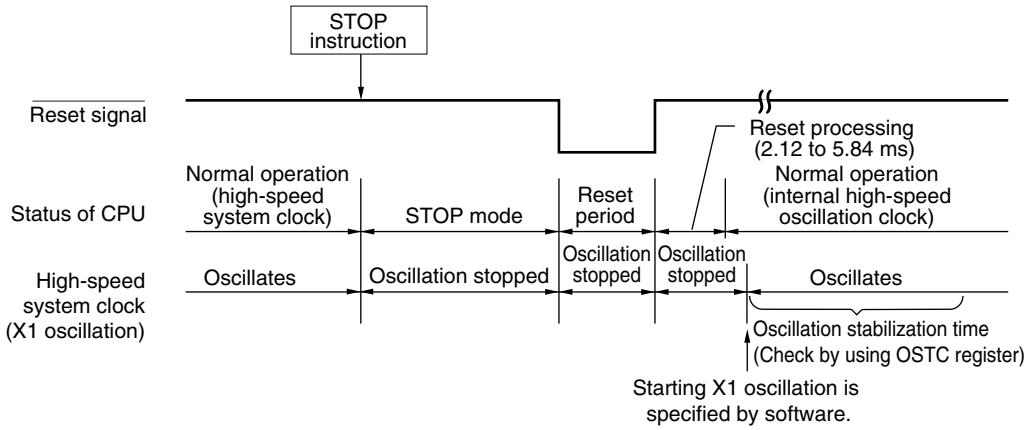
(b) Release by reset signal generation

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 18-6. STOP Mode Release by Reset

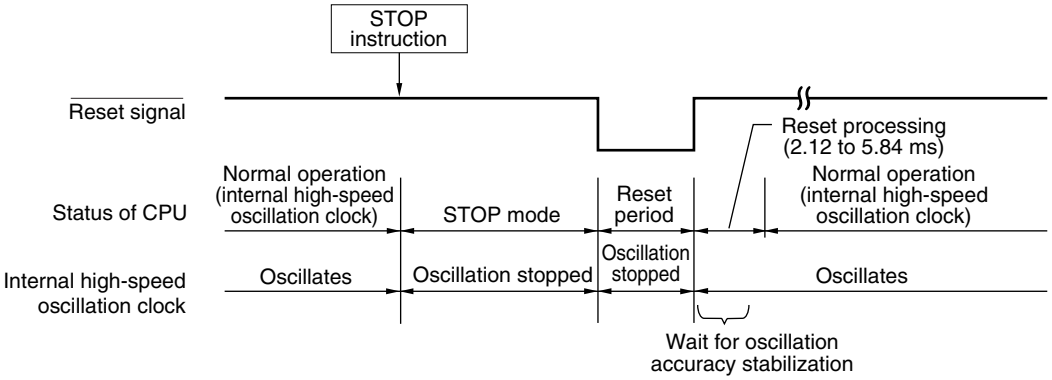
(1) When high-speed system clock is used as CPU clock

<R>



(2) When internal high-speed oscillation clock is used as CPU clock

<R>



Remark fx: X1 clock oscillation frequency

CHAPTER 19 RESET FUNCTION

The following six 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 or input voltage (EXLVI) from external input pin, and detection voltage of the low-voltage detector (LVI)
(5) Internal reset by execution of illegal instruction[Note]
<R> (6) Internal reset by a reset processing check error

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

A reset is affected 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 19-1 and 19-2. 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.

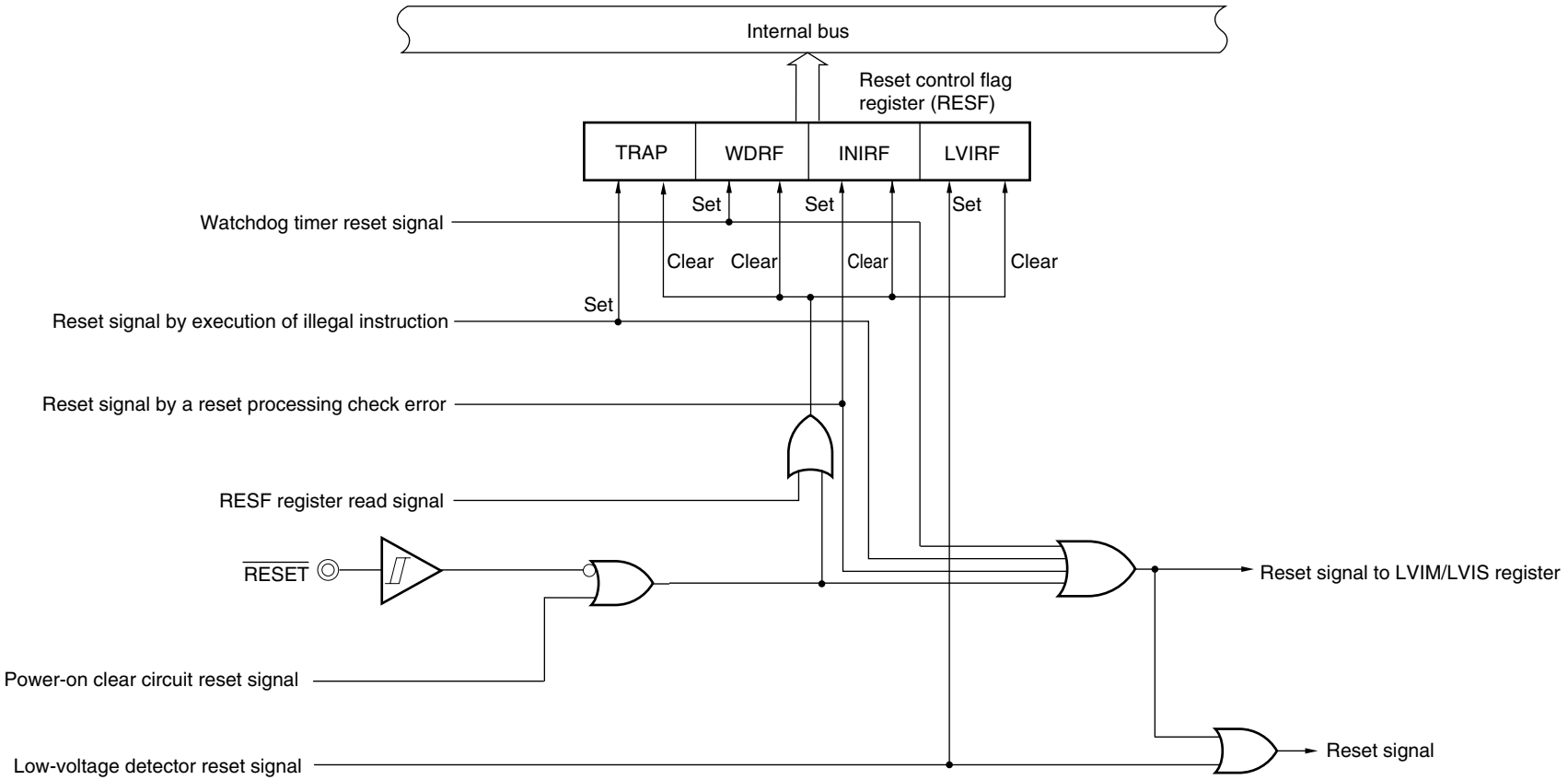
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 19-2 -19-4**) after reset processing. Reset by POC and LVI circuit voltage 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 20 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 21 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, internal low-speed oscillation clock and USB 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

<R> **Figure 19-1. 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 19-2. Timing of Reset by $\overline{\text{RESET}}$ Input

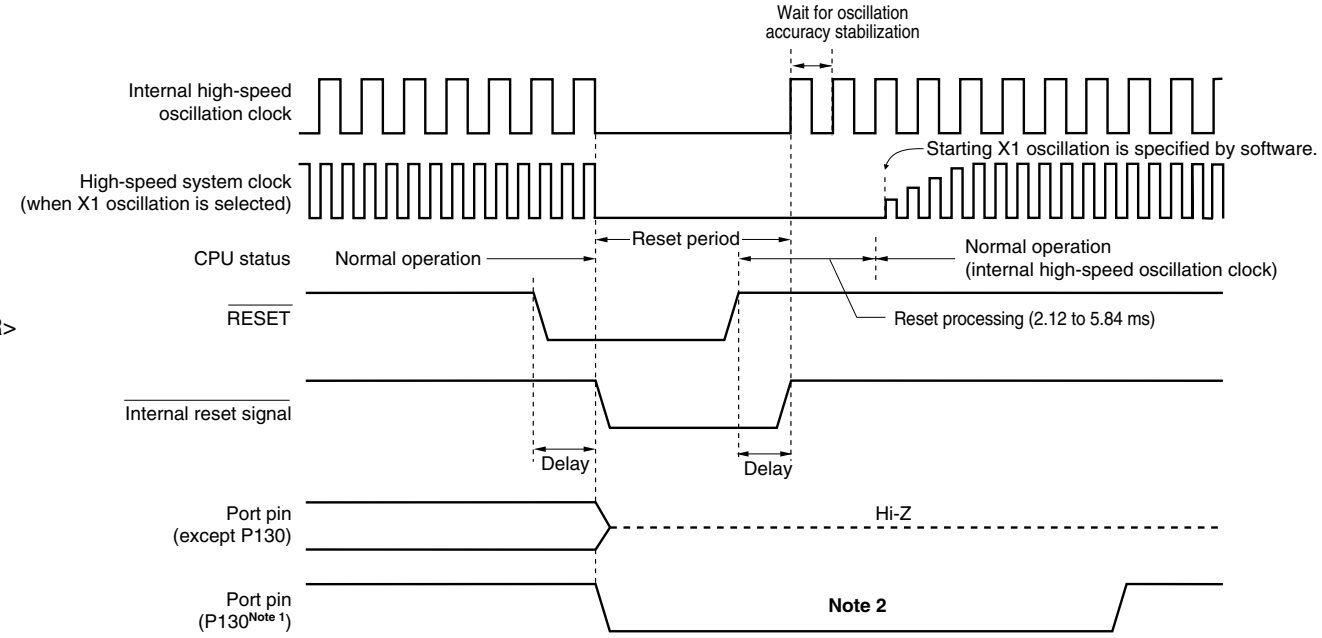
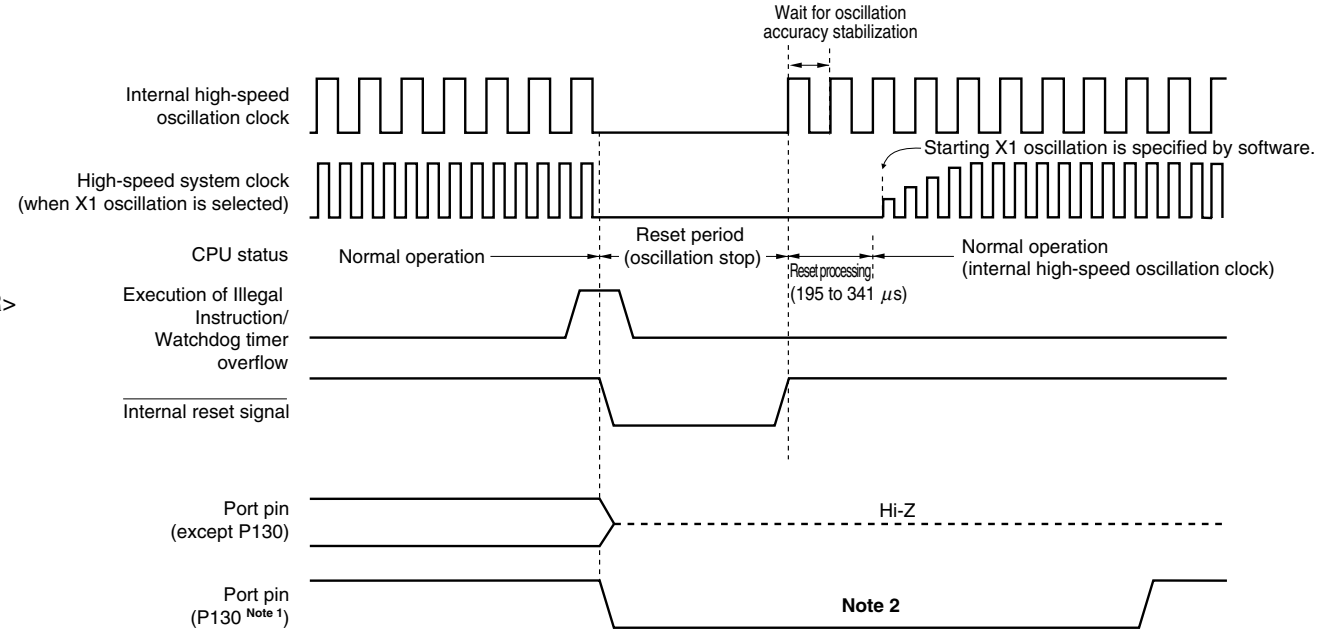


Figure 19-3. Timing of Reset Due to Watchdog Timer Overflow



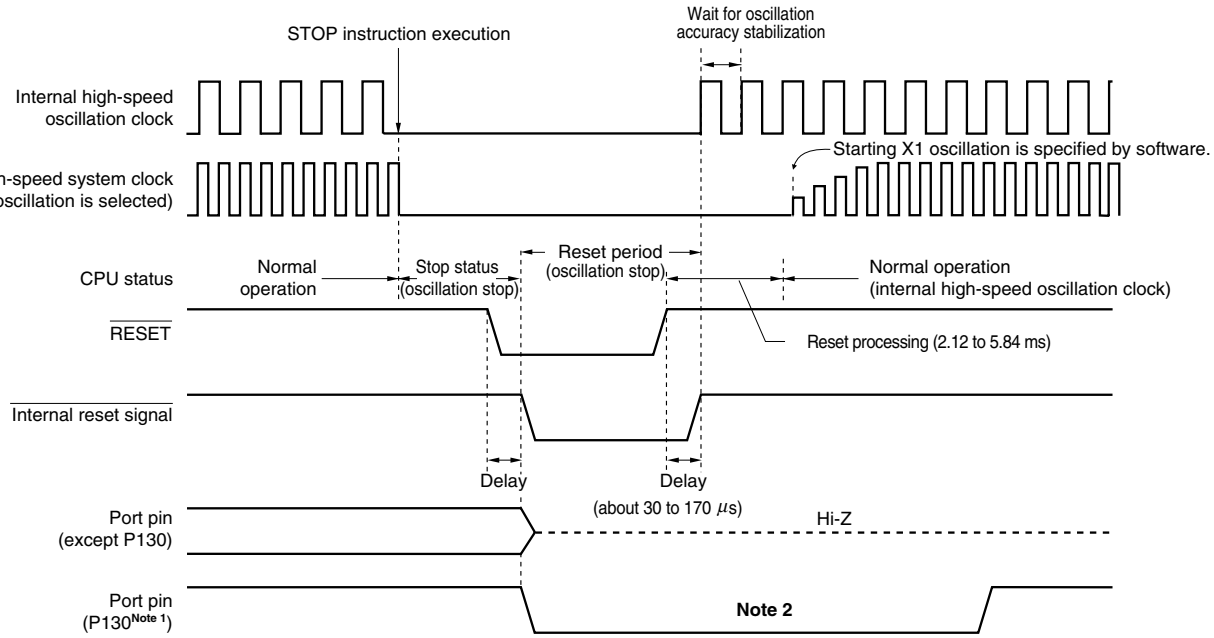
Note 1. 78K0R/KC3-L does not load pin P130.

2. When P130 is set to high-level output before reset is affected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is affected. To release a reset signal to an external device, set P130 to high-level output by software.

Caution A watchdog timer internal reset also resets the watchdog timer.

Figure 19-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

<R>



Note 1. 78K0R/KC3-L does not load pin P130.

2. When P130 is set to high-level output before reset is affected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is affected. To release a reset signal to an external device, set P130 to high-level output by software.

Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see CHAPTER 20 POWER-ON-CLEAR CIRCUIT and CHAPTER 21 LOW-VOLTAGE DETECTOR.

Table 19-1. Operation Statuses during Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}, f_{IH20}	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
	f_{USB}		
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[Note] 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			
Serial array unit (SAU)			
Serial interface (IICA)			
USBF			
Multiplier/divider			
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
Key interrupt function			

Note 78K0R/KE3-L only.

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
f_{USB} : USB clock oscillation frequency.

Table 19-2. 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]
Processor mode control register (PMC)		00H
Port registers (P0 - P7, P11[Note 3], P12, P13[Note 3], P14) (output latches)		00H
Port mode registers (PM0 - PM7, PM11[Note 3], PM12, PM14)		FFH
Port input mode registers 0, 1, 14[Note 3] (PIM0, PIM1, PIM14[Note 3])		00H
Port output mode registers 0, 1, 14[Note 3] (POM0, POM1, POM14[Note 3])		00H
Pull-up resistor option registers (PU0, PU1, PU3, PU5, PU7, PU11[Note 3], PU12, PU14)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
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 1 (NFEN1)		00H
Peripheral enable registers 0, 1 (PER0, PER1)		00H
Operation speed mode control register (OSMC)		00H
Timer array unit (TAU)	Timer data registers 00 – 07 (TDR00 - TDR07)	0000H
	Timer mode registers 00 - 07 (TMR00 - TMR07)	0000H
	Timer status registers 00 - 07 (TSR00 - TSR07)	0000H
	Timer input select registers 0 (TIS0)	00H
	Timer counter registers 00 - 07 (TCR00 - TCR07)	FFFFH
	Timer channel enable status registers 0 (TE0)	0000H
	Timer channel start registers 0 (TS0)	0000H
	Timer channel stop registers 0 (TT0)	0000H
	Timer clock select registers 0 (TPS0)	0000H
	Timer output registers 0 (TO0)	0000H
	Timer output enable registers 0 (TOE0)	0000H
	Timer output level registers 0 (TOL0)	0000H
	Timer output mode registers 0 (TOM0)	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.

3. 78K0R/KE3-L only.

Table 19-2. Hardware Statuses After Reset Acknowledgment (2/4)

Hardware		Status After Reset Acknowledgment[Note 1]
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
	Control register 0 (RTCC0)	00H
	Control register 1 (RTCC1)	00H
	Control register 2 (RTCC2)	00H
Clock output/buzzer output controller	Clock output select registers 0 (CKS0)	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
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H
Serial array unit (SAU)	Serial data registers 00 - 03, 10[Note 3], 11[Note 3], 12, 13 (SDR00 - SDR03, SDR10[Note 3], SDR11[Note 3], SDR12, SDR13)	0000H
	Serial status registers 00 - 03, 10[Note 3], 11[Note 3], 12, 13 (SSR00 - SSR03, SSR10[Note 3], SSR11[Note 3], SSR12, SSR13)	0000H
	Serial flag clear trigger registers 00 - 03, 10[Note 3], 11[Note 3], 12, 13 (SIR00 - SIR03, SIR10[Note 3], SIR11[Note 3], SIR12, SIR13)	0000H
	Serial mode registers 00 - 03, 10[Note 3], 11[Note 3], 12, 13 (SMR00 - SMR03, SMR10[Note 3], SMR11[Note 3], SMR12, SMR13)	0020H
	Serial communication operation setting registers 00 - 03, 10[Note 3], 11[Note 3], 12, 13 (SCR00 - SCR03, SCR10[Note 3], SCR11[Note 3], 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

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.

3. 78K0R/KE3-L only.

Table 19-2. Hardware Statuses after Reset Acknowledgment (3/4)

Hardware		Status After Reset Acknowledgment[Note]
USB Function Controller USBF	UF0 EP0NAK Register (UF0E0N)	00H
	UF0 EP0NAKALL Register (UF0E0NA)	00H
	UF0 EPNAK Register (UF0EN)	00H
	UF0 EPNAK Mask Register (UF0ENM)	00H
	UF0 SNDSIE Register (UF0SDS)	00H
	UF0 CLR Request Register (UF0CLR)	00H
	UF0 SET Request Register (UF0SET)	00H
	UF0 EP Statusn Register (UF0EPSn)(n = 0-2)	00H
	UF0 INT Statusn Register (UF0ISn)(n = 0-4)	00H
	UF0 INT Maskn Register (UF0IMn)(n = 0-4)	00H
	UF0 INT Clearn Register (UF0ICn)(n = 0-4)	FFH
	UF0 FIFO Clearn Register (UF0FICn)(n = 0, 1)	00H
	UF0 Data End Register (UF0DEND)	00H
	UF0 GPR Register (UF0GPR)	00H
	UF0 Mode Control Register (UF0MODC)	00H
	UF0 Mode Status Register (UF0MODS)	00H
	UF0 Active Interface Number Register (UF0AIFN)	00H
	UF0 Active Alternative Setting Register (UF0AAS)	00H
	UF0 Endpoint n Interface Mapping Register (UF0EnIM)(n = 1-4, 7, 8)	00H
	UF0 EP0 Read Register (UF0E0R)	Undefined
	UF0 EP0 Length Register (UF0E0L)	00H
	UF0 EP0 Setup Register (UF0E0ST)	00H
	UF0 EP0 Write Register (UF0E0W)	00H
	UF0 Bulk Out 1 Register (UF0BO1)	Undefined
	UF0 Bulk Out 1 Length Register (UF0BO1L)	00H
	UF0 Bulk In 1 Register (UF0BI1)	00H
	UF0 Device Status Register (UF0DSTL)	00H
	UF0 EPn Status Register L(UF0EnSL)(n = 0-4, 7, 8)	00H
	UF0 Address Register (UF0ADRS)	00H
	UF0 Configuration Register (UF0CNF)	00H
	UF0 Interface n Register (UF0IFn)(n = 0-4)	00H
	UF0 Descriptor Length Register (UF0DSCL)	00H
	UF0 Device Descriptor Register (UF0DDn)(n = 0-17)	Undefined
	UF0 Configuration/ Interface/ End point Descriptor Register (UF0CIEn)(n = 0-255)	Undefined
	USB Function0 Buffer Control Register (UF0BC)	00H

Note During reset input generation or oscillation stabilization wait time, only the PC contents among the hardware statuses become undefined.

Table 19-2. Hardware Statuses after Reset Acknowledgment (4/4)

	Hardware		Status After Reset Acknowledgment Note 1
	Serial interface IICA	IICA shift register (IICA)	00H
		IICA status register (IICS)	00H
		IICA flag register (IICF)	00H
		IICA control register 0 (IICCTL0)	00H
		IICA control register 1 (IICCTL1)	00H
		IICA low-level width setting register (IICWL)	FFH
		IICA high-level width setting register (IICWH)	FFH
		Slave address register (SVA)	00H
	Multiplier/divider	Multiplication/division data register A (L) (MDAL)	0000H
		Multiplication/division data register A (H) (MDAH)	0000H
		Multiplication/division data register B (L) (MDBL)	0000H
		Multiplication/division data register B (H) (MDBH)	0000H
		Multiplication/division data register C (L) (MDCL)	0000H
		Multiplication/division data register C (H) (MDCH)	0000H
		Multiplication/division control register (MDUC)	00H
	Key interrupt	Key return mode register (KRM)	00H
<R>	Reset function	Reset control flag register (RESF)	00H Note 2
	Low-voltage detector	Low-voltage detection register (LVIM)	00H Note 3
		Low-voltage detection level select register (LVIS)	0EH Note 2
	Regulator	Regulator mode control register (RMC)	00H
	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 Note 4)	00H
		External interrupt falling edge enable registers 0, 1 (EGN0, EGN1 Note 4)	00H
	BCD correction circuit	BCD correction result register (BCDAJ)	Undefined

(Note is listed on the next page.)

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.

<R>

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by INIRF	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held
	WDRF bit			Held	Set (1)	Held	Held
	INIRF bit			Held	Held	Set (1)	Held
	LVIRF bit			Held	Held	Held	Set (1)
LVIS		Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Held

3. This value varies depending on the reset source and the option byte.

4. 78K0R/KE3-L only.

19.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0R/KC3-L, KE3-L. 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 register clear TRAP, WDRF, INIRF, and LVIRF flags.

<R> **Figure 19-5. Format of Reset Control Flag Register (RESF)**

Address: FFFA8H After reset: 00H Note 1 R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDRF	0	0	INIRF	LVIRF

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.

INIRF	Internal reset request t by a reset processing check error
0	Internal reset request is not generated, or the RESF register 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 19-3.

<R>

Table 19-3. RESF Status when Reset Request Is Generated

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by INIRF	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held
	WDRF bit			Held	Set (1)	Held	Held
	INIRF bit			Held	Held	Set (1)	Held
	LVIRF bit			Held	Held	Held	Set (1)

CHAPTER 20 POWER-ON-CLEAR CIRCUIT

20.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) 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.

  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.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

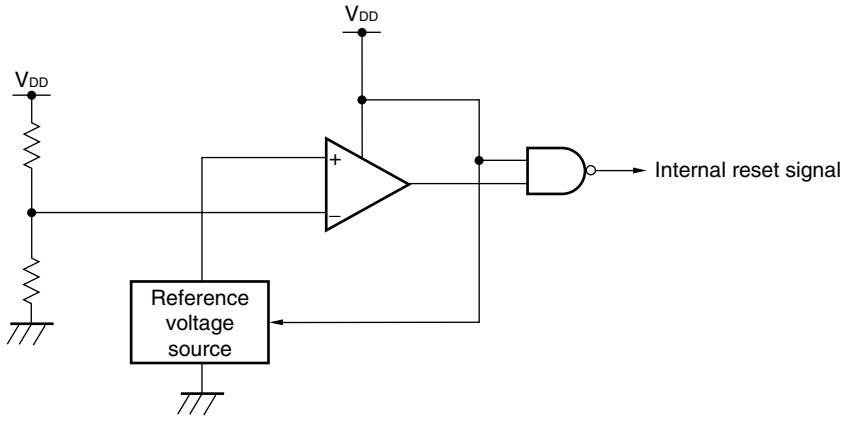
  Caution If an internal reset signal is generated in the POC circuit, TRAP, WDRF, INIRF, and LVIRF flags of the reset control flag register (RESF) is cleared (00H).

  Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), 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, LVI, or illegal instruction execution.
  For details of RESF, see CHAPTER 19 RESET FUNCTION.

20.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 20-1.

Figure 20-1. Block Diagram of Power-on-Clear Circuit



20.3 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), the reset status is released.

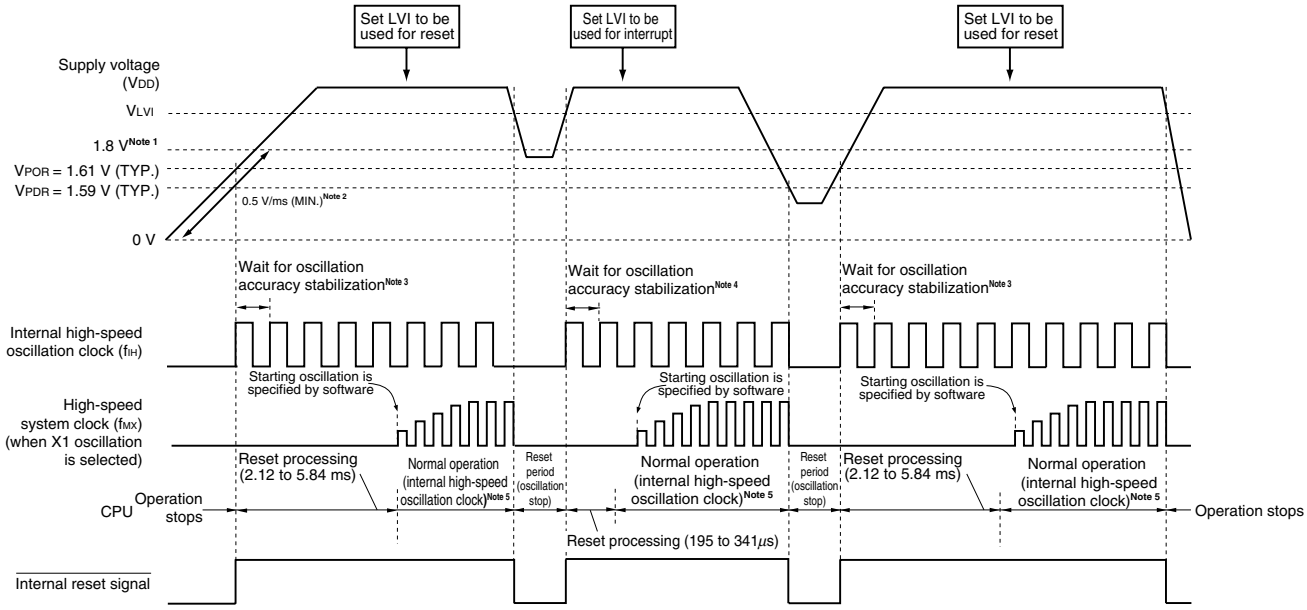
  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.

- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V) are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

<R> **Figure 20-2. 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} ≤ 3.6 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.

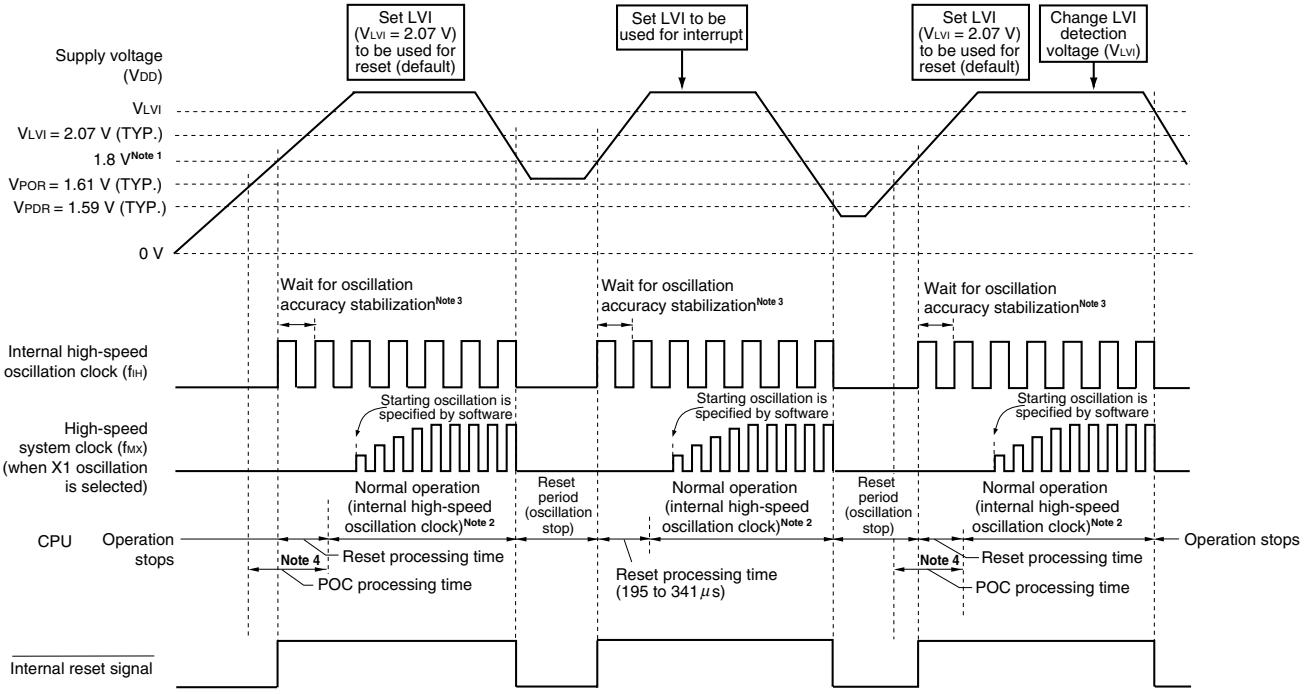
5. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 21 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

<R> **Figure 20-2. 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} ≤ 3.6 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 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. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

4. The following times are required between reaching the POC detection voltage (1.61 V (TYP.)) and starting normal operation.

- **When the time to reach 2.07 V (TYP.) from 1.61 V (TYP.) is less than 5.8 ms:**
  A POC processing time of 2.12 to 5.84 ms is required between reaching 1.61 V (TYP.) and starting normal operation.
- **When the time to reach 2.07 V (TYP.) from 1.61 V (TYP.) is greater than 5.8 ms:**
  A reset processing time of 195 to 341 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 20 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

20.4 Cautions for Power-on-Clear Circuit

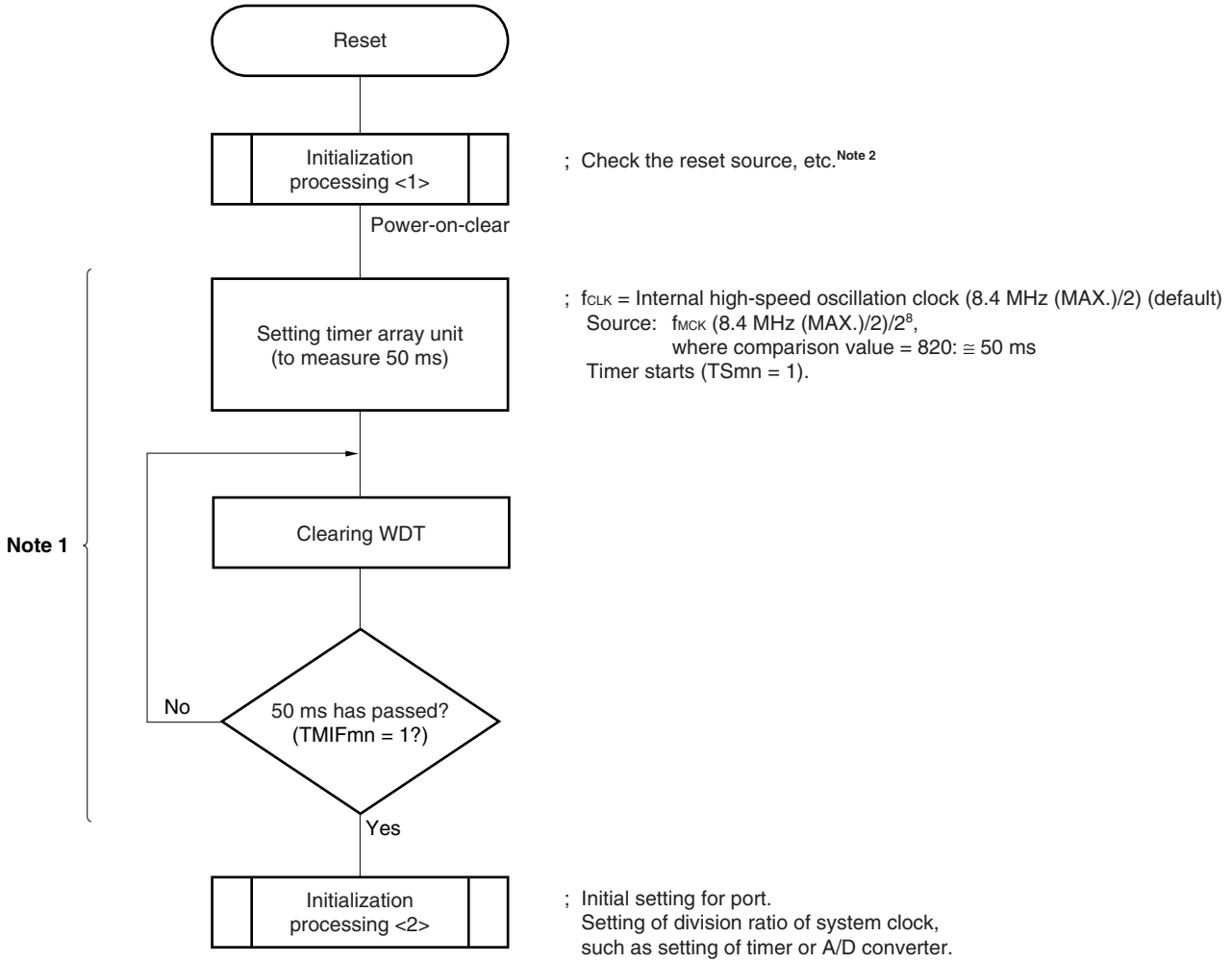
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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.

<R> **Figure 20-3. Example of Software Processing After Reset Release (1/2)**

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage



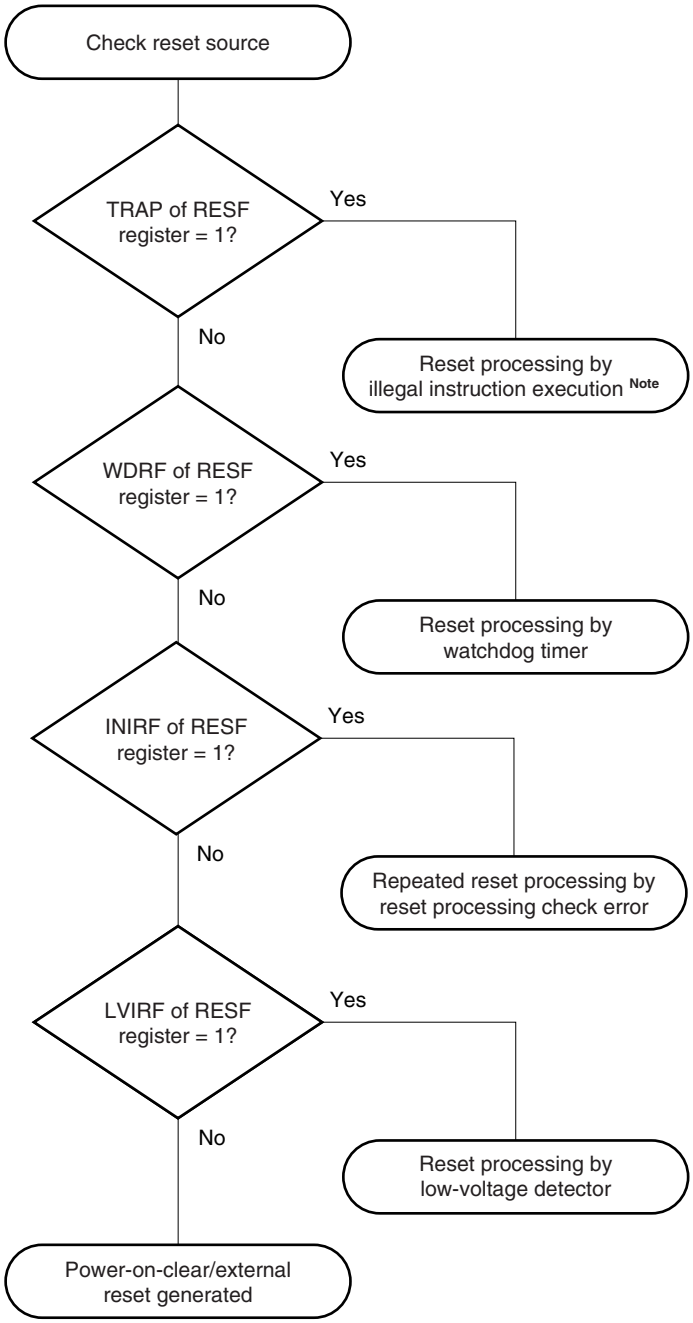
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n = 0 to 7

<R> **Figure 20-3. 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 21 LOW-VOLTAGE DETECTOR

21.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) 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), 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). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V).
- 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}, 11 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

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

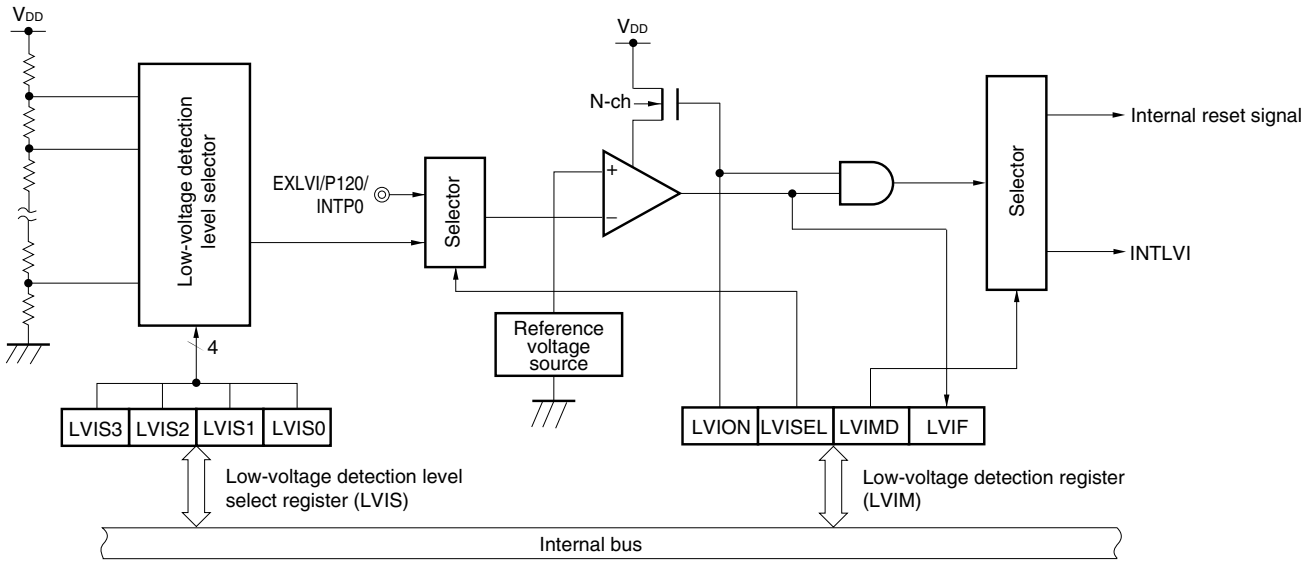
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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 19 RESET FUNCTION**.

21.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 21-1.

Figure 21-1. Block Diagram of Low-Voltage Detector



21.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection registers (LVIM)
- Low-voltage detection level select register (LVIS)
- Port mode register 12 (PM12)

(1) Low-voltage detection registers (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 21-2. 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[Note 3]	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.

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.

4. To read LVIM after writing this register, secure the time of one or more clock.

(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 21-3. Format of Low-Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 0EH Note RW

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	1	0	1	V_{LVI5}(3.45 ±0.1 V)
0	1	1	0	V_{LVI6}(3.30 ±0.1 V)
0	1	1	1	V_{LVI7}(3.15 ±0.1 V)
1	0	0	0	V_{LVI8}(2.99 ±0.1 V)
1	0	0	1	V_{LVI9}(2.84 ±0.1 V)
1	0	1	0	V_{LVI10}(2.68 ±0.1 V)
1	0	1	1	V_{LVI11}(2.53 ±0.1 V)
1	1	0	0	V_{LVI12}(2.38 ±0.1 V)
1	1	0	1	V_{LVI13}(2.22 ±0.1 V)
1	1	1	0	V_{LVI14}(2.07 ±0.1 V)
1	1	1	1	V_{LVI15}(1.91 ±0.1 V)
Other than above				Setting prohibited

Note 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 affected.

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 an LVIIF flag with software because it 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 an LVIIF flag with software because it 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.

4. To read LVIM after writing this register, secure the time of one or more clock.

(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 21-4. 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)

21.4 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}= 1.21 V ±0.1 V), 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). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V).

(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). 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).

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

21.4.1 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 21-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> - <7> above.

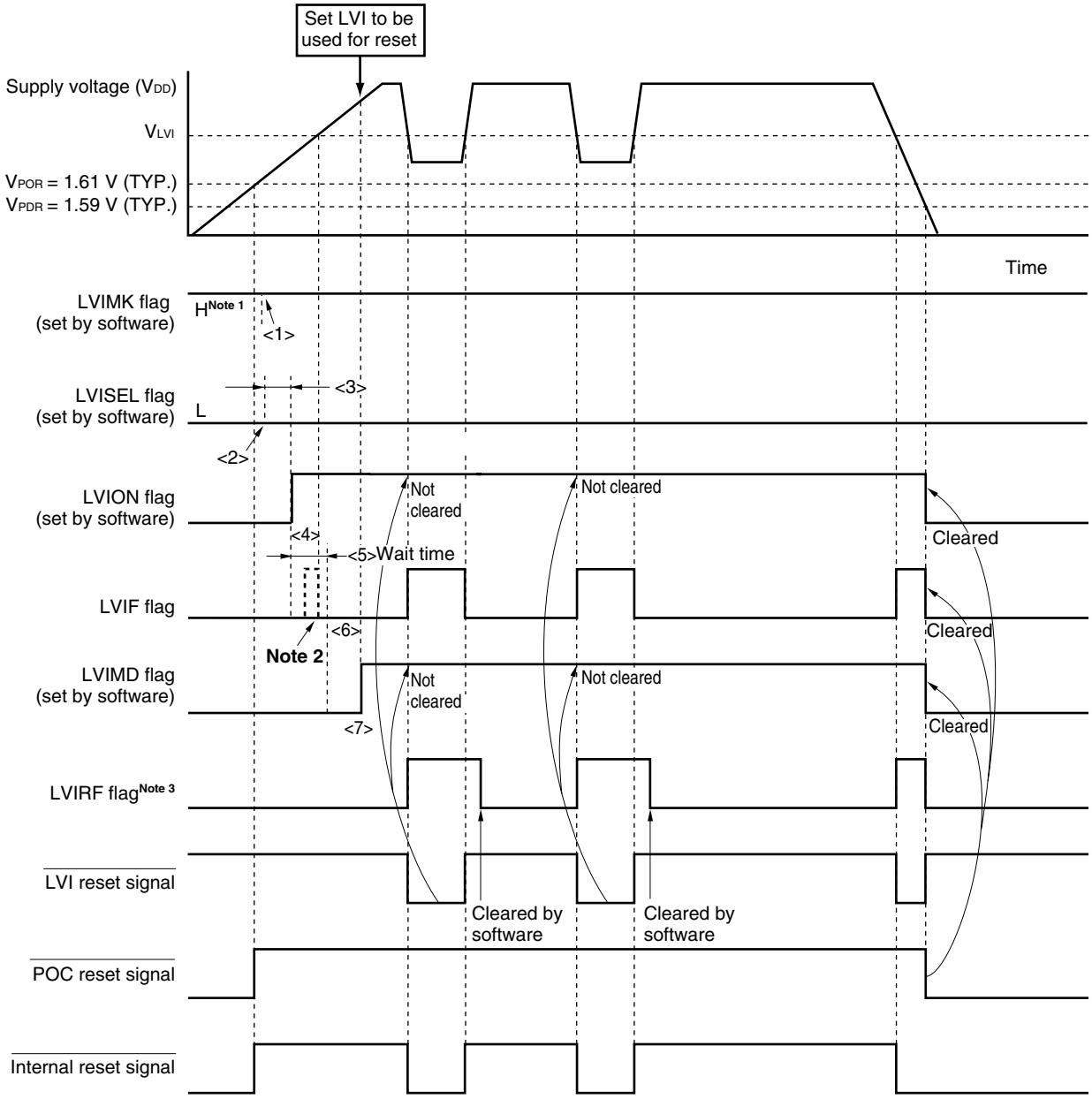
Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

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 21-5. 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 19 RESET FUNCTION.

Remarks 1. <1> to <7> in Figure 21-5 above correspond to <1> to <7> in the description of "When starting operation" in 21.4.1 (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 21-6 shows the timing of the internal reset signal generated by the low-voltage detector.

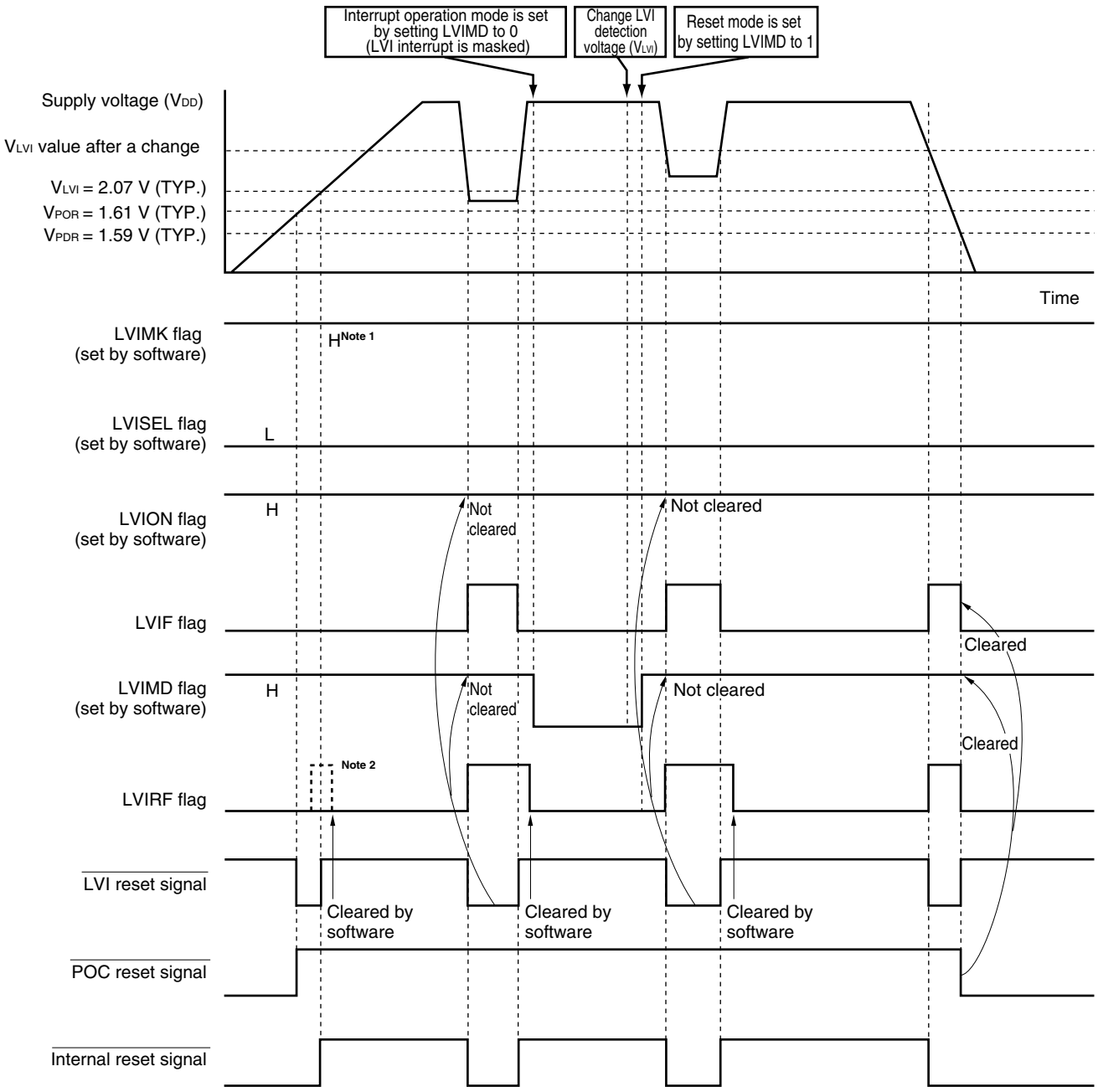
- When stopping operation

  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 21-6. 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 19 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 21-7 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>.

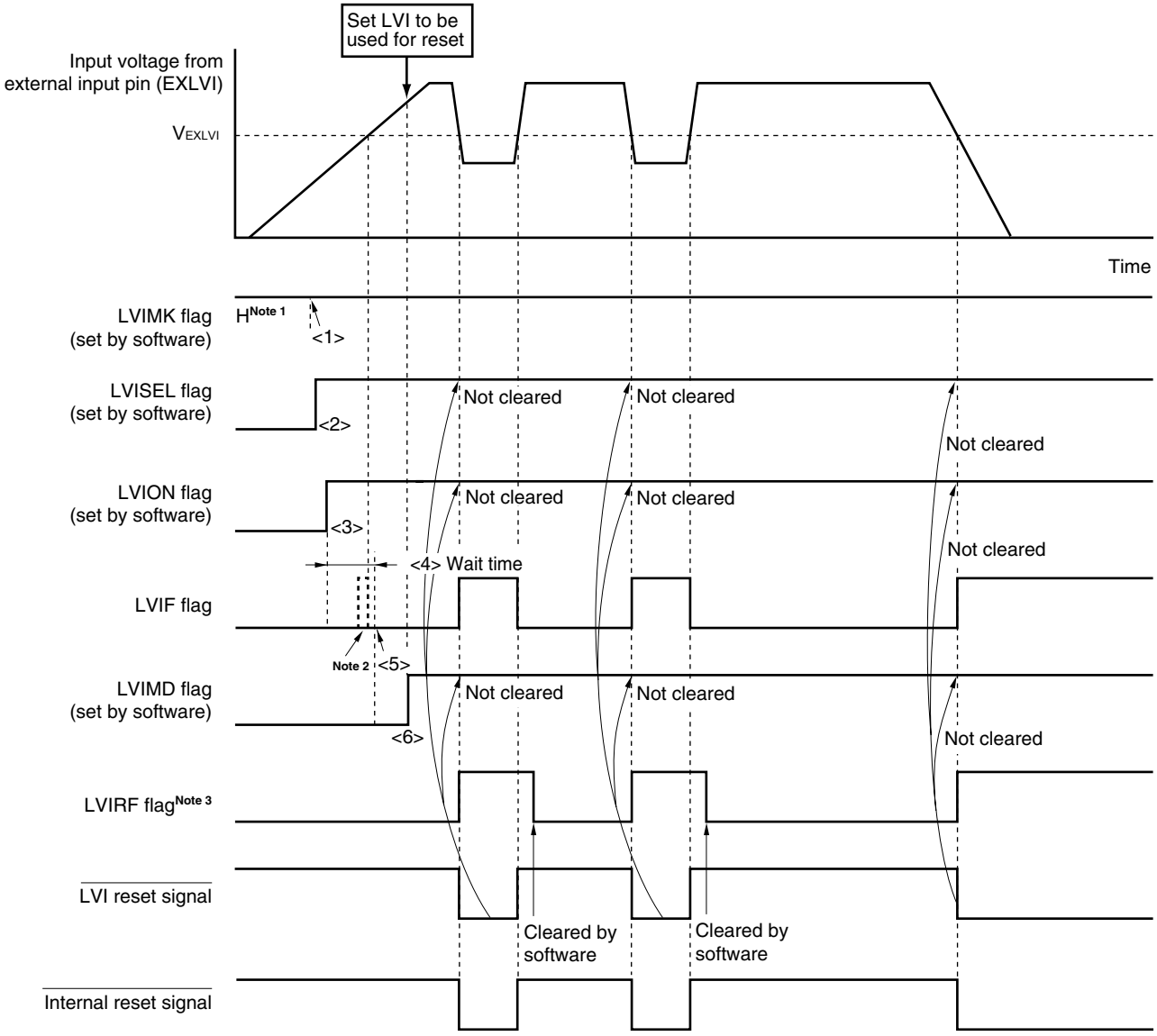
  2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.

  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 21-7. 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 19 RESET FUNCTION.

Remark <1> - <6> in Figure 21-7 above correspond to <1> - <6> in the description of "When starting operation" in 21.4.1 (2) When detecting level of input voltage from external input pin (EXLVI).

21.4.2 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).

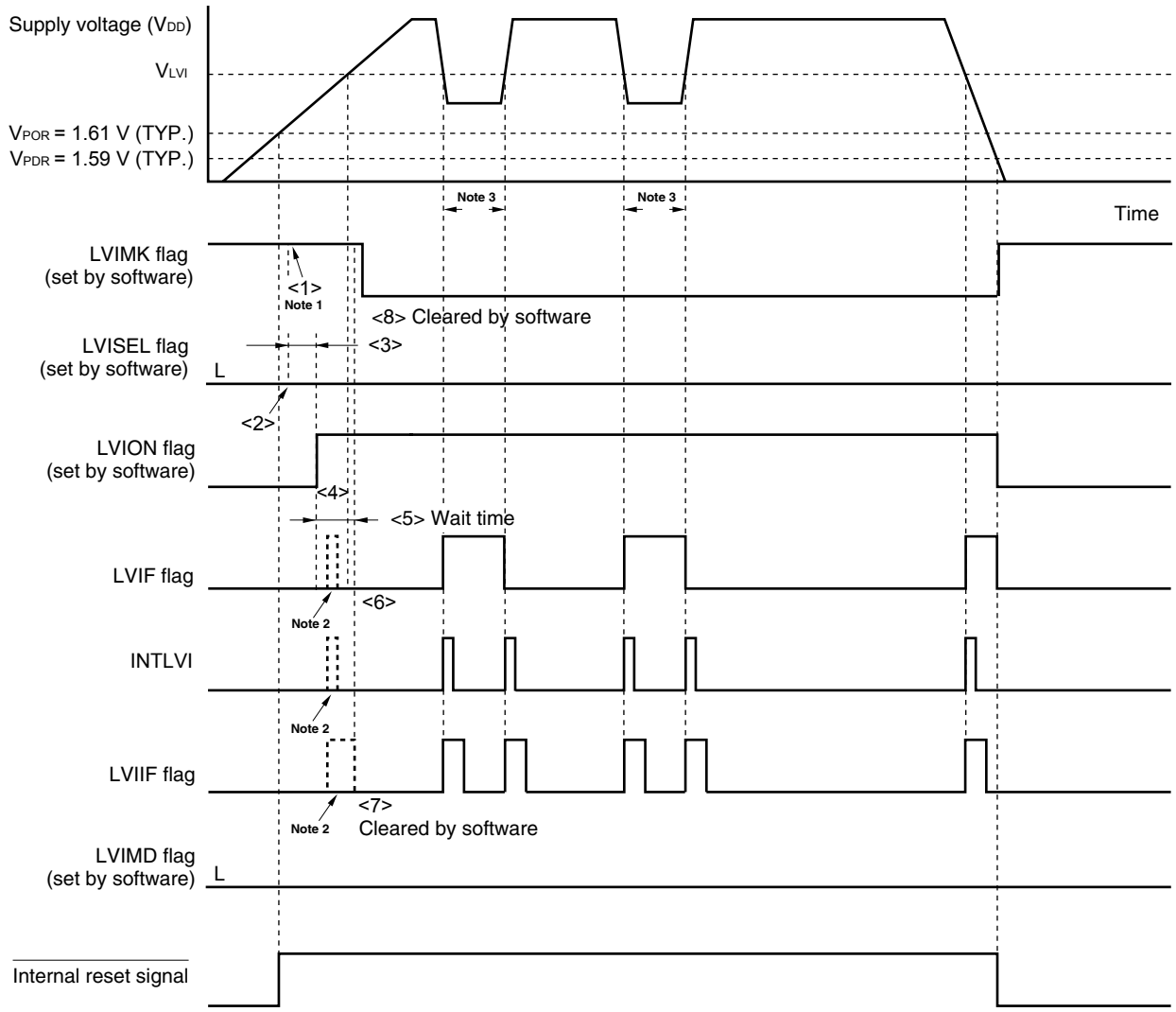
<9> Execute the EI instruction (when vector interrupts are used).

Figure 21-8 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 21-8. 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> - <8> in Figure 21-8 above correspond to <1> - <8> in the description of "When starting operation" in 21.4.2 (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 from the state of "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 21-9 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

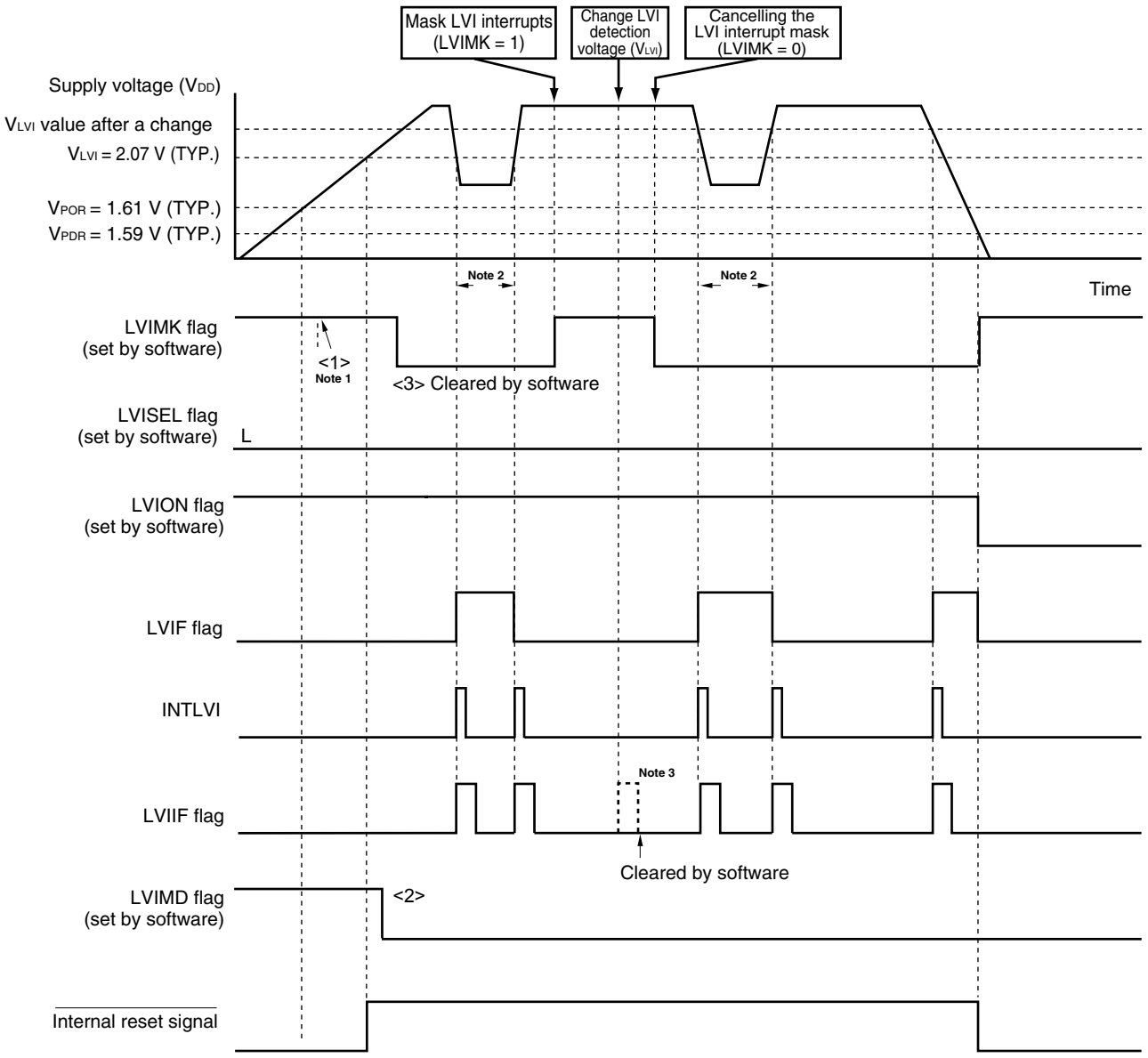
Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

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 19 RESET FUNCTION.

Figure 21-9. 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> - <3> in Figure 21-9 above correspond to <1> - <3> in the description of "When starting operation" in 21.4.2 (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)

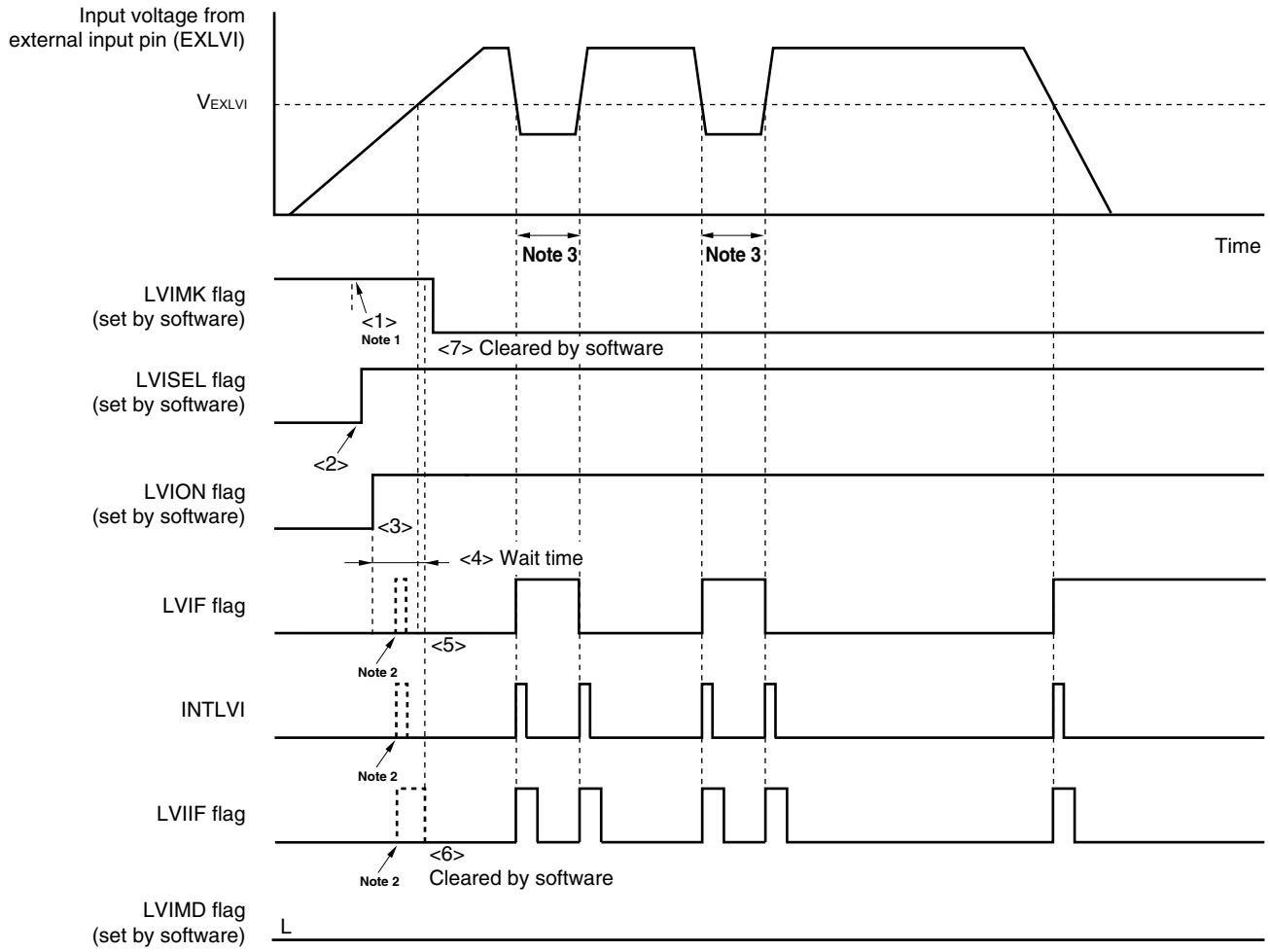
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (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) ≥ 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 21-10 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> - <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 21-10. 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 21-10 above correspond to <1> to <7> in the description of "When starting operation" in 21.4.2 (2) When detecting level of input voltage from external input pin (EXLVI).

21.5 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 21-11**).

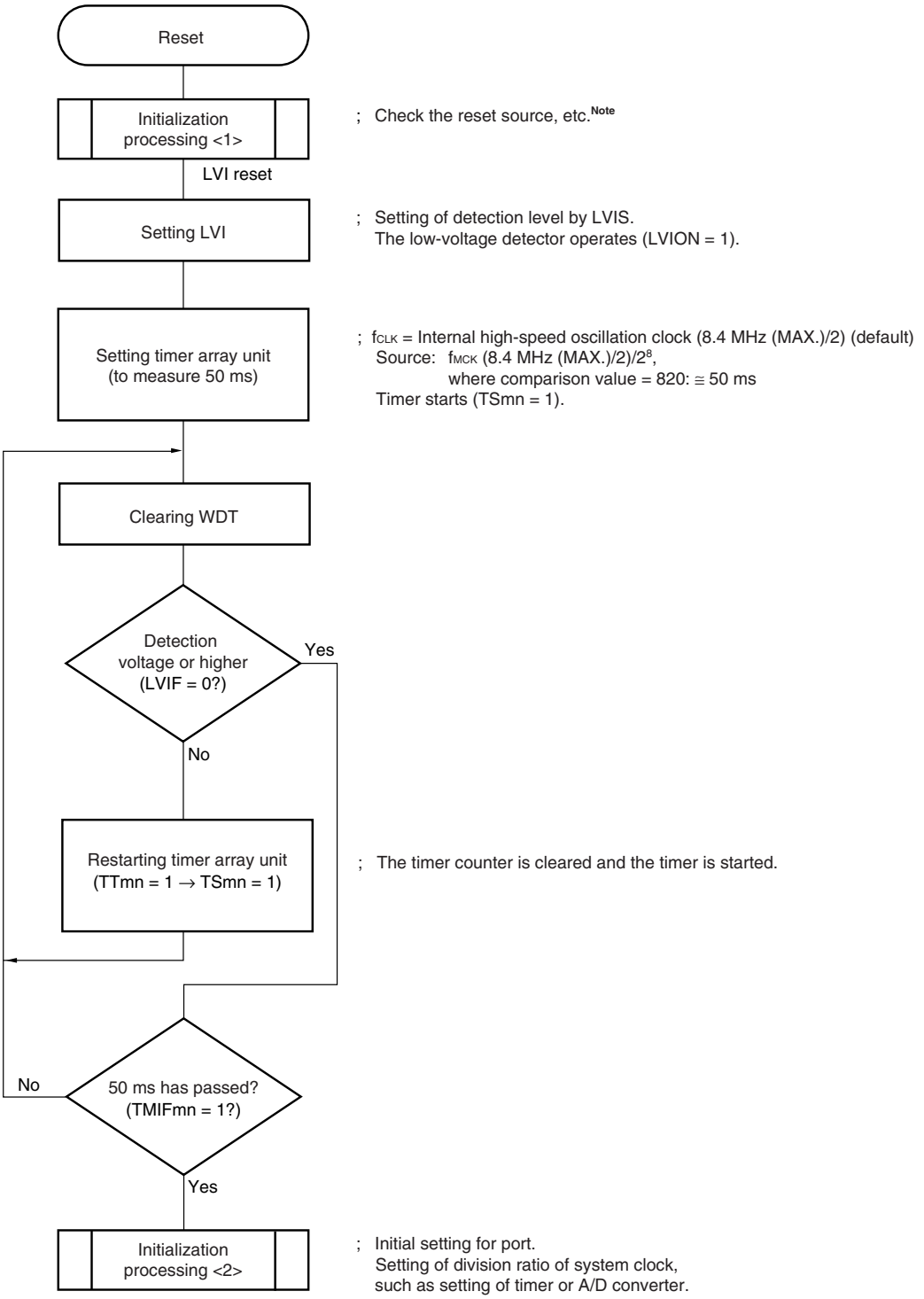
Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- **Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)**
- **Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)**

<R>

Figure 21-11. 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.

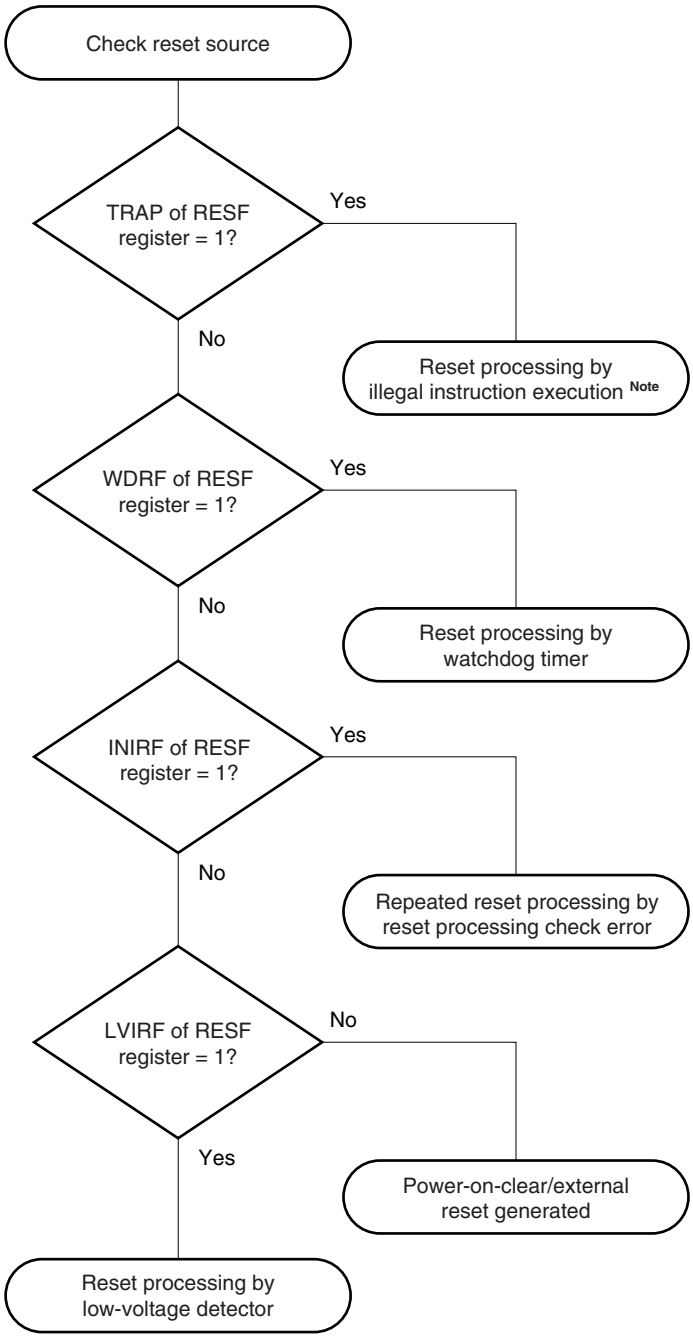
Remarks 1. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- **Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)**
- **Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)**

2. m = 0, 1, n = 0 to 7, mn = 00-07, 10-13

<R> **Figure 21-11. 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.

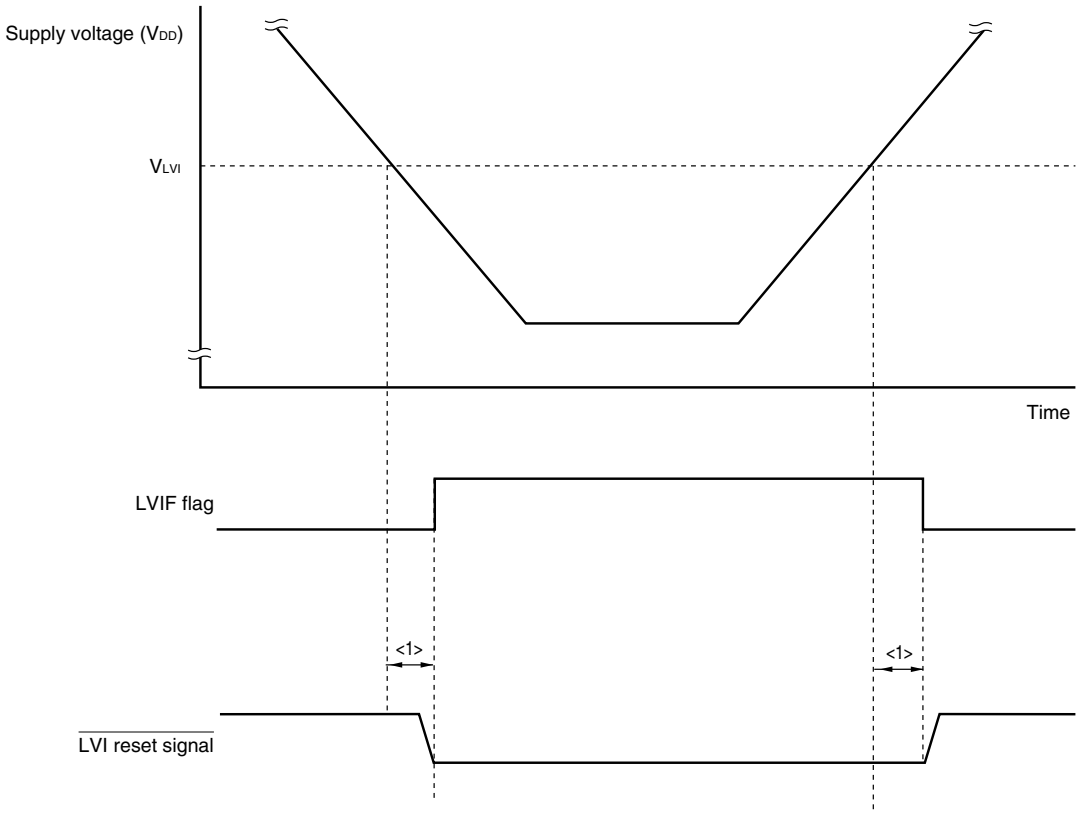
- **Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)**
- **Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)**

(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 21-12**).

Figure 21-12. 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 22 REGULATOR

22.1 Regulator Overview

The 78K0R/KC3-L, KE3-L contains a circuit for operating the device with a constant voltage. At this time, in order to
<R> 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 consumption current mode, 1.8 V (typ.).

22.2 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 22-1. 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 consumption current mode (1.8 V)
00H	Switches normal current mode (2.4 V) and low consumption current mode (1.8 V) according to the condition (refer to **Table 21-1**)
Other than above	Setting prohibited

Cautions 1. When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases.

<When X1 clock is selected as the CPU clock>

$f_X \leq 5$ MHz and $f_{CLK} \leq 1$ MHz

<When the internal high-speed oscillation clock, external input clock, or subsystem clock are selected for the CPU clock>

$f_{CLK} \leq 1$ MHz

(Caution is given on the next page.)

Caution 2. A wait is required to change the operation speed mode control register (OSMC) after changing the RMC register. Wait for 3.5 ms by software when setting to low consumption current mode and 10 μs when setting to normal current mode, as described in the procedure shown below.

- **When setting to low consumption current mode**
  - **<1> Select a frequency of 1 MHz for f_{CLK}.**
  - **<2> Set RMC to 5AH (set the regulator to low consumption current mode).**
  - **<3> Wait for 2 ms.**
  - **<4> Set FLPC and FSEL of OSMC to 1 and 0, respectively.**

- **When setting to normal current mode**
  - **<1> Set RMC to 00H (set the regulator to normal current mode).**
  - **<2> Wait for 10 μs.**
  - **<3> Change FLPC and FSEL of OSMC.**
  - **<4> Change the f_{CLK} frequency.**

Table 22-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low consumption current mode	1.8 V	In STOP mode (except during OCD mode)
		When both the high-speed system clock (f_{MX}), the internal high-speed 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_{XT})
		When both the high-speed system clock (f_{MX}), the internal high-speed 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_{XT}) has been set
Normal current mode	2.4 V	Other than above

CHAPTER 23 OPTION BYTE

23.1 Functions of Option Bytes

Addresses 000C0H -000C3H of the flash memory of the 78K0R/KC3-L, KE3-L form an option byte area.

Option bytes consist of user option byte (000C0H - 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 - 000C3H are replaced by 010C0H - 010C3H. Therefore, set the same values as 000C0H - 000C3H - 010C0H - 010C3H.

Caution Be sure to set FFH - 000C2H (000C2H/010C2H when the boot swap operation is used).

23.1.1 User option byte (000C0H - 000C2H/010C0H - 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 - 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 - 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 - 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

23.1.2 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 - 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

23.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 23-1. Format of User Option Byte (000C0H/010C0H)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time 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)

<R>

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 34.5 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.71 ms)
0	0	1	$2^8/f_{IL}$ (7.42 ms)
0	1	0	$2^9/f_{IL}$ (14.84 ms)
0	1	1	$2^{10}/f_{IL}$ (29.68 ms)
1	0	0	$2^{12}/f_{IL}$ (118.72 ms)
1	0	1	$2^{14}/f_{IL}$ (474.90 ms)
1	1	0	$2^{15}/f_{IL}$ (949.80 ms)
1	1	1	$2^{17}/f_{IL}$ (3799.19 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode[Note 2]
1	Counter operation enabled in HALT/STOP mode

Figure 23-1. Format of User Option Byte (000C0H/010C0H)

Notes 1. Set the same value as 000C0H - 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 23-2. 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 - 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 highspeed 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 23-3. 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 - 000C2H, as these addresses are reserved areas. Also set FFH - 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

23.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 23-4. Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debug operation. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debug operation. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H - 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.

23.4 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
      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
      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 - 000C3H is switched to 010C0H - 010C3H. Describe to 010C0H - 010C3H, therefore, the same values as 000C0H - 000C3H as follows.

```
OPT2  CSEG   AT   010C0H
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^10/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      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 - 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 24 FLASH MEMORY

The 78K0R/KC3-L, KE3-L incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

24.1 Writing with Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the 78K0R/KC3-L and KE3-L.

- PG-FP5, FL-PR5
- QB-MINI2

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/KC3-L, KE3-L has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0R/KC3-L, KE3-L is mounted on the target system.

Remark FA series is product of Naito Densei Machida Mfg. Co., Ltd.

Table 24-1. Wiring Between 78K0R/KC3-L, KE3-L and Dedicated Flash Memory Programmer

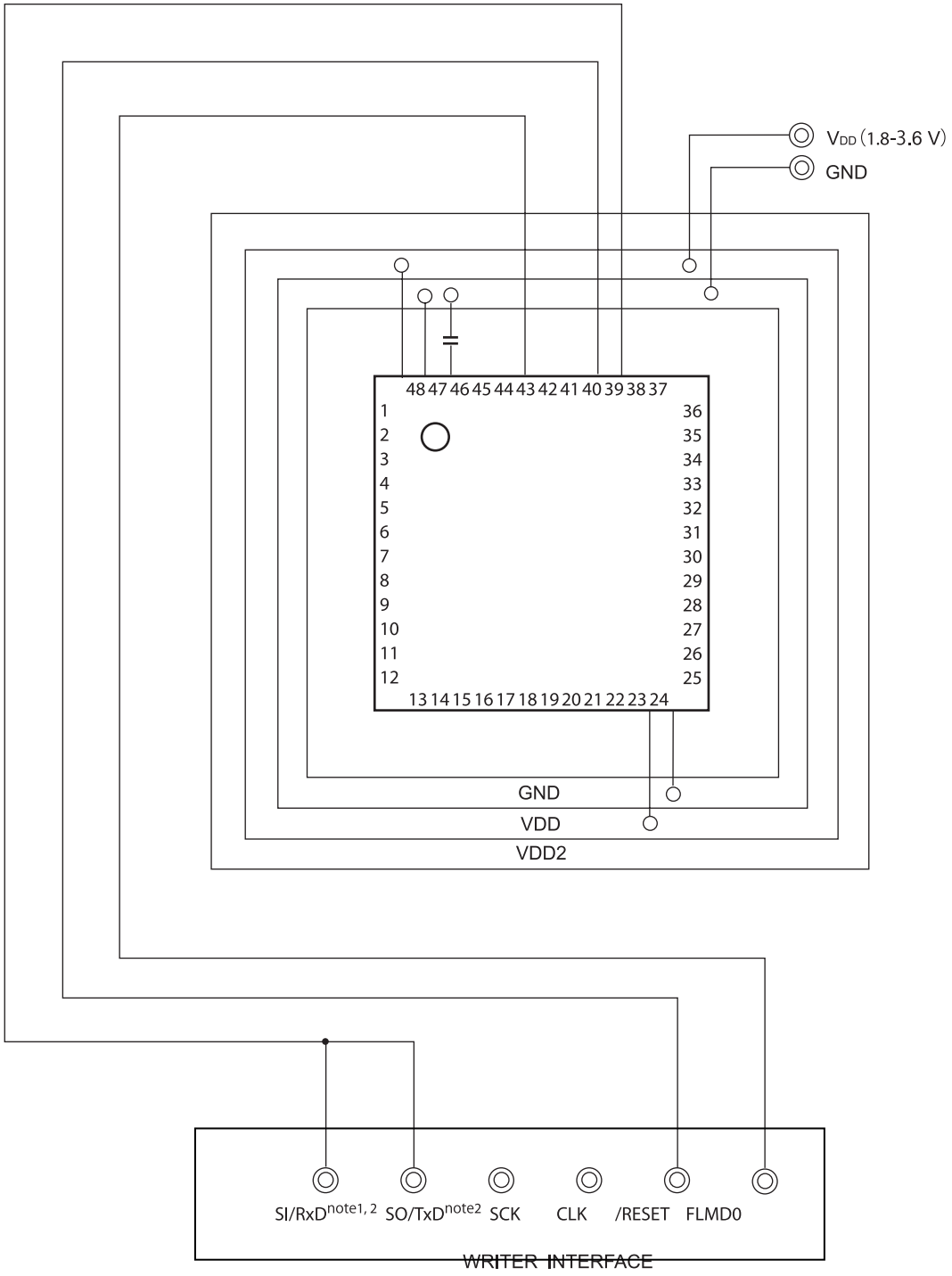
Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No.	
				KC3-L 48 pin	KE3-L 64 pin
Signal Name	I/O	Pin Function		TQFP (7x7)	LQFP (10x10), TQFP (7x7), FBGA (5x5)
SI/RxD Notes 1, 2	Input	Receive signal	TOOL0/P40	39	9
SO/TxD Note 2	Output	Transmit signal			
SCK	Output	Transfer clock	–	–	–
CLK	Output	Clock output	–	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	40	10
FLMD0	Output	Mode signal	FLMD0	43	13
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	48	19
			EV_{DD}		20
			AV_{REF}	23	59
GND	–	Ground	V_{SS}	47	17
			EV_{SS}		18
			AV_{SS}	24	60

Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

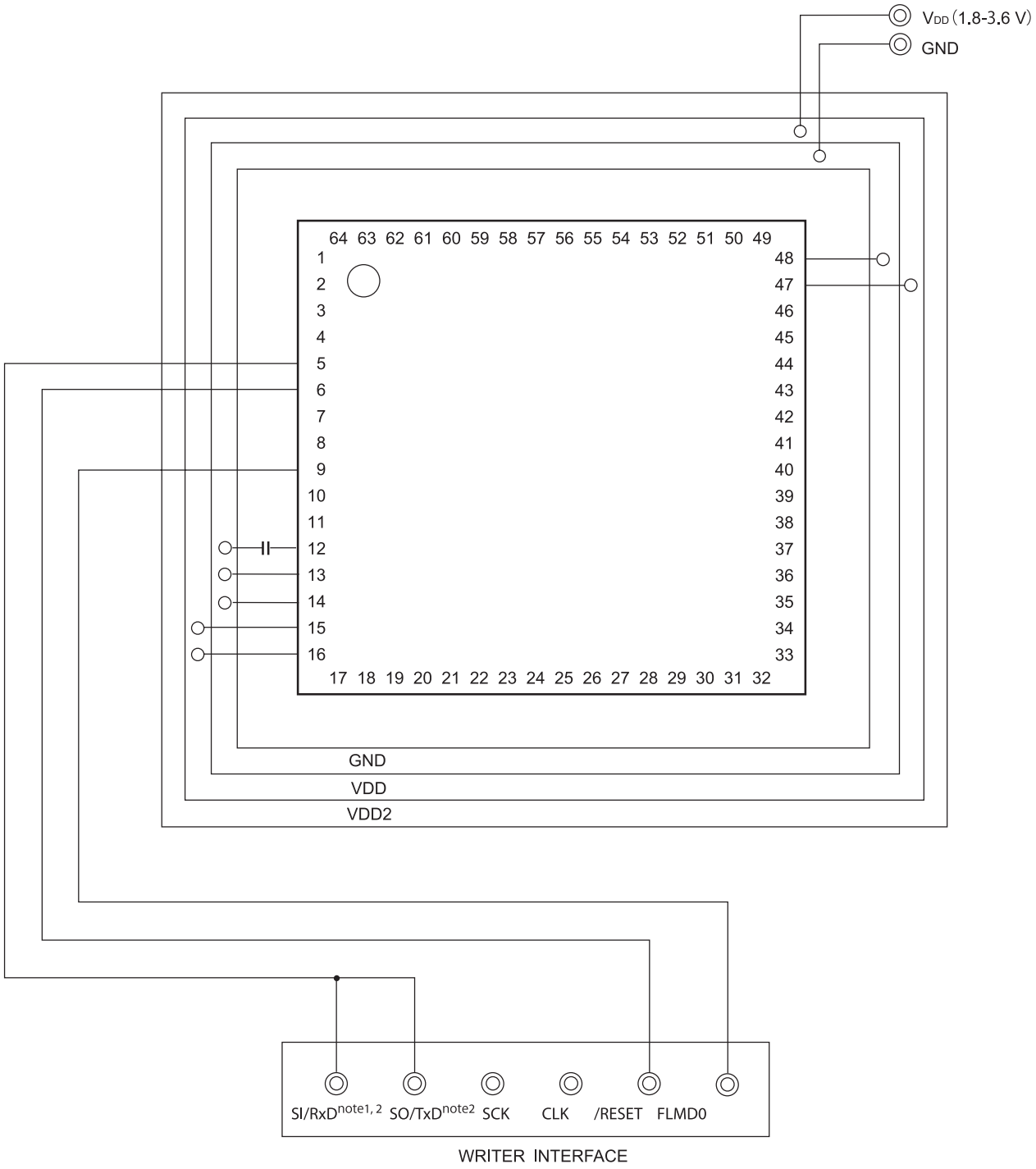
Figure 24-1. Example of Wiring Adapter for Flash Memory Writing (48 Pin product of 78K0R/KC3-L)



Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

Figure 24-2. Example of Wiring Adapter for Flash Memory Writing (64 Pin product of 78K0R/KE3-L)



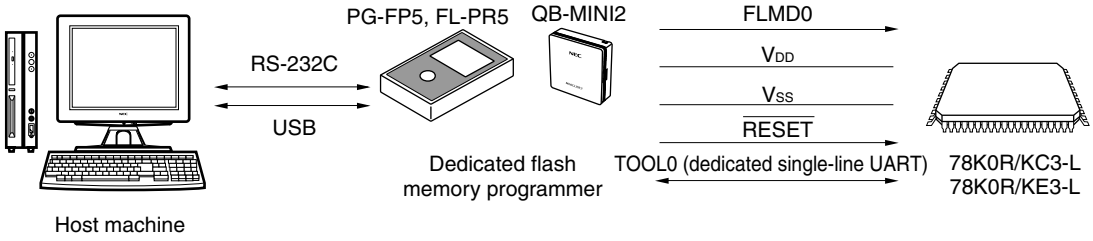
Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

24.2 Programming Environment

The environment required for writing a program to the flash memory of the 78K0R/KC3-L, KE3-L is illustrated below.

Figure 24-3. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

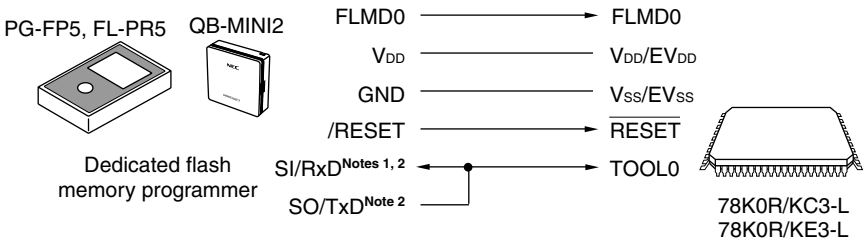
To interface between the dedicated flash memory programmer and the 78K0R/KC3-L, KE3-L, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

24.3 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0R/KC3-L, KE3-L is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the 78K0R/KC3-L, KE3-L.

Transfer rate: 115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps[Note 1]

Figure 24-4. Communication with Dedicated Flash Memory Programmer



<R> **Notes 1.** When using a transfer rate of 1 Mbps, do not use the wide voltage mode.
2. This pin is not required to be connected when using PG-FP5 or FL-PR5.
3. Connect SI/RxD or SO/TxD when using QB-MINI2.

The dedicated flash memory programmer generates the following signals for the 78K0R/KC3-L, KE3-L. See the manual of PG-FP5, FL-PR5, or MINICUBE2 for details.

Table 24-2. Pin Connection

Dedicated Flash Memory Programmer			78K0R/KC3-L, KE3-L	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_{REF}	◎
GND	–	Ground	V_{SS}, EV_{SS}, AV_{SS}	◎
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI/RxD Notes 1, 2	Input	Receive signal	TOOL0	◎
SO/TxD Note 2	Output	Transmit signal		
SCK	Output	Transfer clock	–	×

Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.
2. Connect SI/RxD or SO/TxD when using QB-MINI2.

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

24.4 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.

24.4.1 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 **24.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

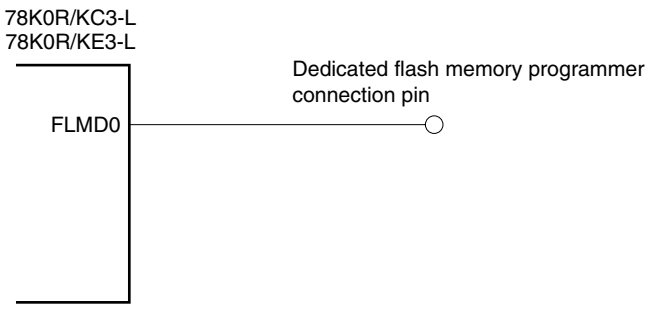
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(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 24-5. FLMD0 Pin Connection Example



24.4.2 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 EV_{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).

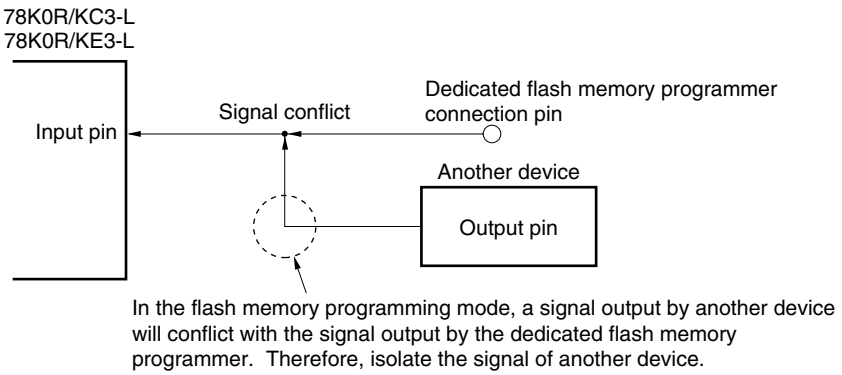
Remark The SAU and IICA pins are not used for communication between the 78K0R/KC3-L, KE3-L and dedicated flash memory programmer, because single-line UART is used.

24.4.3 $\overline{\text{RESET}}$ 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 24-6. Signal Conflict ($\overline{\text{RESET}}$ Pin)



24.4.4 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.

24.4.5 REGC pin

<R> 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.

24.4.6 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.

24.4.7 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_{REF}, and AV_{SS}) as those in the normal operation mode.

Please set QB-MINI2 power supply switch to 3 or T before use.

24.5 Registers that Control 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 24-7. 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

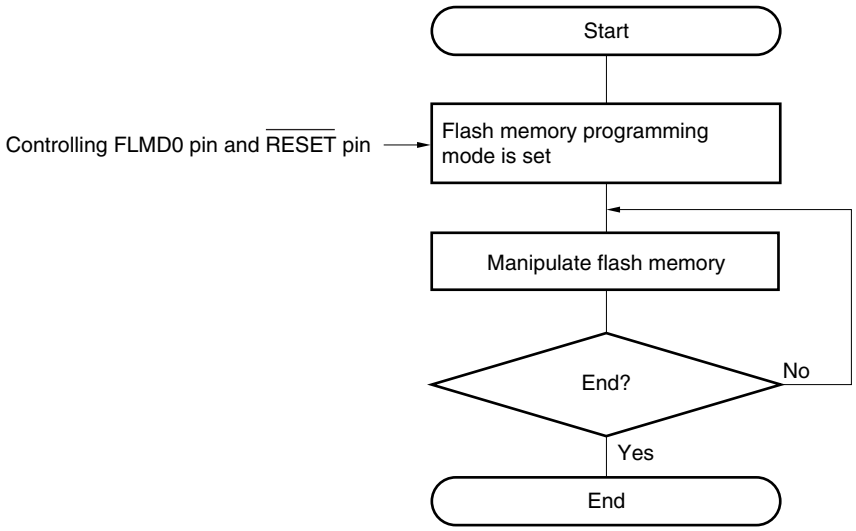
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

24.6 Programming Method

24.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 24-8. Flash Memory Manipulation Procedure



24.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0R/KC3-L, KE3-L 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 24-9. Flash Memory Programming Mode

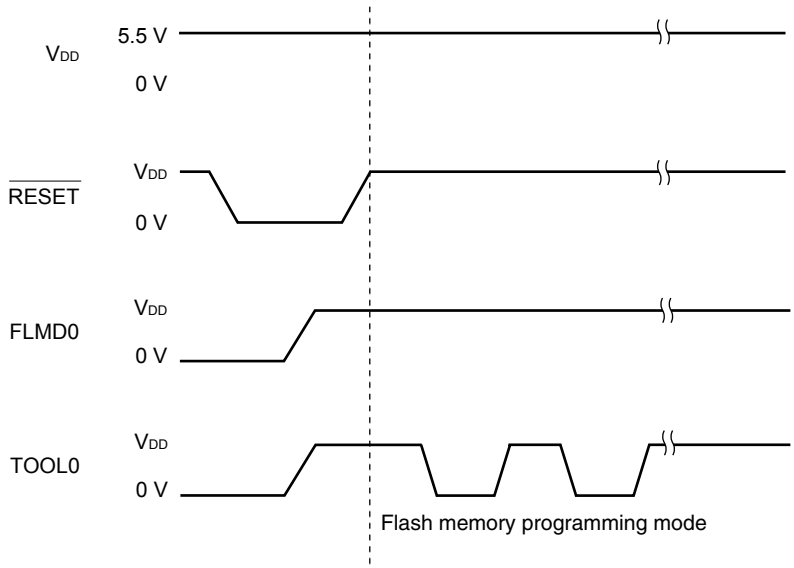


Table 24-3. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0 V	Normal operation mode
V_{DD}	Flash memory programming mode

In flash memory programming mode, the executable voltage range for write/ erase/ verify can differ in the following two modes.

Table 24-4. Programming mode and Executable supply of write/erase/ verify

Mode	Executable supply of write/erase/ verify	Writing Clock Frequency
Wide voltage mode	1.8 V-3.6 V	4 MHz (MAX.)
Full speed mode	2.7 V-3.6 V	20 MHz (MAX.)

Please set either of modes after matching with the write voltage range. To set wide voltage mode, full speed mode, use the self Flash Memory Programmer GUI.

Caution If erase is performed in wide voltage mode, then write and verify is possible only in wide voltage mode. However, if the area erased in wide voltage mode is to be erased again in the full speed mode, then, write/ verify can be performed only in full speed mode.

Remark See 24.6.4 Communication command for communication command details.

24.6.3 Selecting communication mode

Communication mode of the 78K0R/KC3-L, KE3-L is as follows.

Table 24-5. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (single-line UART)	UART	115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps[Note 3]	–	–	TOOL0

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

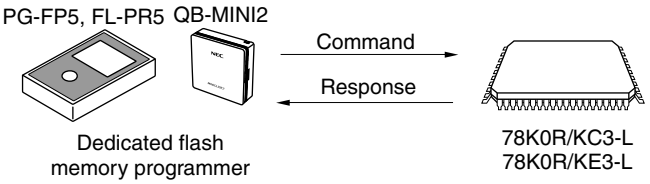
2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

<R> **3.** When using a transfer rate of 1 Mbps, do not use the wide voltage mode.

24.6.4 Communication commands

The 78K0R/KC3-L, KE3-L communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/KC3-L, KE3-L are called commands, and the signals sent from the 78K0R/KC3-L, KE3-L to the dedicated flash memory programmer are called response.

Figure 24-10. Communication Commands



The flash memory control commands of the 78K0R/KC3-L, KE3-L are listed in the table below. All these commands are issued from the programmer and the 78K0R/KC3-L, KE3-L perform processing corresponding to the respective commands.

Table 24-6. 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/KC3-L, KE3-L information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0R/KC3-L, KE3-L 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/KC3-L, KE3-L returns a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0R/KC3-L, KE3-L are listed below.

Table 24-7. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

24.7 Security Settings

The 78K0R/KC3-L, KE3-L supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. 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 block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting. In addition, execution of the batch erase (chip erase) command.

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 24-7 shows the relationship between the erase and write commands when the 78K0R/KC3-L, KE3-L security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see 24.8.2 for details).

Table 24-8. 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 shield window function (see 24.8.2 for details).

Table 24-9. 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		

24.8 Flash Memory Programming by Self-Programming

The 78K0R/KC3-L, KE3-L supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the 78K0R/KC3-L, KE3-L 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.

<R> **4. Disable DMA operation (DENn = 0) during the execution of self programming library functions.**

<R> **Remarks 1.** For details of the self-programming function and the 78K0R/KC3-L and KE3-L self-programming library, refer to **78K0R Microcontroller Self Programming Library Type2 User's Manual (U19193E).**

2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

In self programming function, when writing in flash memory programming, the executable voltage range for write/ erase/ verify can differ in the following two modes.

Table 24-11. Programming mode and Executable supply of write/erase/ verify

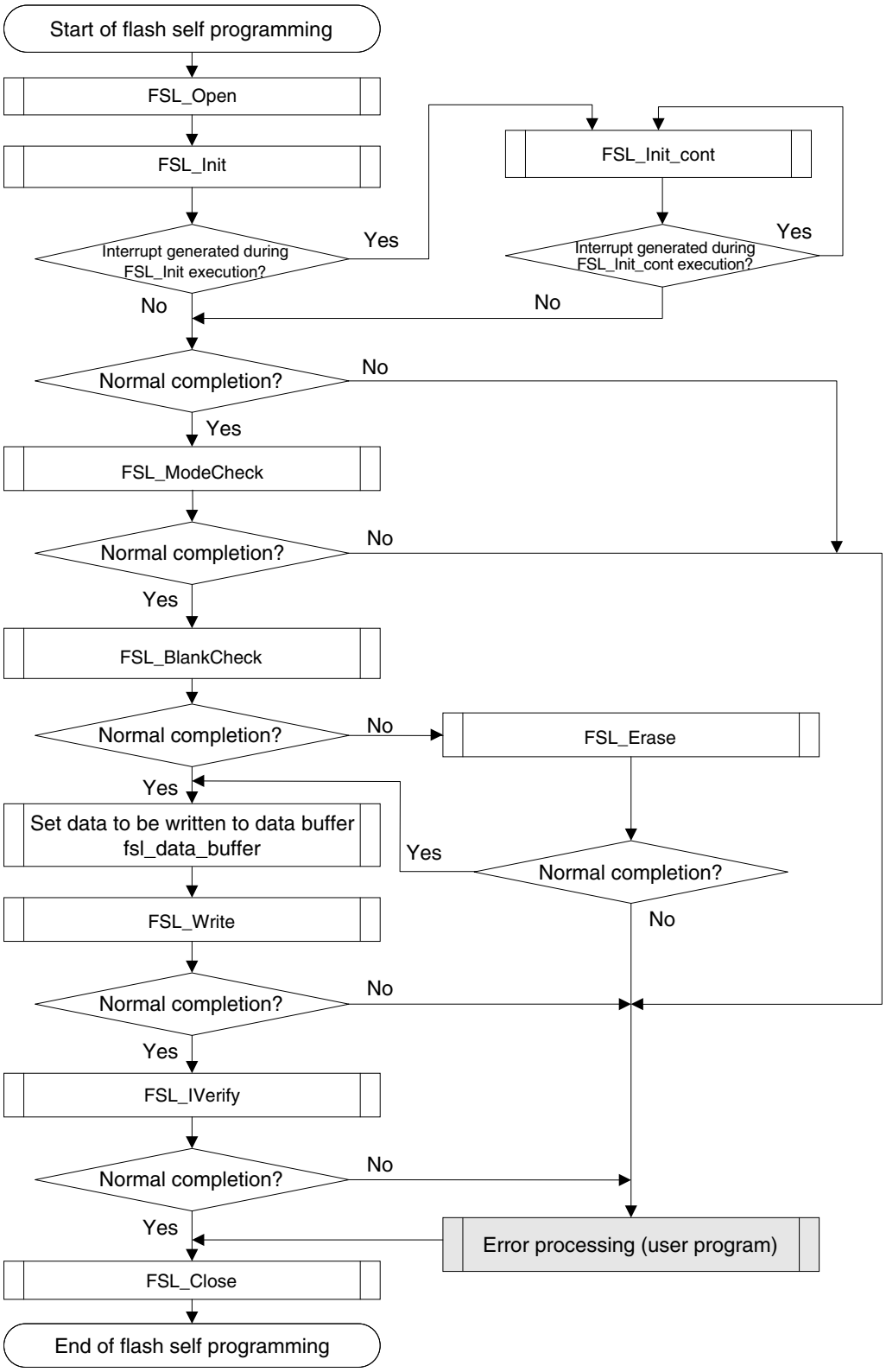
Mode	Executable supply of write/erase/ verify	Writing Clock Frequency
Wide voltage mode	1.8 V-3.6 V	4 MHz (MAX.)
Full speed mode	2.7 V-3.6 V	20 MHz (MAX.)

Please set either of modes after matching with the write voltage range. While setting modes in order to execute the "FSL_Init" constant from Renesas self programming library, use wide voltage mode If function "fsl_low_voltage_u08" is 01H, and full speed mode if it is 00H.

Caution If erase is performed in wide voltage mode, then write and verify is possible only in wide voltage mode. However, if the area erased in wide voltage mode is to be erased again in the full speed mode, then, write/ verify can be performed only in full speed mode.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

<R> **Figure 24-11. Flow of Self Programming (Rewriting Flash Memory)**



24.8.1 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.

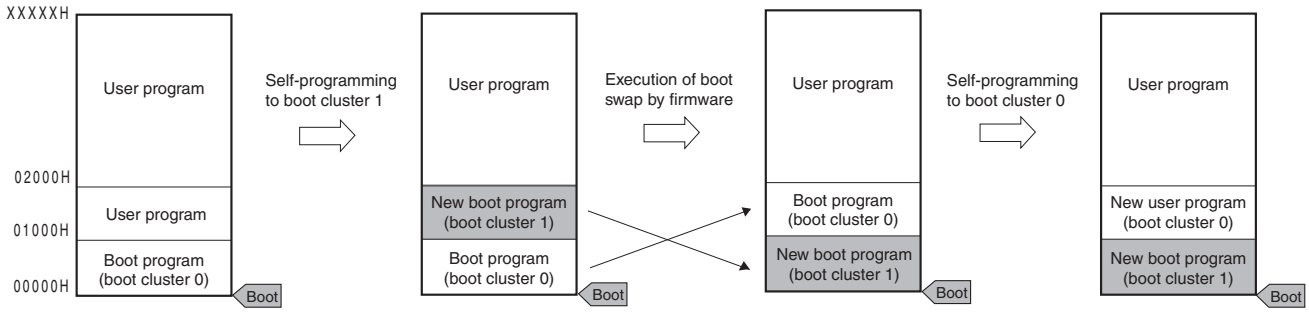
The boot swap function is used to avoid this problem.

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/KC3-L, KE3-L, 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 24-12. Boot Swap Function

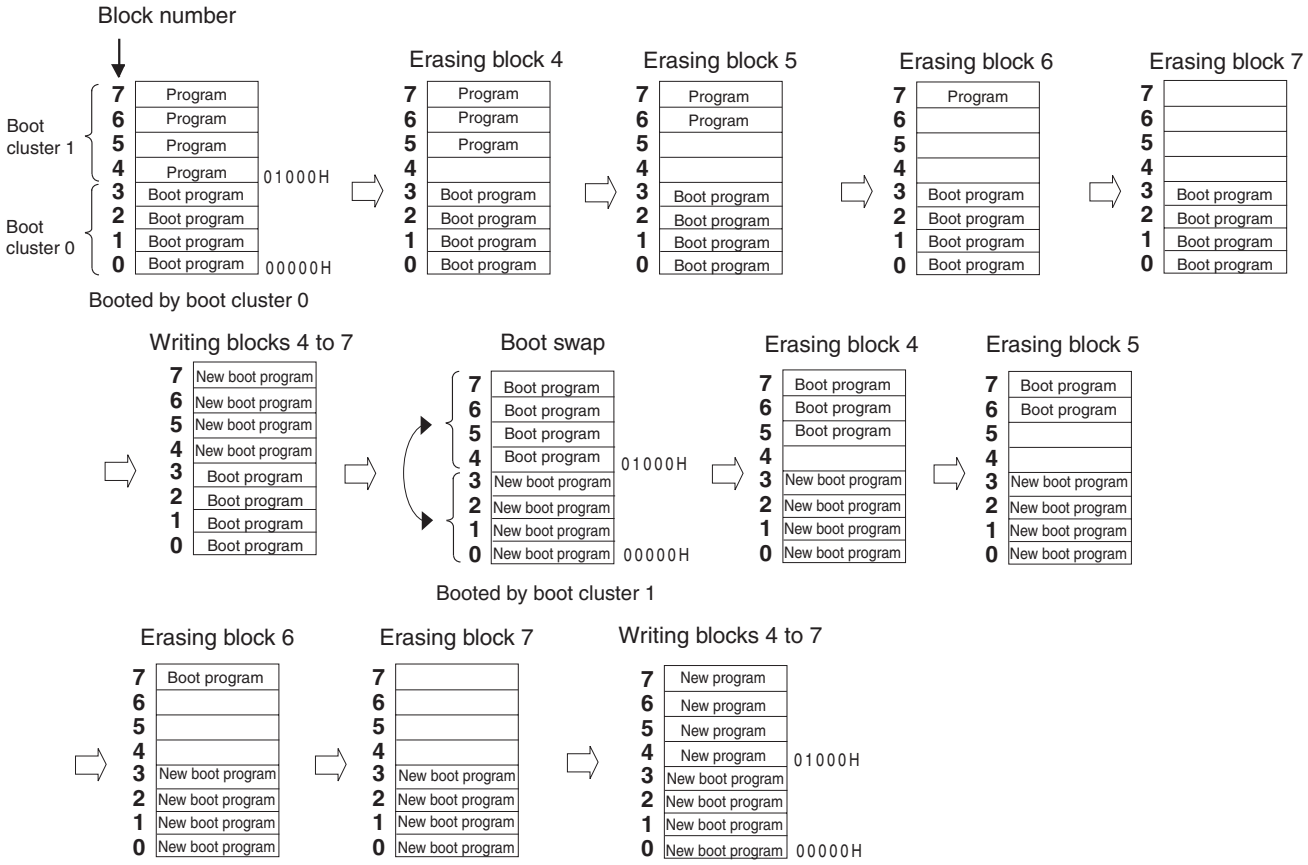


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 24-13. Example of Executing Boot Swapping



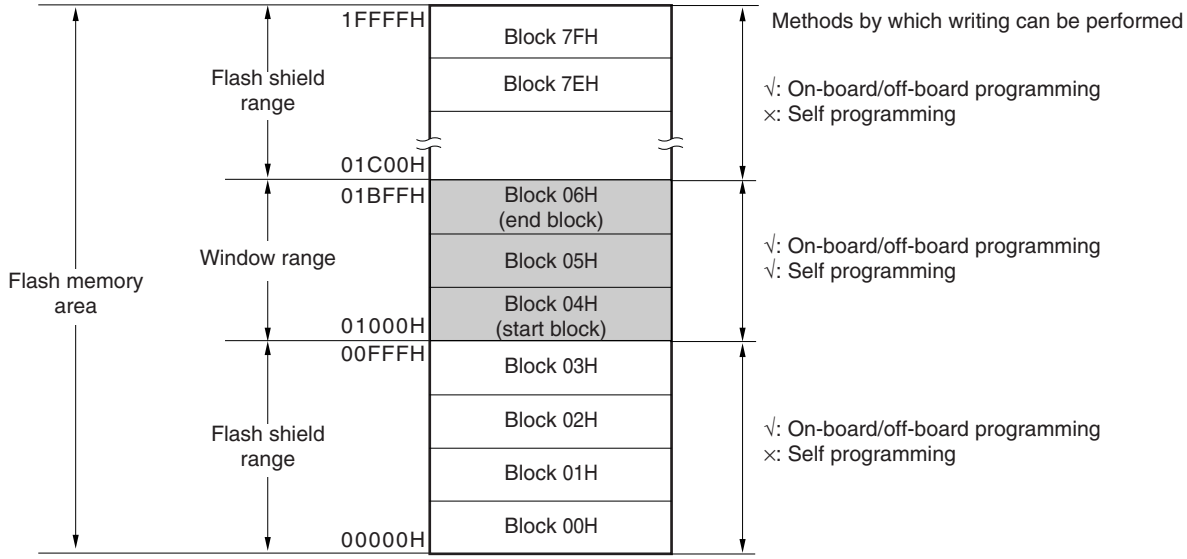
24.8.2 Flash shield window function

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.

Figure 24-14. Flash Shield Window Setting Example
(Target Devices: μPD78F1024, 78F1026, 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 24-9. 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 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 24.7 Security Settings to prohibit writing/erasing during on-board/off-board programming.

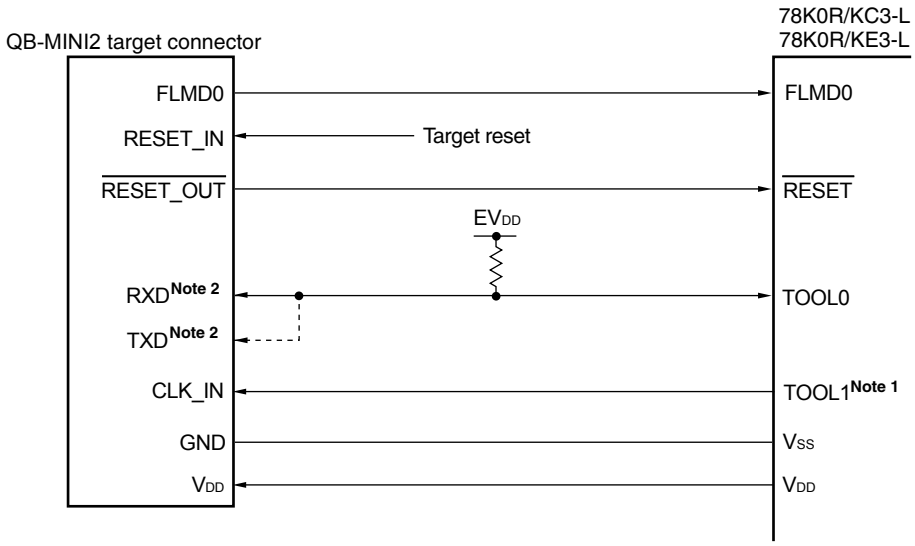
CHAPTER 25 ON-CHIP DEBUG FUNCTION

25.1 Connecting QB-MINI2 to 78K0R/KC3-L, KE3-L

The 78K0R/KC3-L, KE3-L uses the V_{DD}, FLMD0, $\overline{\text{RESET}}$, 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/KC3-L, KE3-L has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 25-1. Connection Example of QB-MINI2 and 78K0R/KC3-L, KE3-L



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.

Caution 1 .When communicating in 2-line mode, a clock with a frequency of half that of the CPU clock frequency is output from the TOOL1 pin. A resistor or ferrite bead can be used as a countermeasure against fluctuation of the power supply caused by that clock.

2. Please set the switch to 3 or T at the time of QB-MINI2 connection and then set power supply either to 3 V or to no power supply. Please set the switch to T when using USB communication, and then supply the power from 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 25-1 lists the differences between 1-line mode and 2-line mode.

<R> **Table 25-1. Differences Between 1-Line Mode and 2-Line Mode**

Communication Mode	Flash Memory Programming Function
1-line mode	Available
2-line mode	None

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.

25.2 On-Chip Debug Security ID

The 78K0R/KC3-L, KE3-L has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 23 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 - 010CDH in advance, because 000C3H, 000C4H - 000CDH and 010C3H, and 010C4H - 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 25-2. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

25.3 Securing of User Resources

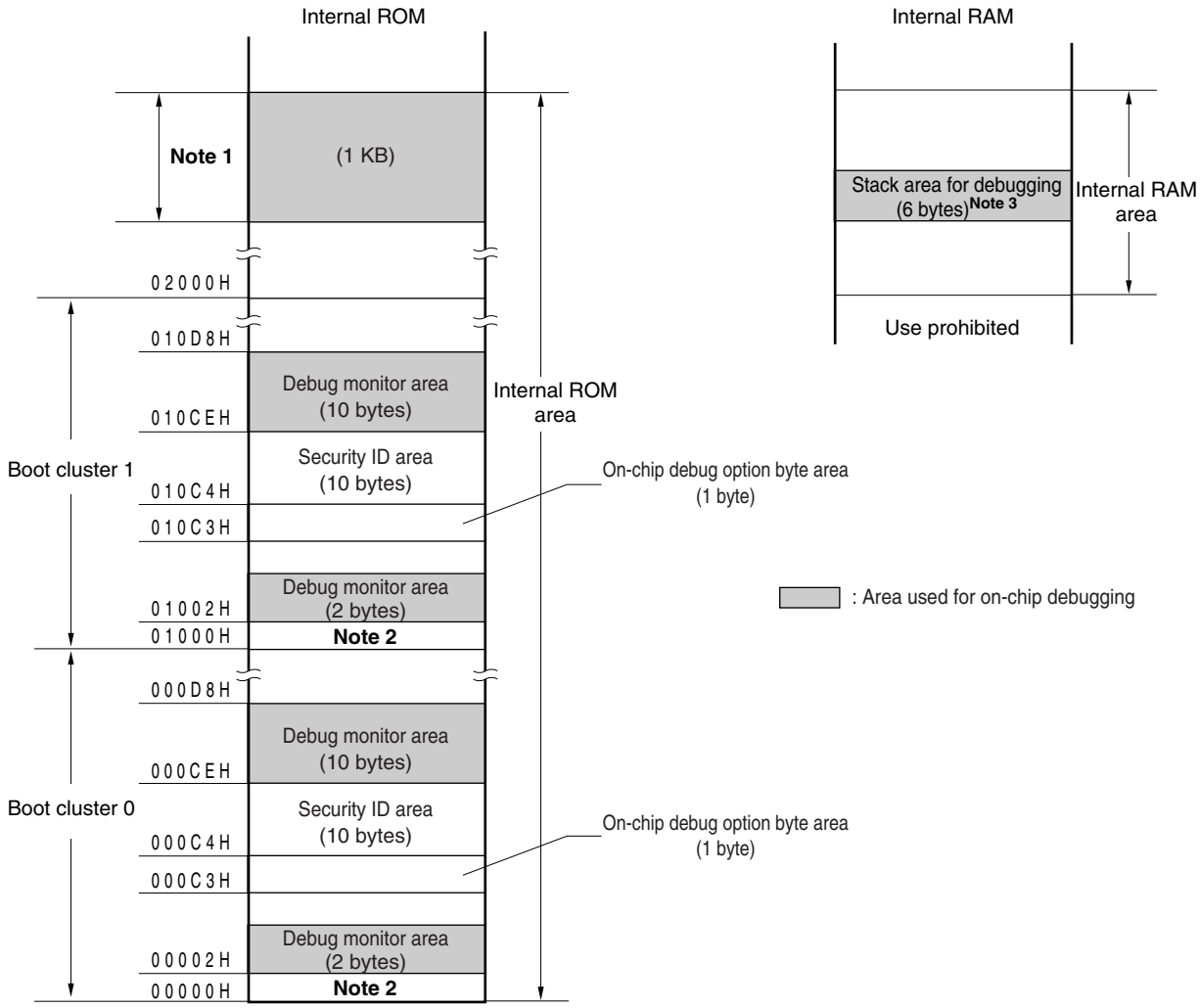
To perform communication between the 78K0R/KC3-L, KE3-L and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

(1) Securing of memory space

The shaded portions in Figure 25-2 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 25-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products	Internal ROM	Address
μPD78F1022	64 KB	0FC00H to 0FFFFH
μPD78F1023, 78F1025	96 KB	17C00H to 17FFFH
μPD78F1024, 78F1026	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 26 BCD 10 CORRECTION CIRCUIT

26.1 BCD 10 Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

26.2 Registers Used by BCD 10 Correction Circuit

The BCD 10 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 26-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

26.3 BCD 10 Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 1: 99 + 89 = 188

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #99H ; <1>`	99H	–	–	–
`ADD A, #89H ; <2>`	22H	1	1	66H
`ADD A, !BCDADJ ; <3>`	88H	1	0	–

Examples 2: 85 + 15 = 100

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #85H ; <1>`	85H	–	–	–
`ADD A, #15H ; <2>`	9AH	0	0	66H
`ADD A, !BCDADJ ; <3>`	00H	1	1	–

Examples 3: 80 + 80 = 160

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #80H ; <1>`	80H	–	–	–
`ADD A, #80H ; <2>`	00H	1	0	60H
`ADD A, !BCDADJ ; <3>`	60H	1	0	–

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 − 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	–	–	–
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	–

CHAPTER 27 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 27-5 Operation List indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

27.1 Conventions Used in Operation List

27.1.1 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 27-1 below, R0, R1, R2, etc.) can be used for description.

Table 27-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbol (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note]) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See Table 3-5 SFR List for the symbols of the special function registers.
The extended special function registers can be described to operand !addr16 as symbols. See Table 3-6 Extended SFR (2nd SFR) List for the symbols of the extended special function registers.

27.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 27-2. 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
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

27.1.3 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 27-3. 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

27.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

An interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 27-4. Use Example of PREFIX Operation Code

Instruction	Operation code				
	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.

27.2 Operation List

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clock plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clock plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

3. cnt indicates the bit shift count.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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 SFR area or extended 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 (fCLK) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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			
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. When the internal RAM area, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 27-5. 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	
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0	
	BTCLR	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit	
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
		ES:[HL].bit, $addr20	4	4/6[Note 3]	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit	
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1	
	SKNC	–	2	1	–	Next instruction skip if CY = 0	
	SKZ	–	2	1	–	Next instruction skip if Z = 1	
	SKNZ	–	2	1	–	Next instruction skip if Z = 0	
	SKH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 0	
	SKNH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 1	
CPU control	SEL	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, SFR area or extended 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_{CLK}) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

3. n indicates the number of register banks (n = 0 to 3)

CHAPTER 28 ELECTRICAL SPECIFICATIONS

Cautions 1. The 78K0R/KC3-L, KE3-L has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. Pins provided for products are as given below.

(1) Port functions

Port	78K0R/KC3-L	78K0R/KE3-L
Port0	P02-P04	
Port1	P10-P17	
Port2	P20-P27	
Port3	P31	
Port4	P40, P41	P40-P43
Port5	P50, P51	P50-P53
Port6	P60-P63	
Port7	P70-P73	P70-P77
Port11	–	P110, P111
Port12	P120-P124	
Port13	–	P130
Port14	P140	P140, P142-P144

(2)Non- port Pins(1/2)

Function	78K0R/KC3-L	78K0R/KE3-L
Supply, Grand	V_{DD}, EV_{DD}, AV_{REF}, V_{SS}, EV_{SS}, AV_{SS}	
Regulator	REGC	
Reset	$\overline{\text{RESET}}$	
Clock oscillation	X1, X2, XT1, XT2, EXCLK	
Write Flash	FLMD0	
Interrupt	INTP0-INTP6	INTP0-INTP6, INTP8-INTP11
Timer	TI01-TI03, TO01-TO03	TI00-TI04, TO00-TO04
Real time counter	RTCDIV, RTCCL	

(2)Non-port Pins(2/2)

Function		78K0R/KC3-L	78K0R/KE3-L
Serial interface	UART0	RxD0, TxD0	
	UART1	RxD1, TxD1	
	UART2	–	RxD2, TxD2
	UART3	RxD3, TxD3	
	CSI00	SCK00, SI00, SO00	
	CSI10	SCK10, SI10, SO10	
	CSI20	–	SCK20, SI20, SO20
	IIC10	SCL10, SDA10	
	IIC20	–	SCL20, SDA20
	IICA	SCL0, SDA0	
A/D Convertor		ANI0-ANI7	
Buzzer output/ Clock output		PCLBUZ0	
Key interrupt		KR0-KR3	KR0-KR7
LVI circuit		EXLVI	
On chip debug function		TOOL0, TOOL1	
USB		USBP, USBM, USBPUC	

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +3.8	V
	EV_{DD}		−0.5 to +3.8	V
	V_{SS}		−0.5 to +0.3	V
	EV_{SS}	EV_{SS}= EV_{SS} 1	−0.5 to +0.3	V
	AV_{REF}		−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}	P02 - P04, P10 - P17, P31, P40 - P43, P50 - P53, P70 - P77, P110, P111, P120 - P124, P140, P142 - P144, 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 - P63 (N-ch open-drain)	−0.3 to +6.5	V
	V_{I3}	P20 - P27	−0.3 to AV_{REF} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
	V_{I4}	USBP,USBM	−0.3 to EV_{DD} +0.3 Note 1	
Output voltage	V_{O1}	P02 - P04, P10 - P17, P31, P40 - P43, P50 - P53, P60 - P63, P70 - P77, P110, P111, P120, P130, P140, P142 - P144	−0.3 to EV_{DD} +0.3 Note 1	V
	V_{O2}	P20 - P27	−0.3 to AV_{REF} +0.3	V
	V_{O3}	USBP,USBM,USBPUC	−0.3 to EV_{DD} +0.3 Note 3	V
Analog Input Voltage	V_{AN}	ANI10-ANI17	−0.3 to AV_{REF} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V

Notes 1. Must be 6.5 V or lower.

<R> **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.**

3. Must be 3.8V or lower

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin		−10	mA
		Total of all pins −80 mA	P02 - P04, P40 - P43, P120, P130, P140, P142 - P144	−25	mA
			P10 - P17, P31, P50 - P53, P70 - P77, P110, P111	−55	mA
	I_{OH2}	Per pin	P20 - P27	−0.5	mA
		Total of all pins		−2	mA
<R>	I_{OH3}	Per pin	USBP, USBM	−3	mA
Output current, low	I_{OL1}	Per pin		30	mA
		Total of all pins 200 mA	P02 - P04, P40 - P43, P120, P130, P140, P142 - P144	60	mA
			P10 - P17, P31, P50 - P53, P60 - P63, P70 - P77, P110, P111	140	mA
	I_{OL2}	Per pin	P20 - P27	1	mA
		Total of all pins		5	mA
<R>	I_{OL3}	Per pin	USBP, USBM	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

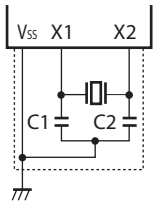
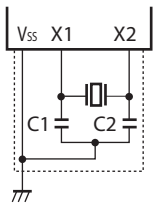
Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure 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.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (fx)Note	3.6 V ≤ V_{DD} ≤ 3.6 V	2.0		20.0	MHz
			1.8 V ≤ V_{DD} ≤ 3.6 V	2.0		5.0	MHz
Crystal resonator		X1 clock oscillation frequency (fx)Note	3.6 V ≤ V_{DD} ≤ 3.6 V	2.0		20.0	MHz
			1.8 V ≤ V_{DD} ≤ 3.6 V	2.0		5.0	MHz

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.

3. Please use an accurate crystal or ceramic oscillator below 500 ppm, when using the USB controller. In case of a USB clock generated from a resonator/oscillator with error above 500ppm, data transmission reception may not happen according to USB standard.

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.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

Internal Oscillator Characteristics

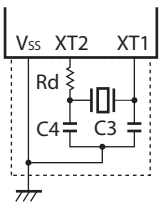
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
	Internal high-speed oscillation clock frequency Note	f_{IH1M}	Consumption current mode	0.87	1.0	1.13	MHz
<R>		f_{IH8M}		7.6	8. 0	8.4	MHz
<R>		f_{IH20M}	2.7V ≤ V_{DD} ≤ 3.6 V	19.0	20.0	21.0	MHz
	Internal low-speed oscillation clock frequency	f_{IL}		25.5	30	34.5	kHz
			2.7V ≤ V_{DD} ≤ 3.6 V	27	30	33	kHz

Note 1. This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.
2. ±15% during consumption mode

XT1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 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 = 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)
Toyama Murata Manufacturing Co., Ltd..	CSTCE12M0GH5L99-R0	SMD	12.0	(33)	(33)	0	2.0	3.6
	CSTCE16M0VH3L99-R0	SMD	16.0	(15)	(15)	0		
	CSTCE20M0VH3L99-R0	SMD	20.0	(15)	(15)	0		

(2) X1 oscillation: Crystal 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)
KYOCERA KINSEKI Corporation	HC49SFWB	Lead	12.0	10	10	0	1.8	3.6
	HS49SFWB	Lead	16.0	10	10	0		
	HS49SFWB	Lead	20.0	10	10	0		

(3) XT1 oscillation: Crystal resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	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	Ultra-low power consumption oscillation	10	10	0	1.8	3.6
					10	10	0		
	SSP-T7-FL				4	4	0		
	VT-200-FL	Lead			4	4	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.

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/KC3-L, 78K0R/KE3-L so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (1/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, 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 P02 - P04, P10 - P17, P31, P40 - P43, P50 - P53, P110, P111, P120, P130, P140, P142 - P144	2.7 V ≤ V_{DD} ≤ 3.6 V			–3.0	mA
			1.8 V ≤ V_{DD} <3.6 V			–1.0	mA
		Total of P02 - P04, P40 - P43, P120, P130, P140, P142 - P144 (When duty = 70%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			–20.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			–5.0	mA
		Total of P10 - P17, P31, P50 - P53, P70 - P77, P110, P111,USBPUC (When duty = 70%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			–30.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			–10.0	mA
		Total of all pins (When duty = 60%[Note 2])	2.7V ≤ V_{DD} ≤ 3.6 V			–29.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			–15.0	mA
	I_{OH2}	Per pin for P20 - P27	AV_{REF} =V_{DD}			–0.1	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from EV_{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 P02 - P04, P10, P12, P40, P142 - P144 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.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (2/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, 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 P02 - P04, P10 - P17, P31, P40 - P43, P50 - P53, P70 - P77, P110, P111, P120, P130, P140, P142 - P144	2.7 V ≤ V_{DD} ≤ 3.6 V			1.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			0.5	mA
		Per pin for P60 - P63	2.7V ≤ V_{DD} ≤ 3.6 V			3.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			2.0	mA
		Total of P02 - P04, P40 - P43, P120, P130, P140, P142 - P144 (When duty = 70%[Note 2])	2.7V ≤ V_{DD} ≤ 3.6 V			15.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			9.0	mA
		Total of P10 - P17, P31, P50 - P53, P60 - P63,P70 - P77, P110, P111 (When duty = 70%[Note 2])	2.7V ≤ V_{DD} ≤ 3.6 V			35.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			20.0	mA
		Total of all pins (When duty = 60%[Note 2])	2.7V ≤ V_{DD} ≤ 3.6 V			50.0	mA
			1.8 V ≤ V_{DD} < 3.6 V			29.0	mA
	I_{OL2}	Per pin for P20 - P27	AV_{REF} = V_{DD}			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 EV_{SS}, 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.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (3/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P02, P12, P13, P15, P41, P52, P111, P123, P124, P144		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P03 ,P04, P10, P11, P14, P16, P17, P31, P40, P42, P43, P50, P51, P53, P70 - P77, P110- P122, P140, P142, P143, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH3}	P03, P04, P10, P11, P142, P143	TTL input buffer 2.7V ≤ V_{DD} ≤ 3.6 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 - P27	AV_{REF} = V_{DD}	0.7AV_{REF}		AV_{REF}	V
	V_{IH5}	P60 - P63	Normal input buffer	0.7V_{DD}		6.0	V
	V_{IH6}	FLMD0		0.9V_{DD} Note1		V_{DD}	V
	V_{IH7}	USBP,USBM	V_{DD} =3.3V ± 10%	2.0		V_{DD}	
Input voltage, low	V_{IL1}	P02, P12, P13, P15, P41, P45, P52, P111, P123, P124, P144		0		0.3V_{DD}	V
	V_{IL2}	P03 ,P04, P10, P11, P14, P16, P17, P31, P40, P42, P43, P50, P51, P53, P70 - P77, P110, P120-P122, P140, P142, P143, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	0		0.2V_{DD}	V
	V_{IL3}	P03, P04, P10, P11, P142, P143	TTL input buffer 2.7 V ≤ V_{DD} < 3.6 V	0		0.5	V
			TTL input buffer V_{DD} < 2.7 V	0		0.2	V
	V_{IL4}	P20 - P27	AV_{REF} = V_{DD} < 3.6 V	0		0.3AV_{REF}	V
	V_{IL5}	P60 - P63		0		0.3V_{DD}	V
	V_{IL6}	FLMD0 Note 2		0		0.1V_{DD}	V
	V_{IL7}	USBP,USBPM		0		0.8	

Note 1. Must be 0.9V_{DD} or higher when used in the flash memory programming mode.

2. When disabling writing of the flash memory, connect the FLMD0 pin processing directly to V_{SS}, and maintain a voltage less than 0.1V_{DD}.

Cautions 1. The maximum value of V_{IH} of pins P02 - P04, P10, P12, P40, and P142 - P144 is V_{DD}, even in the N-ch open-drain mode.

2. For P122/EXCLK, the value of V_{IH} and V_{IL} differs according to the input port mode or external clock mode.Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (4/9)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P02 - P04, P10 - P17, P31, P40 - P43, P50 - P53,P70 - P77, P110, P111, P120, P130, P140, P142 - P144	IOH = 1mA	VDD – 0.5			V
				VDD – 0.5			V
	V_{OH2}	P20 - P27	AVREF = VDD, IOH = –100 μA	AVREF – 0.5			V
	V_{OH3}	USBP,USBM	VDD = 3.3 V ± 10 %, RLPD =15 kΩ VSS connected	2.8		3.6	V
	V_{OH4}	USBPUC	VDD = 3.3 V ± 10 %, IOH = –100 μA	VDD – 0.5			V
Output voltage, low	V_{OL1}	P02-P04, P10-P17, P31, P40-P43, P50-P53, P70-P77, P110, P111, P120, P130 , P140, P142-P144	2.7 V ≤ VDD ≤ 3.6 V, IOL1 = 1.0 mA			0.5	V
			1.8 V ≤ VDD ≤ 3.6 V, IOL1 = 0.5 mA			0.4	V
	V_{OL2}	P20-P27	AVREF = VDD, IOL2 = 0.4 mA			0.4	V
	V_{OL3}	P60-P63	2.7 V ≤ VDD ≤ 3.6 V, IOL1 = 3.0 mA			0.4	V
			1.8 V ≤ VDD ≤ 3.6 V, IOL1 = 2.0 mA			0.4	V
	V_{OL4}	USBP, USBM	VDD = 3.3 V ± 10 %, RLPU =1.5 kΩ VDD connected			0.3	V
	V_{OL5}	USBPUC	VDD = 3.3 V ± 10 %, IOL1 = 1 mA			0.4	V

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (5/9)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P02-P04, P10-P17, P31, P40-P43, P50-P53, P60-P63, P70-P77, P110, P111, P120, P140, P142-P144, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P20-P27	$V_I = AV_{REF}$				1	μA
	I_{LIH3}	P121-P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	Input port			1	μA
				In resonator connection			10	μA
	I_{LIH4}	USBP, USBM	$V_I = V_{DD}$				10	μA
Input leakage current, low	I_{LIL1}	P02-P04, P10-P17, P31, P40-P43, P50-P53, P60-P63, P70-P77, P110, P111, P120, P140, P142-P144, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{SS}$				− 1	μA
	I_{LIL2}	P20-P27	$AV_{REF} = V_{DD}$ $V_I = V_{SS}$				− 1	μA
	I_{LIL3}	P121-P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	Input port			− 1	μA
				In resonator connection			− 10	μA
	I_{LIL4}	USBP, USBM	$V_I = V_{SS}$				− 10	μA

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

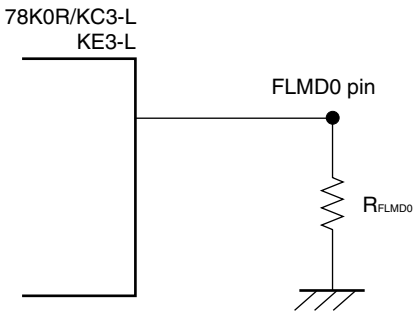
Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (6/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
On-chip pull-up resistance	R_U	P02-P04, P10-P17, P31, P40-P43, P50-P53, P70-P77, P110, P111, P120, P140, P142-P144	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.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (7/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
<R>	Supply current	I_{DD1} Note 1	Operating mode	f_{MX} = 20 MHz Note 2, V_{DD} = 3.3 V	Square wave input		6.1	8.3	mA
					Resonator connection		6.4	8.6	mA
				f_{MX} = 16 MHz Note 2, V_{DD} = 3.3 V	Square wave input		5.1	7.0	mA
					Resonator connection		5.3	7.4	mA
				f_{MX} = 12 MHz Note 2, 3, V_{DD} = 3.3 V	Square wave input		4.1	5.6	mA
					Resonator connection		4.3	5.9	mA
				f_{MX} = 10 MHz Note 2, 3, V_{DD} = 3.3 V	Square wave input		3.4	4.8	mA
					Resonator connection		3.5	4.9	mA
				f_{MX} = 5 MHz Note 2, 3, V_{DD} = 3.3 V	Square wave input		1.9	2.7	mA
					Resonator connection		2.0	2.8	mA
				f_{MX} = 5 MHz Note 2, 3, V_{DD} = 2.0 V	Square wave input		1.3	2.2	mA
					Resonator connection		1.4	2.2	mA
				f_{IH20} = 20 MHz Note 4	V_{DD} = 3.3 V		6.3	8.6	mA
				f_{IH} = 8 MHz Note 4	V_{DD} = 3.3 V		2.7	3.8	mA
				f_{IH} = 1 MHz Note 4, 5	V_{DD} = 3.3 V		210	389	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = –40 to +50°C	V_{DD} = 3.3 V		4.3	9.3	μA
					V_{DD} = 2.0 V		4.3	9.3	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = –40 to +70°C	V_{DD} = 3.3 V		4.3	12.3	μA
					V_{DD} = 2.0 V		4.3	12.3	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = –40 to +85°C	V_{DD} = 3.3 V		4.3	15.5	μA
					V_{DD} = 2.0 V		4.3	15.5	μA

Notes 1. Total current flowing into V_{DD}, EV_{DD}, and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillator, 20 MHz internal high-speed oscillator, subsystem clock, and USB are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0, FLPC and FSEL (bits 1 and 0 of operation speed mode control register (OSMC)) = 0 and 0.

4. When high-speed system clock, subsystem clock, and USB are stopped.

5. When consumption current mode is low (RMC = 5AH, OSMC = 02H)

6. When internal high-speed oscillation, 20 MHz internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer is stopped.

Remark 1. f_{MX} : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

f_{IH20}: 20 MHz internal high-speed oscillation clock frequency

f_{IH} : Internal high-speed oscillation clock frequency

f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

2. RMC: Regulator mode control register

3. Temperature conditions of TYP value are T_A = 25 °C

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (8/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
<R>	Supply current	I_{DD2} Note 1	HALT mode	f_{MX} = 20 MHz Note 2, V_{DD} = 3.3 V	Square wave input		1.2	3.6	mA
					Resonator connection		1.5	3.9	mA
				f_{MX} = 16 MHz Note 2, V_{DD} = 3.3 V	Square wave input		1.0	3.0	mA
					Resonator connection		1.3	3.3	mA
				f_{MX} = 12 MHz Note 2, V_{DD} = 3.3 V	Square wave input		0.8	2.4	mA
					Resonator connection		1.1	2.7	mA
				f_{MX} = 10 MHz Note 2, 3, V_{DD} = 3.3 V	Square wave input		0.55	2.1	mA
					Resonator connection		0.65	2.2	mA
				f_{MX} = 5 MHz Note 2, 3, V_{DD} = 3.3 V	Square wave input		0.40	1.8	mA
					Resonator connection		0.45	1.8	mA
				f_{MX} = 5 MHz Note 2, 3, V_{DD} = 2.0 V	Square wave input		0.28	1.3	mA
					Resonator connection		0.33	1.4	mA
				f_{IH20} = 20 MHz Note 4	V_{DD} = 3.3 V		1.4	3.9	mA
				f_{IH} = 8 MHz Note 4	V_{DD} = 3.3 V		0.48	1.8	mA
				f_{IH} = 1 MHz Note 4, 5	V_{DD} = 3.3 V		55	168	μA
				f_{SUB} = 32.768 kHz Note 6, T_A = –40 to +50°C	V_{DD} = 3.3 V		1.0	3.7	μA
					V_{DD} = 2.0 V		1.0	3.7	μA
				f_{SUB} = 32.768 kHz Note 6, T_A = –40 to +70°C	V_{DD} = 3.3 V		1.0	6.1	μA
					V_{DD} = 2.0 V		1.0	6.1	μA
				f_{SUB} = 32.768 kHz Note 6, T_A = –40 to +85°C	V_{DD} = 3.3 V		1.0	8.9	μA
					V_{DD} = 2.0 V		1.0	8.9	μA
		I_{DD3} Note 7	STOP mode	T_A = –40 to +50°C			0.37	2.8	μA
				T_A = –40 to +70°C			0.37	5.2	μA
				T_A = –40 to +85°C			0.37	7.9	μA

Notes 1. Total current flowing into V_{DD}, EV_{DD}, and AV_{REF}, 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 A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.

2. When internal high-speed oscillator, 20 MHz internal high-speed oscillator, subsystem clock, and USB are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0, FLPC and FSEL (bits 1 and 0 of operation speed mode control register (OSMC)) = 0 and 0.

4. When high-speed system clock, subsystem clock, and USB are stopped.

5. When consumption current mode is low (RMC = 5AH, OSMC = 02H)

6. When internal high-speed oscillation, 20 MHz internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer and USB are stopped. When the real-time counter is operated

7. Total current flowing into V_{DD}, EV_{DD}, and AV_{REF}, 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 A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. When subsystem clock, watchdog timer, and USB are stopped.

Remark 1. f_{MX} **: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)**

f_{IH20}**: 20 MHz internal high-speed oscillation clock frequency**

f_{IH} **: Internal high-speed oscillation clock frequency**

f_{SUB}**: Subsystem clock frequency (XT1 clock oscillation frequency)**

2. RMC: Regulator mode control register

3. Temperature conditions of TYP value are T_A **= 25°C**

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

DC Characteristics (9/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
	RTC operating current	I_{RTC} Notes 1, 2	f_{SUB} = 32.768 kHz		V_{DD} = 3.3 V		0.2	1.0	μA
					V_{DD} = 2.0 V		0.2	1.0	
	Watchdog timer operating current	I_{WDT} Notes 2, 3	f_{IL} = 30 kHz				0.31	0.35	μA
	A/D converter operating current	I_{ADC} Note 4	During conversion at maximum speed	High speed mode 2	AV_{REF} = V_{DD} = 3.0 V		0.72	1.6	mA
				Normal mode	AV_{REF} = V_{DD} = 3.0 V		0.86	1.9	mA
				Low voltage mode	AV_{REF} = V_{DD} = 3.0 V		0.37	0.8	mA
	LVI operating current	I_{LVI} Note 5					9	18	μA
<R>	USB operating current	I_{USB} Note 6	f_{USB} = 48 MHz		V_{DD} = 3.3 V		14.8	21.0	mA

Notes 1. Current flowing only to the real-time counter (excluding the operating current of the XT1 oscillator). The TYP. value of the current value of the 78K0R/KC3-L, KE3-L is the sum of the TYP. values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time counter operates in 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_{SUB}/2, the TYP. value of I_{DD2} includes the real-time counter operating current.

2. When internal high-speed oscillator and high-speed system clock are stopped.

3. Current flowing only to the watchdog timer (including the operating current of the 30 kHz internal oscillator). The current value of the 78K0R/KC3-L, KE3-L is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates during f_{CLK} = f_{SUB}/2 or STOP mode.

4. Current flowing only to the A/D converter (AV_{REF} pin). The current value of the 78K0R/KC3-L, KE3-L is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

5. Current flowing only to the LVI circuit. The current value of the 78K0R/KC3-L, KE3-L is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates in the Operating, HALT or STOP mode.

<R> **6. Current including the PLL operating current of USB clock.**

Remark 1. fil :Low speed internal occiliasion clock frequency
f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)
fclk: CPU/ Peripheral hardware clock frequency

2. Temperature conditions of TYP value are T_A = 25 °C

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

AC Characteristics

(1) Basic operation (1/6)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

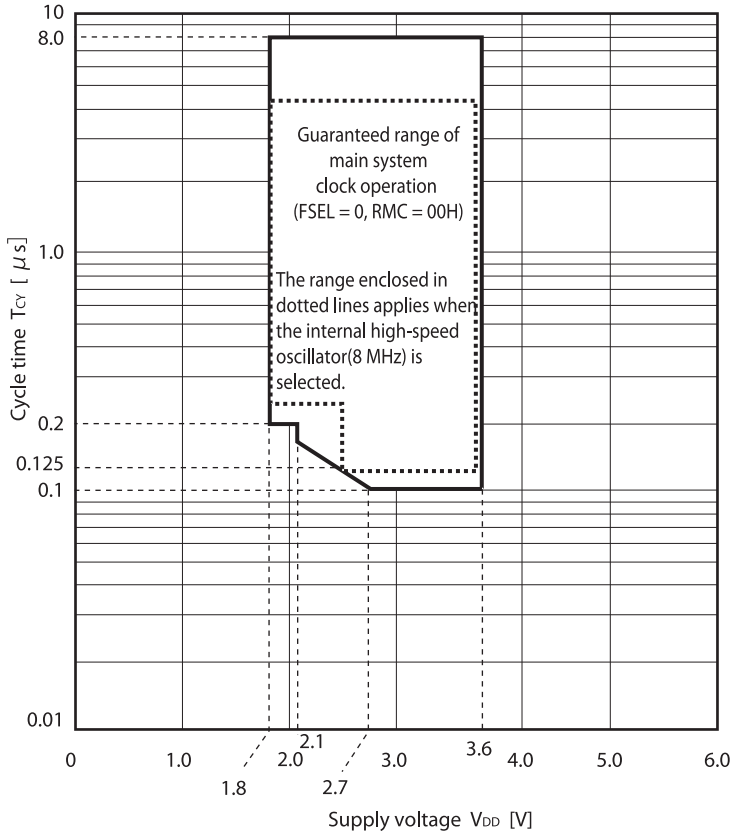
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	Normal current mode	2.7 V ≤ V_{DD}	0.05		8	μs
				V_{DD} < 2.7 V	0.2		8	μs
			Low consumption current mode		1		8	μs
		Subsystem clock (f_{SUB}) operation		SDIV=1	57.2	61	62.5	μs
		In the self programming mode	Normal current mode	2.7 V ≤ V_{DD}	0.05		0.5	μs
				V_{DD} < 2.7 V	0.2		0.5	μs
			Low consumption current mode			1		μs
External main system clock frequency	f_{EX}	EXCLK		2.7 V ≤ V_{DD}	2.0		20.0	MHz
				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}	EXCLK		2.7 V ≤ V_{DD}	24			ns
				V_{DD} < 2.7 V	96			ns
TI00-TI04 TI13 input high-level width, low-level width	t_{TIH}, t_{TIL}	TI00-TI04			2/f_{MCK}+ 10			ns
TO00-TO04 output frequency	f_{TO}	TO00-TO04		2.7 V ≤ V_{DD}			10	MHz
				V_{DD} < 2.7 V			5	MHz
PCLBUZ0 output frequency	f_{PCL}			2.7 V ≤ V_{DD}			10	MHz
				V_{DD} < 2.7 V			5	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP00-INTP02, INTP04-INTP06, INTP08-INTP11			1			μs
Key Interrupt input high-level width, low-level width	t_{KR}	KR0-KR7			250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}	$\overline{\text{RESET}}$			10			μs

Remarks 1. f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the TMR0n register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7))

2. For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to CHAPTER 22 REGULATOR.

(1) Basic operation (2/6)

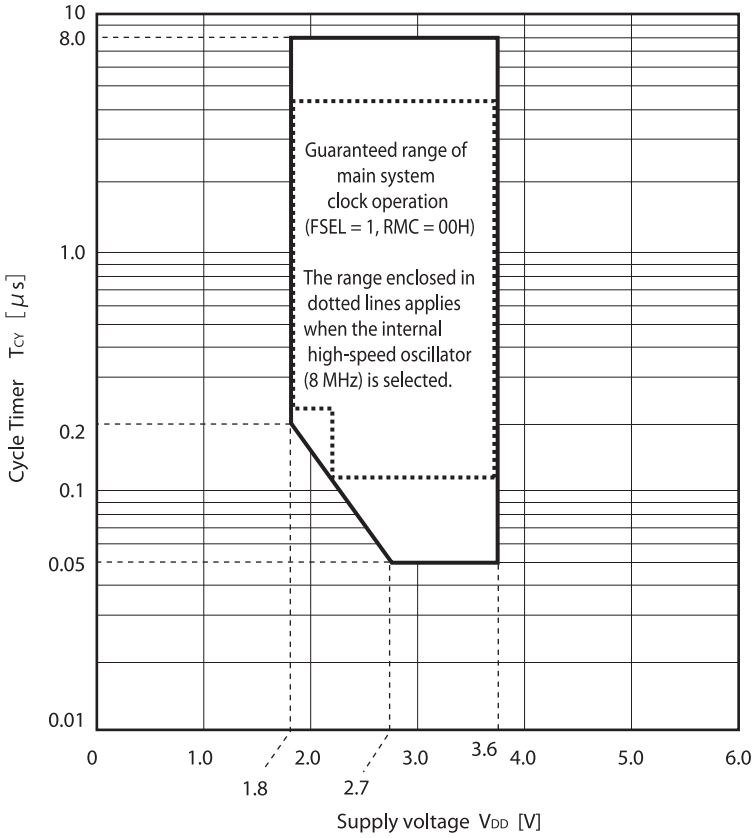
Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)



Remark FSEL: Bit 0 of the operation speed mode control register (OSMC)
RMC: Regulator mode control register

(1) Basic operation (3/6)

Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)

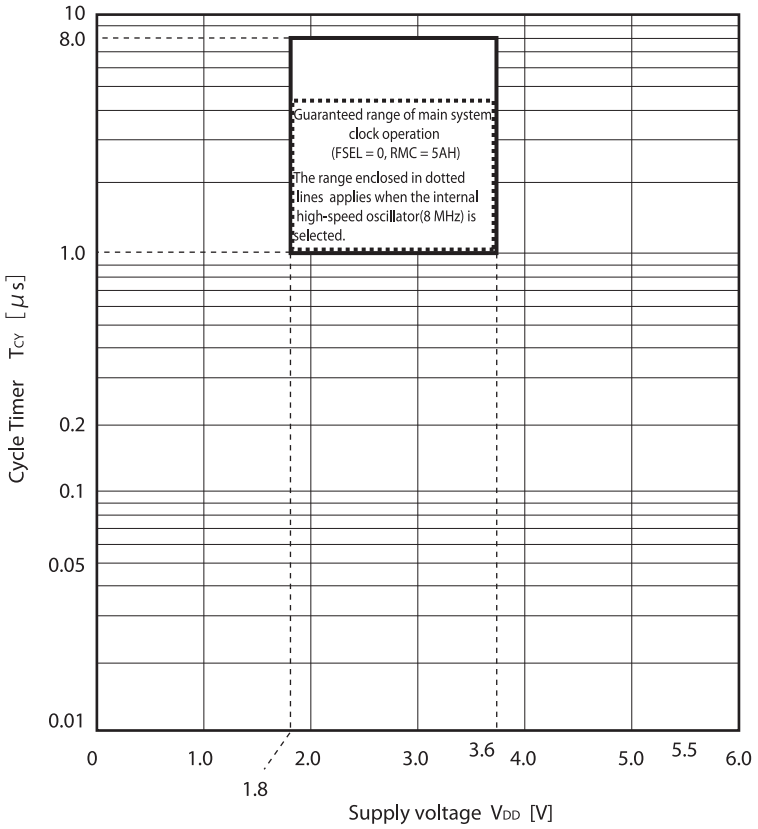


Caution Set FSEL = 0 to shift to STOP mode while $V_{DD} \leq 2.7$ V.

Remark FSEL : Bit 0 of the operation speed mode control register (OSMC)
RMC : Regulator mode control register
f_{IH} : High speed internal oscillation clock frequency
f_{EX} : External main system clock frequency.

(1) Basic operation (4/6)

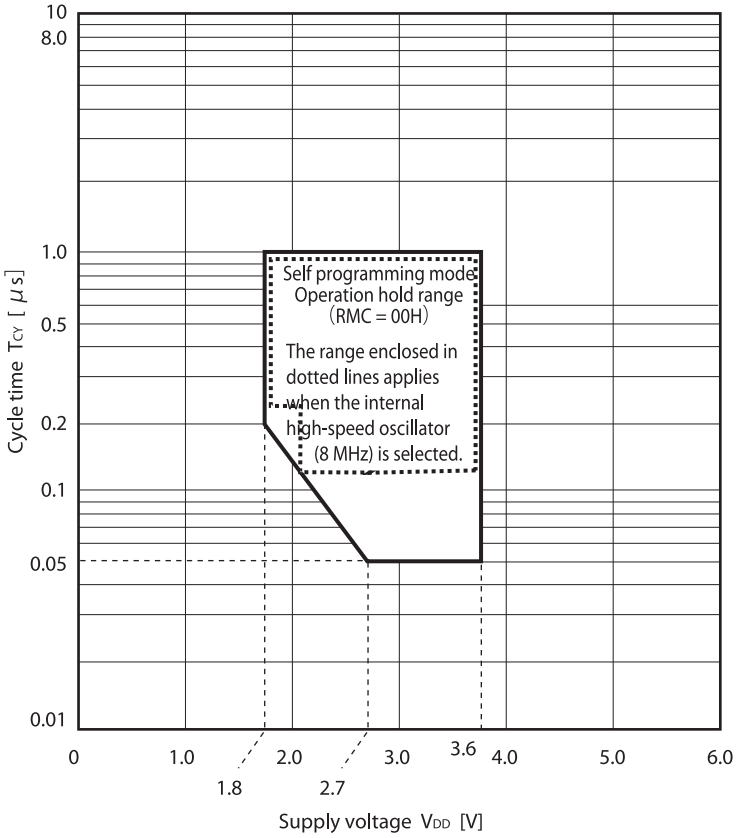
Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)



Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)
RMC: Regulator mode control register
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)



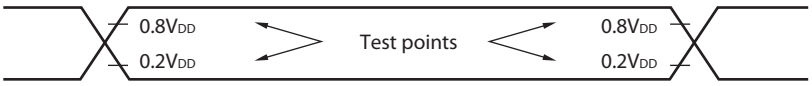
Remarks.1 RMC: Regulator mode control register

2 The self programming function cannot be used when CPU operates with the subsystem clock.

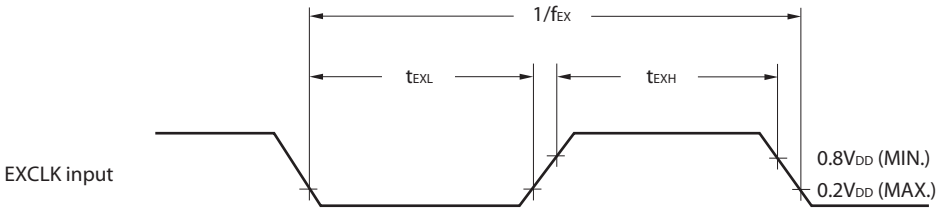
3 When RMC is set to 5AH, entire voltage range is set to 1MHz

(1) Basic operation (6/6)

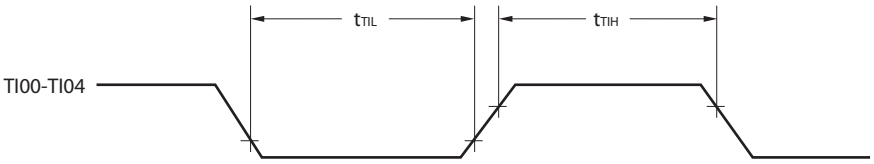
AC Timing Test Points (Excluding external bus interface)



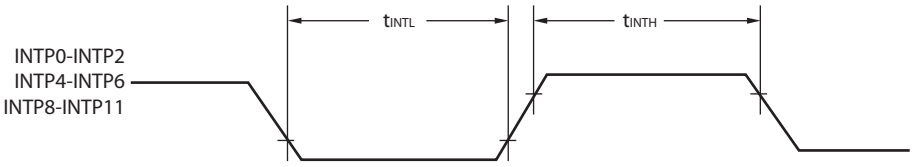
External Main System Clock Timing



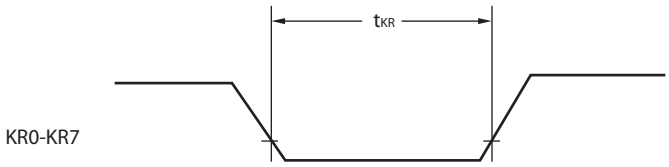
TI Timing



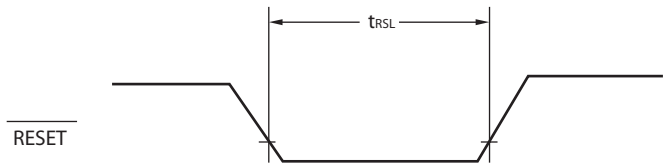
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

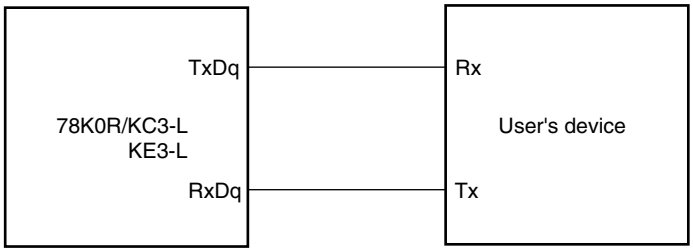
(2) Serial interface: Serial array unit (1/17)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

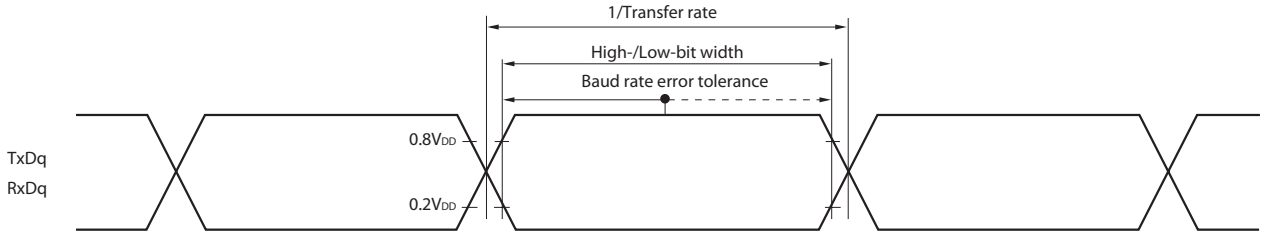
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

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 POMg registers.

Remarks 1. q: UART number (q = 0 - 3), g: PIM and POM number (g = 0, 1, 14)

2. f_{MCK}: Serial array unit operation clock frequency

(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 0 to 4))

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (2/17)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(b) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V	400			ns
		V_{DD} = EV_{DD} < 2.7 V	800			ns
$\overline{SCKp}$ high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V	t_{KCY1}/2 – 35			ns
		V_{DD} = EV_{DD} < 2.7 V	t_{KCY1}/2 – 80			ns
SIp setup time(to$\overline{SCKp}$↑)[Note 1]	t_{SIK1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V	100			ns
		V_{DD} = EV_{DD} < 2.7 V	190			ns
SIp SIp hold time (to $\overline{SCKp}$↑)[Note 1]	t_{KSI1}		30			ns
Delay time from $\overline{SCKp}$↓ to SOp output [Note 2]	t_{KSO1}	C = 50 pF[Note 3]			40	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. 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 port input mode register (PIMg) and port output mode register (POMg) registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM and POM number (g = 0, 1, 14)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 ,2)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (3/17)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(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}	f_{MCK} >16 MHz		8/f_{MCK}			ns
		f_{MCK} ≤ 16 MHz		6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}			f_{KCY2}/2			ns
SIp setup time (to $\overline{SCKp}$↑) Note 1	t_{SIK2}			1/f_{MCK}+ 80			ns
SIp hold time (from $\overline{SCKp}$↑) Note 1	t_{KSI2}			50			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 2	t_{KSO2}	C = 50 pF Note 3	2.7 V ≤ V_{DD} = EV_{DD} 3.6 V			1/f_{MCK}+120	ns
			V_{DD} = EV_{DD} < 2.7V			1/f_{MCK}+120	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 delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. 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 port input mode register (PIMg) and port output mode register (POMg) registers.

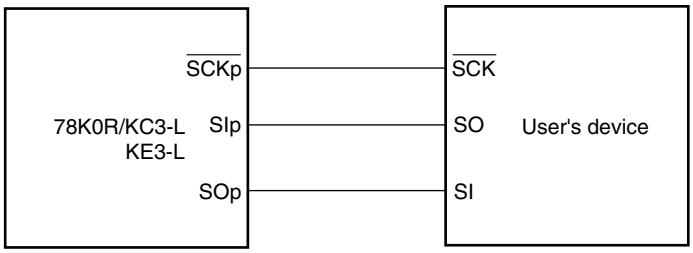
Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM and POM number (g = 0, 1, 14)

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)

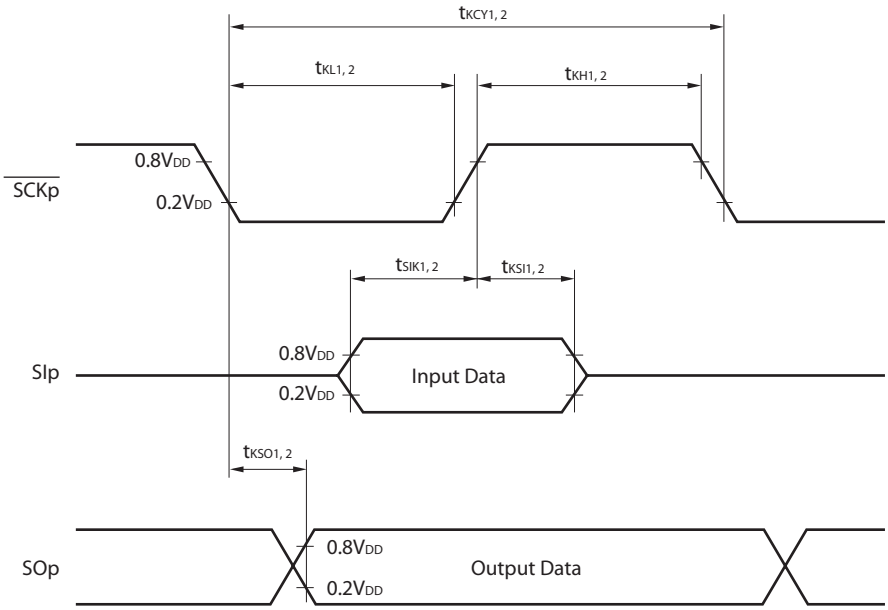
Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (4/17)

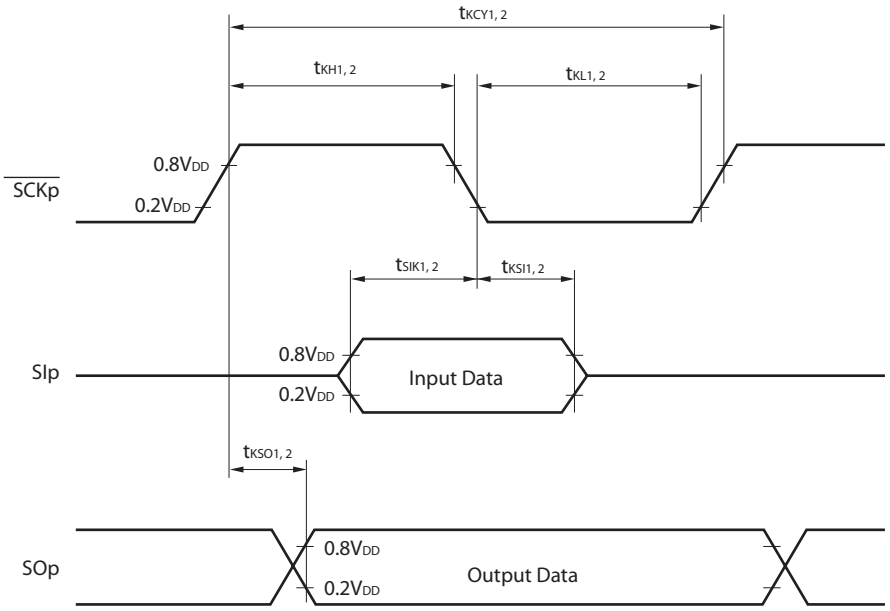
CSI mode connection diagram (during communication at same potential)



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)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (5/17)

(T_A = –40 to +85°C, 3.6 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(d) During communication at same potential (simplified I^2C mode)

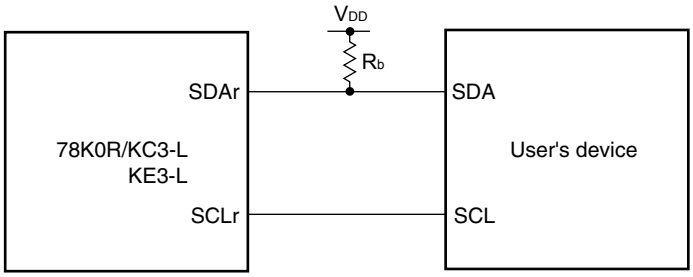
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ		400	kHz
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 5 kΩ		300	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	995		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 5 kΩ	1500		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	995		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 5 kΩ	1500		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 120		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 5 kΩ	1/f_{MCK} + 230		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	0	410	ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 5 kΩ	0	460	ns

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

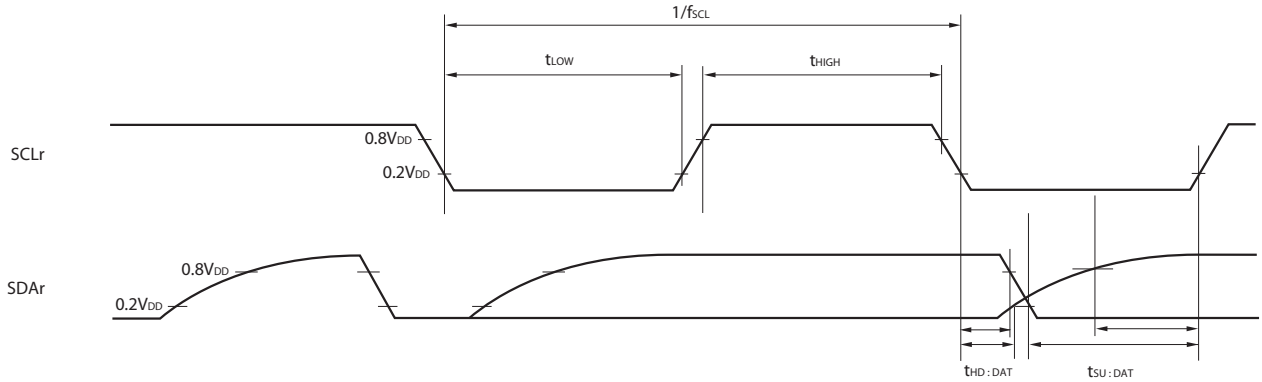
(2) Serial interface: Serial array unit (6/17)

(T_A = –40 to +85°C, 3.6 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Simplified I²C mode mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



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 port input mode register (PIMg) and port output mode register (POMg) 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 and POM number (g = 0, 14)

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)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (7/17)

(T_A = –40 to +85°C, 3.6 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(e) During 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	3.6 V ≤ V_{DD} ≤ 3.6 V, 3.6 V ≤ V_b ≤ 4.0 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 port input mode register (PIMg) and port output mode register (POMg) registers.

.

Remarks 1. q: UART number (q = 0-3), g: PIM and POM number (g = 0, 1, 14)

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 to 3))

3. 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.
2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (8/17)

(T_A = –40 to +85°C, 3.6 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(e) During 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	2.7 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V 2.3 V ≤ V_b ≤ 2.7 V				Note1	
				f_{CLK} = 16.8 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V			1.2 Note 2	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 2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 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.

Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the port input mode register (PIMg) and port output mode register (POMg) registers.

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 = 0-3), and POM number (g = 0, 1, 14)

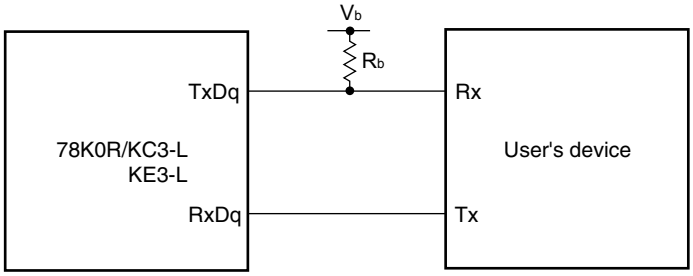
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3))

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.
When 2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V:V_{IH} = 2.0 V, V_{IL} = 0.5 V

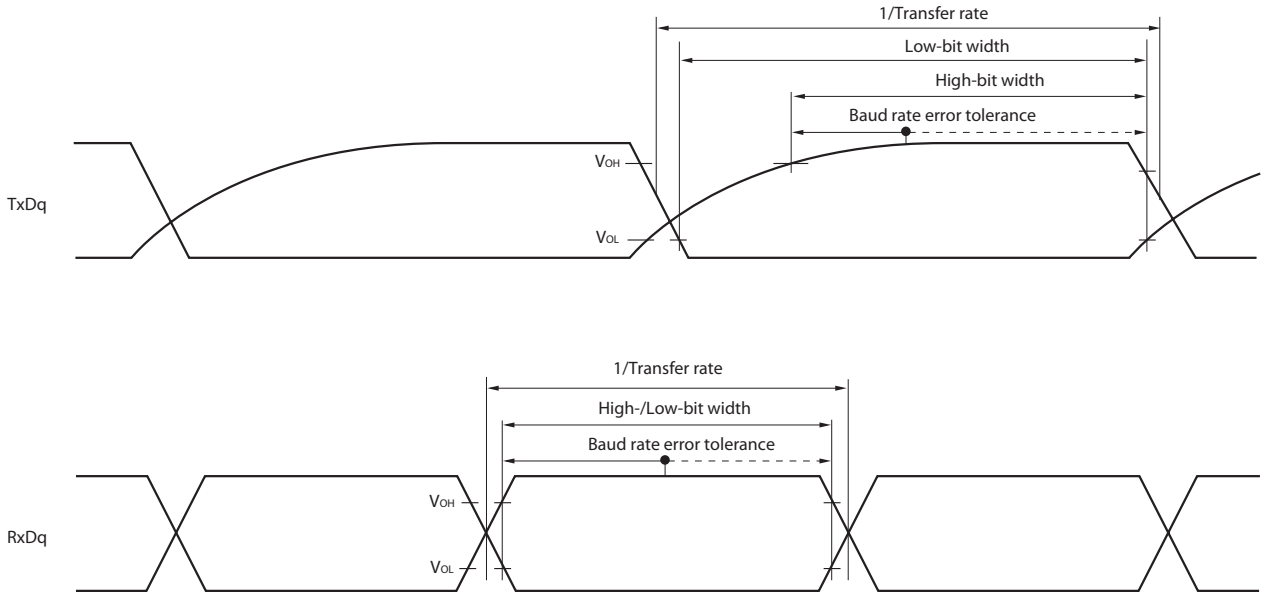
Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (9/17)

UART mode connection diagram (during communication at different potential)



UART mode bit width (during 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 port input mode register (PIMg) and port output mode register (POMg) registers.

Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, V_b[V]: Communication line voltage

2. q: UART number (q = 0 3), g: PIM and POM number (g = 0, 1, 14)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (10/17)

(T_A = −40 to +85 °C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = 0 V)

(f) During 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}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ	1000			ns
$\overline{SCKp}$ high-level width	t_{KH1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 275			ns
$\overline{SCKp}$ low-level width	t_{KL1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ	t_{KCY1}/2 –35			ns
SIp setup time (to $\overline{SCKp}$↑) Note	t_{SIK1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ	380			ns
SIp hold time (from $\overline{SCKp}$↑) Note	t_{KSI1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note	t_{KSO1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ			320	ns

Note When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 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 port input mode register (PIMg) and port output mode register (POMg) registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM and POM number (g = 0, 1, 14)

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 (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.

When 2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V:V_{IH} = 2.0 V, V_{IL} = 0.5 V

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (11/17)

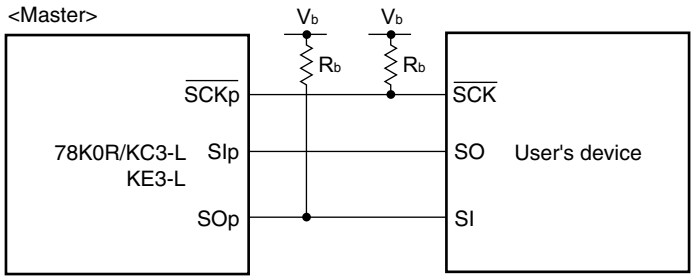
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = 0 V)

(f) During 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 ($to\overline{SCKp}\downarrow$)Note	t_{SIK1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ	100			ns
SIp hold time ($to\overline{SCKp}\downarrow$)Note	t_{KSI1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}\uparrow$ to SOp output Note	t_{KSO1}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 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 (during 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 port input mode register (PIMg) and port output mode register (POMg) registers .

Remarks 1. p: CSI number (p = 01, 10, 20), g: PIM and POM number (g = 0, 1, 14)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0,2)

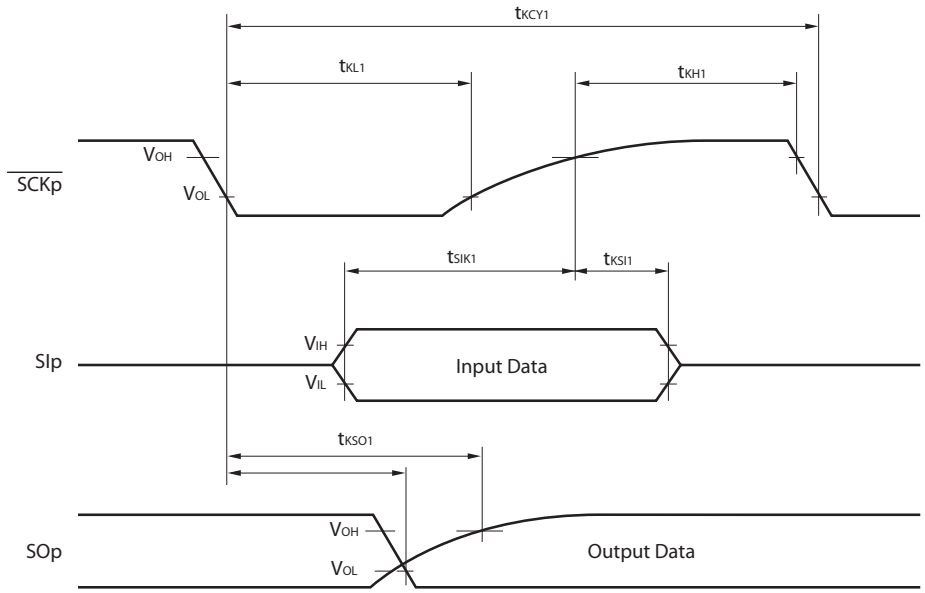
3. R_b[Ω]:Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line (SOp, $\overline{SCKp}$) load capacitance, V_b[V]: Communication line voltage

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.
When 2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V:V_{IH} = 2.0 V, V_{IL} = 0.5 V

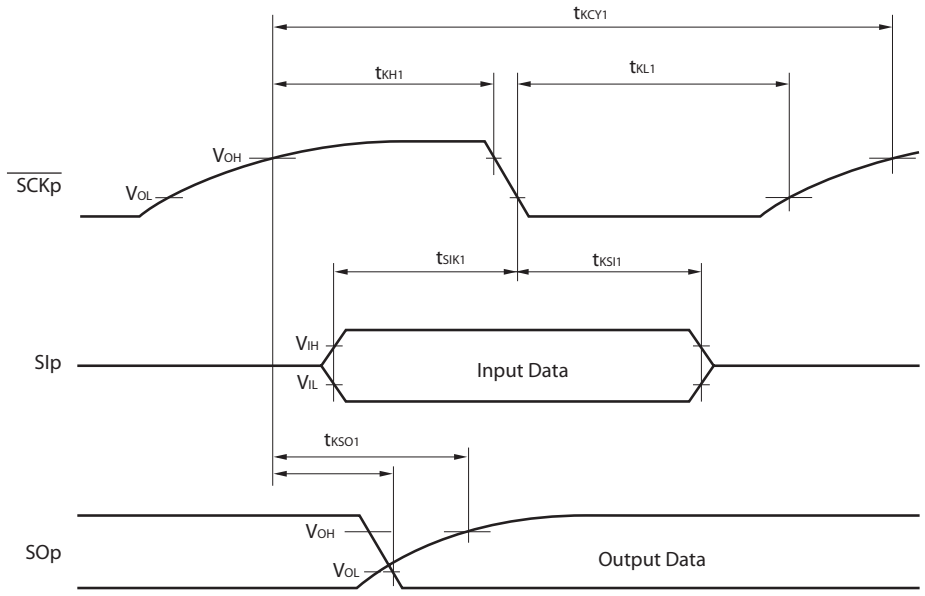
Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (12/17)

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution1. Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ port input mode register (PIMg) and port output mode register (POMg) registers.

2. For V_{IH},V_{IL} refer toTTL buffer specifications. Voltage ofV_{OH},V_{OL} same as V_{IH},V_{IL}.

Remarks 1. p: CSI number (p = 00, 20), g: PIM and POM number (g = 0, 1, 14)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 ,2)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (13/17)

(T_A = −40 to +85 °C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = 0 V)

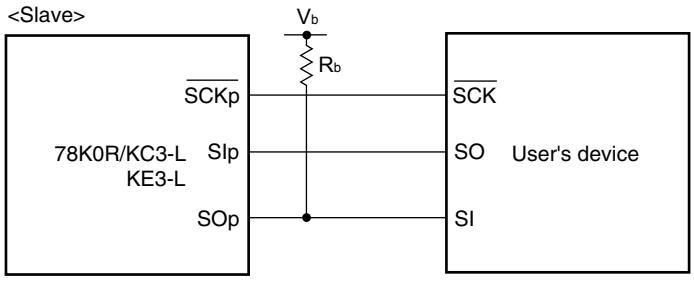
(g) During Communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V	17.5 MHz < f_{MCK}	18/f_{MCK}			ns
			15 MHz < f_{MCK} ≤ 17.5 MHz	16/f_{MCK}			ns
			12.5 MHz < f_{MCK} ≤ 15 MHz	14/f_{MCK}			ns
			10 MHz < f_{MCK} ≤ 12.5 MHz	12/f_{MCK}			ns
			7.5 MHz < f_{MCK} ≤ 10 MHz	10/f_{MCK}			ns
			5 MHz < f_{MCK} ≤ 7.5 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 5 MHz	6/f_{MCK}			ns
$\overline{SCKp}$high,low level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2–35			ns
Sip set up time (to $\overline{SCKp}$↑)[Note 1]	t_{SIK2}			1/f_{MCK} + 90			ns
Sip hold time (to$\overline{SCKp}$↑)[Note 1]	t_{KSI2}			50			ns
Delay time from $\overline{SCKp}$↓ to SOp[Note 2]	t_{KSO2}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 50 pF, R_b = 2.7 kΩ				1/f_{MCK} + 400	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 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 (during communication at different potential)



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 port input mode register (PIMg) and port output mode register (POMg) registers.

(Remark is given on the next page.)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (14/17)

Remarks 1. p: CSI number (p = 01, 10, 20), g: PIM and POM number (g = 0, 1, 14)

2. R_b[Ω]:Communication line (SOp) pull-up resistance,
C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

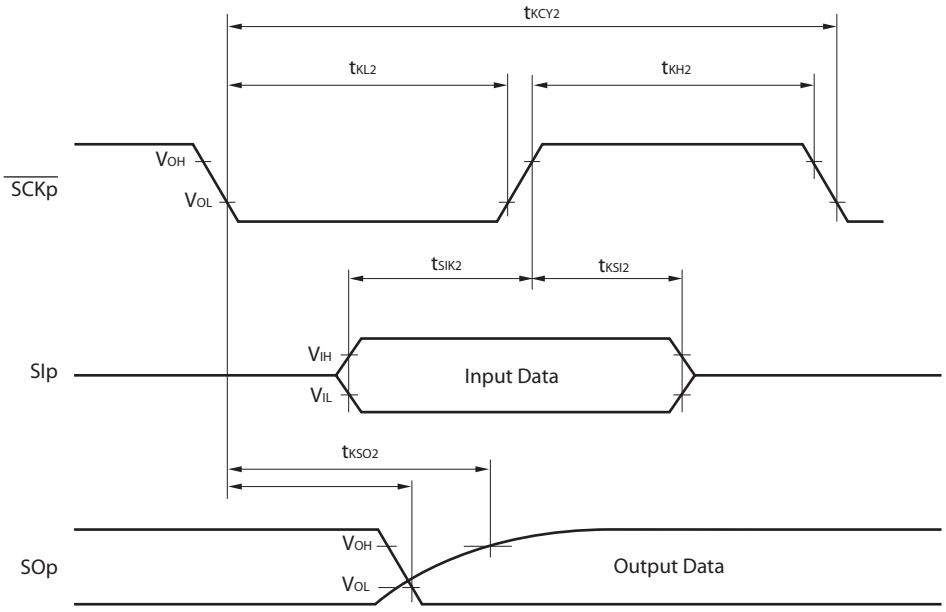
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 ,2))

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.
When 2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V:V_{IH} = 2.0 V, V_{IL} = 0.5 V

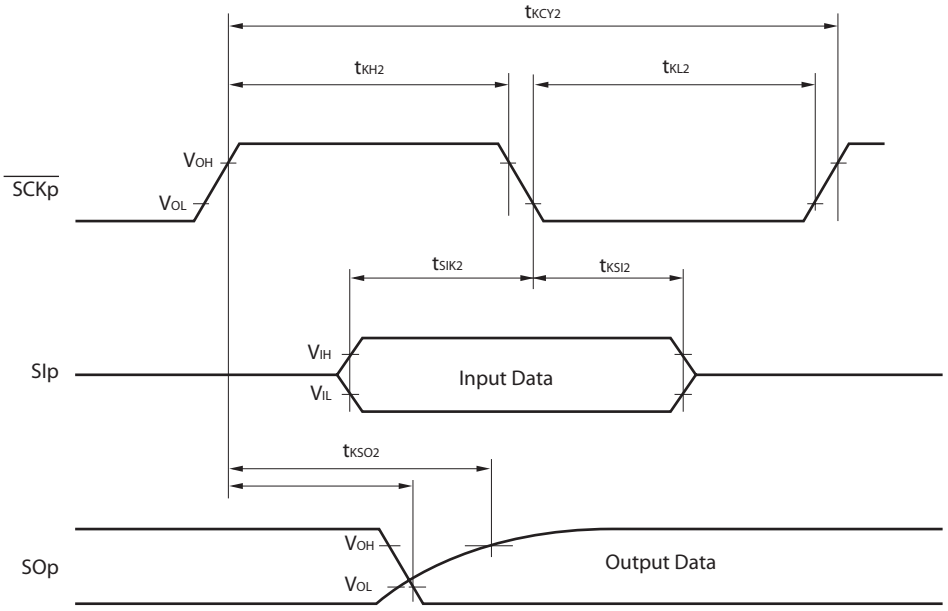
Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (15/17)

CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (slave mode) (during 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 port input mode register (PIMg) and port output mode register (POMg) registers.

Remarks 1. p: CSI number (p = 00, 20), g: PIM and POM number (g = 0, 1, 14)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 ,2)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (16/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(h) During Communication at different potential (2.5 V, 3 V) (simplified I^2C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 100 pF, R_b = 2.7 kΩ		400	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 100 pF, R_b = 2.7 kΩ	1065		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 100 pF, R_b = 2.7 kΩ	445		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} = EV_{DD} < 3.6 V 2.3 V ≤ V_b ≤ 2.7 V C_b = 100 pF, R_b = 2.7 kΩ	0	410	ns

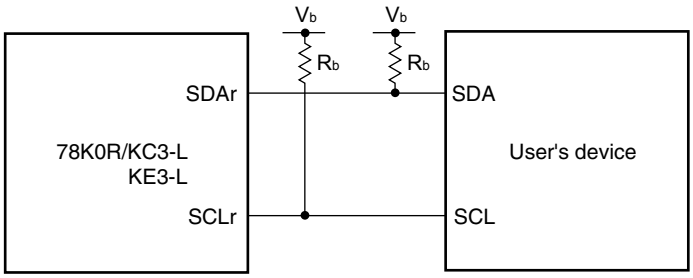
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the N-ch open drain output (V_{DD} tolerance) mode for SCLr by using port input mode register (PIMg) and port output mode register (POMg) registers.

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, POM number (g = 0,1, 14)
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.
When 2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V:V_{IH} = 2.0 V, V_{IL} = 0.5 V

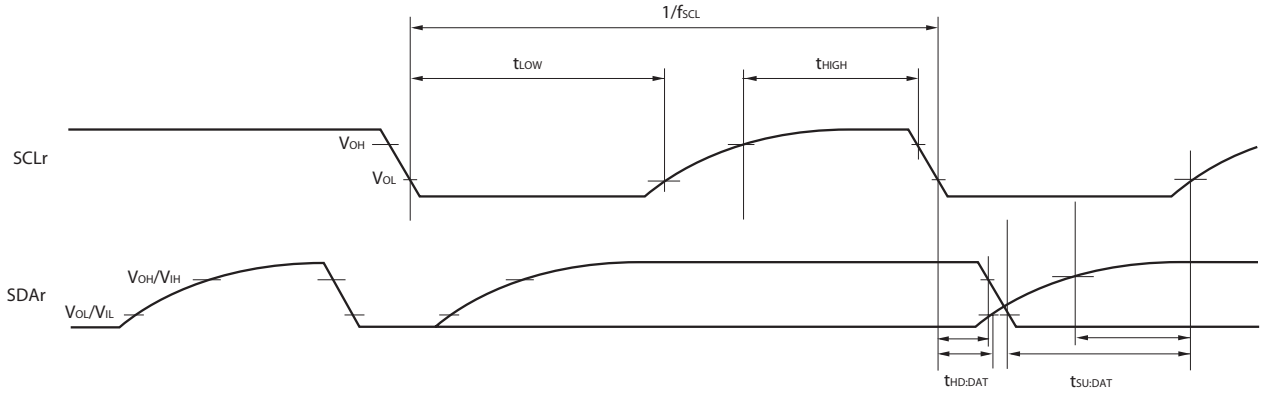
Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(2) Serial interface: Serial array unit (17/17)

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Caution1. 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 port input mode register (PIMg) and port output mode register (POMg) registers.

2. For V_{IH},V_{IL} refer to TTL buffer specifications. Voltage for V_{OH},V_{OL} is same as V_{IH},V_{IL}

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 and POM number (g = 0, 1,14)

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

(3) Serial interface: IICA

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

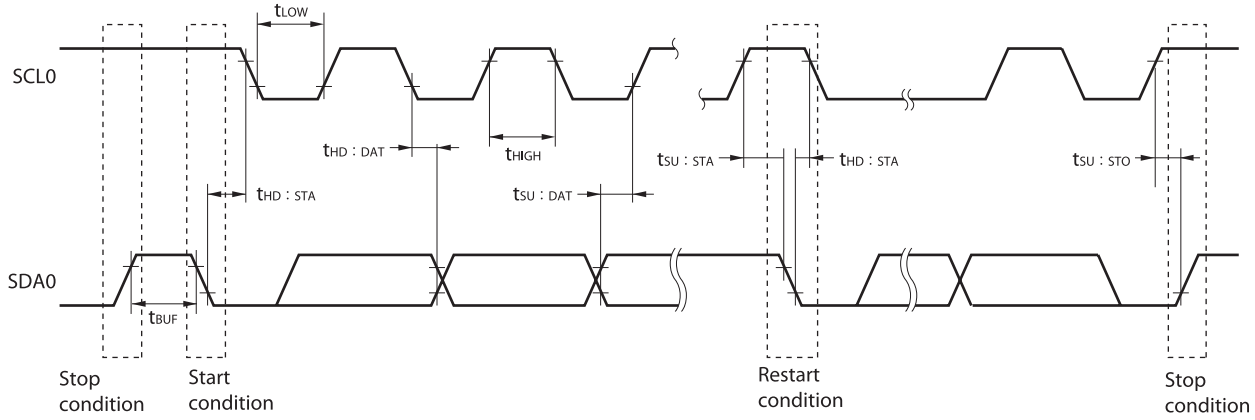
(a) IICA

Parameter	Symbol	Conditions	Standard Mode		High-Speed Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{SCL}	High speed mode:f_{CLK} ≥ 3.5 MHz Normal speed 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.

IICA serial transfer timing



(4) Serial interface: On-chip debug (UART)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) On-chip debug (UART)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			$f_{CLK}/2^{12}$		$f_{CLK}/6$	bps
		Serial programming mode (f_{CLK} = 20 MHz, V_{DD} ≥ 2.7V, C_b = 50 pF)			3.33	Mbps
TOOL1 output frequency	f_{TOOL1}	V_{DD} ≥ 2.7 V			10	MHz
		V_{DD} < 2.7 V			2.5	MHz

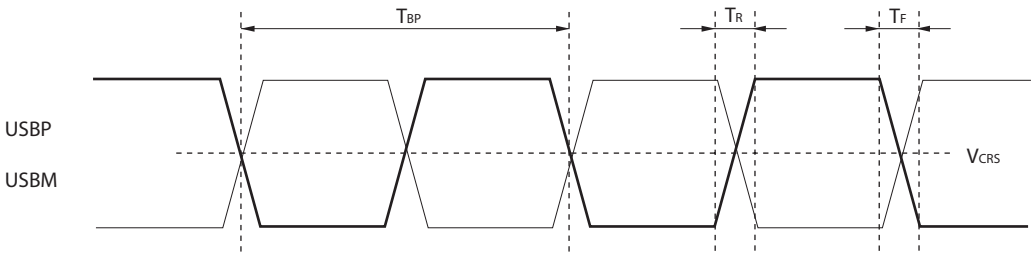
Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

USBF charachteristics(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USBM, USBP pin output rise Rise/Fall time	T_R, T_F	C_L = 50 pF[Note]	4		20	ns
Full speed data speed	T_{DRATE}		11.97	12.00	12.03	Mbps
Bit period	TBP		83.12	83.33	83.54	ns
USBM, USBP pin rise/ fall time matching	T_{RFM}	T_R/T_F	90		110	%
USBM, USBP pin output signal cross point voltage	V_{CRS}		1.3		2.0	V

Note C_L is the output line load capacity of USBM, USBP.

USBF Timing



Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

A/D Converter Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Overall error[Notes 1, 2]	AINL	2.7 V ≤ AV_{REF} ≤ 3.6 V				±0.35	%FSR
Conversion time	tCONV	High speed mode 2 Normal mode	High speed mode	3.4		66.6	μs
			Normal mode	8.6		66.6	μs
		Voltage boost mode	Low voltage mode	24.1		66.6	μs
Zero-scale error[Notes 1, 2]	EZS					±0.25	%FSR
Full-scale error[Notes 1, 2]	EFS					±0.25	%FSR
Integral linearity error[Note 1]	ILE					±2.5	LSB
Differential linearity error[Note 1]	DLE					±1.5	LSB
Analog input voltage	VAIN			AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

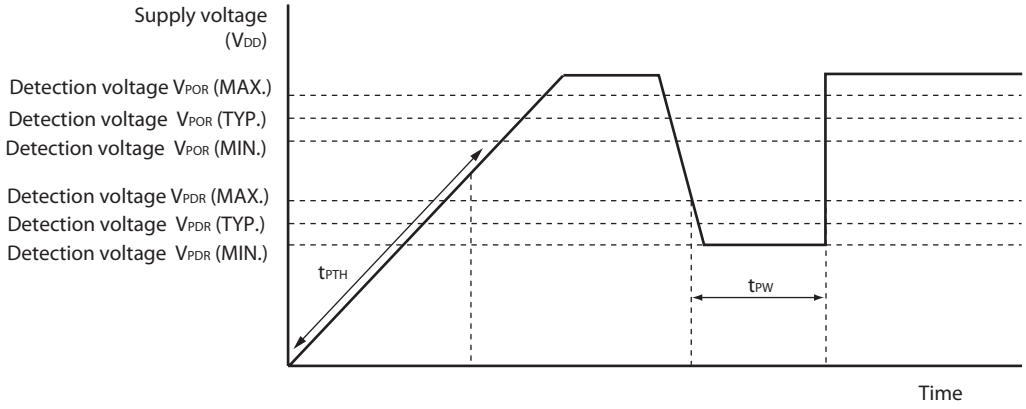
2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

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}		1.50	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



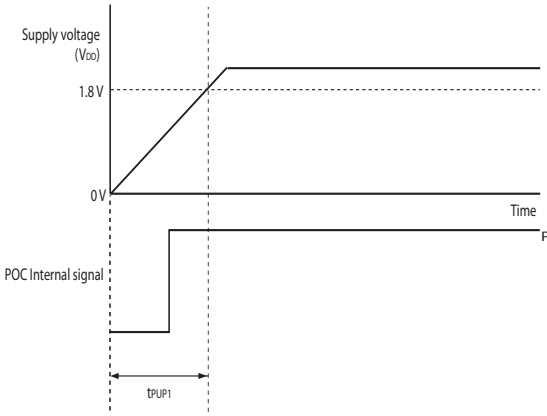
Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

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

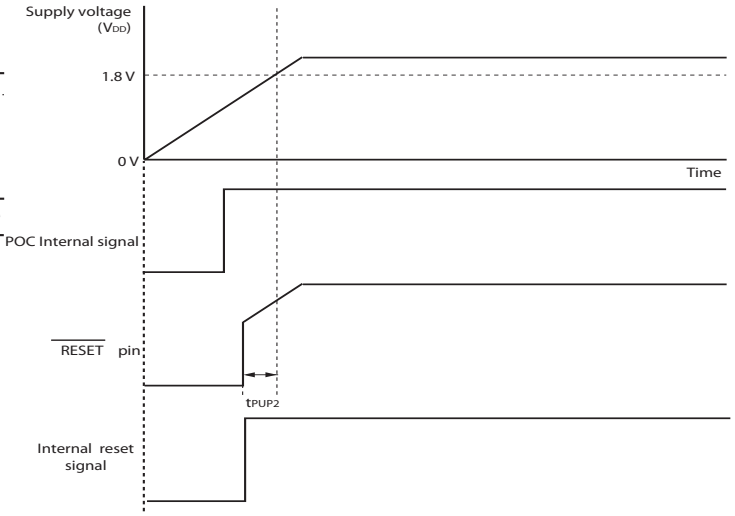
Note Make sure to raise the power supply in a shorter time than this.

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)



Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

LVI Circuit Characteristics (T_A = –40 to +85°C, $V_{PDR} \le V_{DD} = EV_{DD} \le 3.6$ V, $V_{SS} = EV_{SS}$ = 0 V)

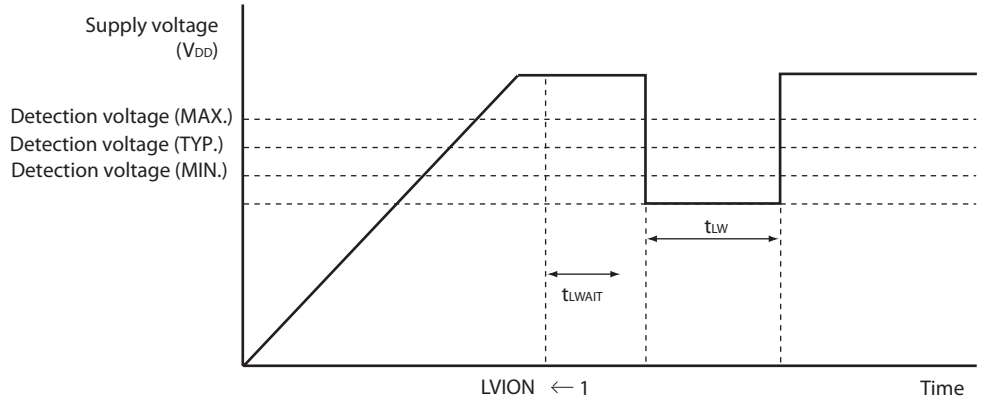
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI5}			3.45 ± 0.1		V
		V_{LVI6}			3.30 ± 0.1		V
		V_{LVI7}			3.15 ± 0.1		V
		V_{LVI8}			2.99 ± 0.1		V
		V_{LVI9}			2.84 ± 0.1		V
		V_{LVI10}			2.68 ± 0.1		V
		V_{LVI11}			2.53 ± 0.1		V
		V_{LVI12}			2.38 ± 0.1		V
		V_{LVI13}			2.22 ± 0.1		V
		V_{LVI14}			2.07 ± 0.1		V
		V_{LVI15}			1.91 ± 0.1		V
	External input pin[Note 1]	V_{EXLVI}	EXLVI < V_{DD}, 1.8 V ≤ V_{DD} ≤ 3.6 V		1.21 ± 0.1		V
	Power supply voltage on power application	V_{PUPLVI}	When LVI default start function enabled is set		2.07 ± 0.2		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 = 6 - 15

LVI Circuit Timing

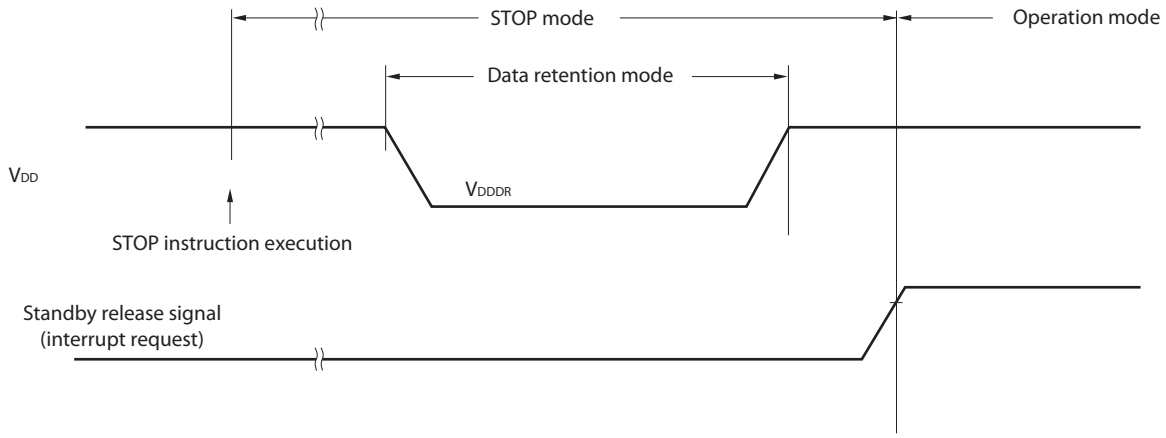


Caution Provided pins may change according to products. Please refer to the table given at the beginning of this chapter in Caution 2.

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		3.6	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained before 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} ≤ 3.6 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
CPU/peripheral hardware clock frequency	f_{CLK}	2.7 V ≤ V_{DD} ≤ 3.6 V		2		20	MHz
		1.8 V ≤ V_{DD} ≤ 3.6 V		1		5	MHz
Number of rewrites (number of deletes per block)	C_{erwr}	Used for updating programs When using flash memory programmer and Renesas Electronics self programming library	Retained for 15 years	1,000			Times
		Used for updating data When using Renesas Electronics EEPROM emulation library	Retained for 5 years	10,000			Times

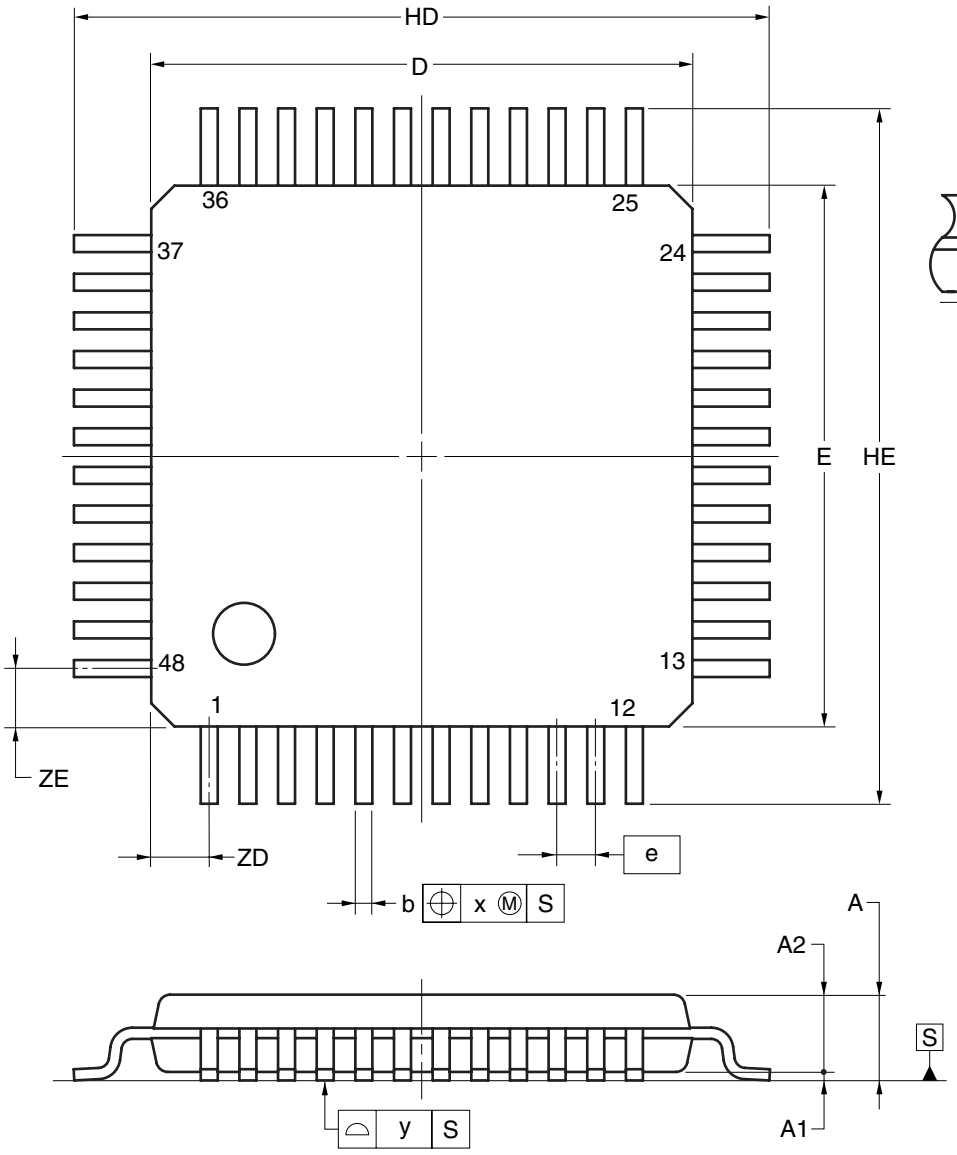
Remark When updating data multiple times, use the flash memory as one for updating data.

CHAPTER 29 PACKAGE DRAWINGS

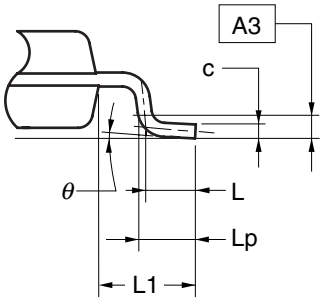
29.1 78K0R/KC3-L (48 pin product)

μPD78F1022GA-HAA-AX, 78F1023GA-HAA-AX, 78F1024GA-HAA-AX

48-PIN PLASTIC TQFP (FINE PITCH) (7x7)



detail of lead end



(UNIT:mm)

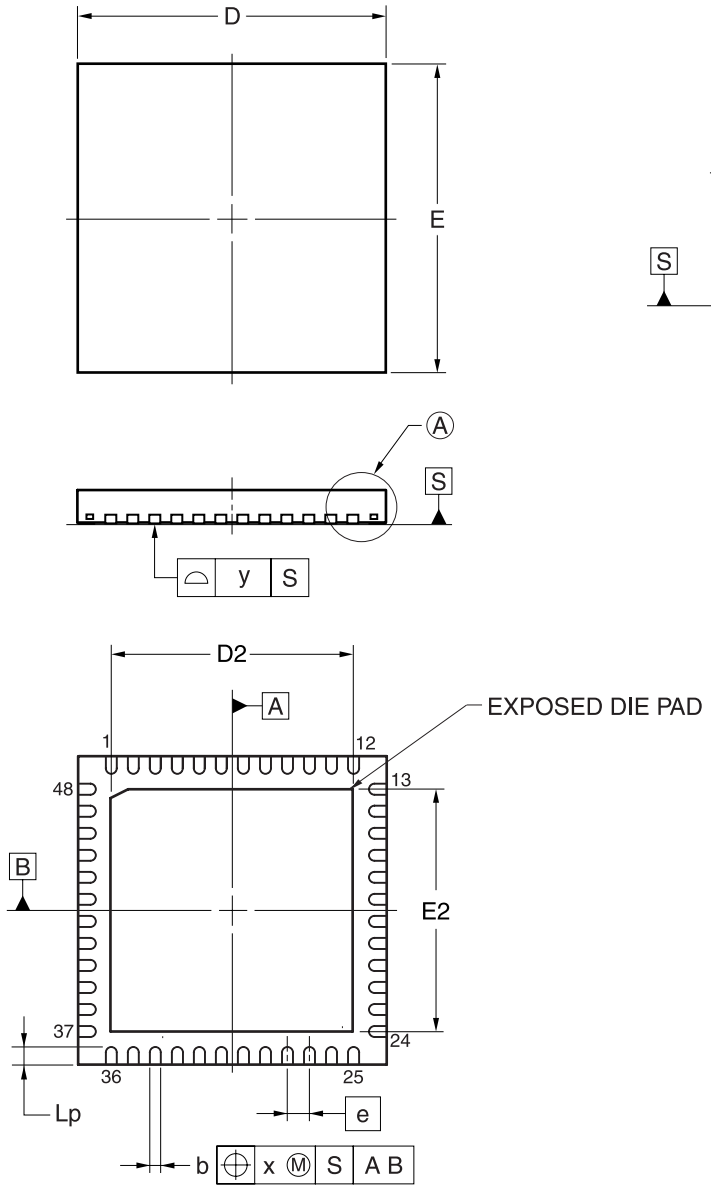
ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.20 MAX.
A1	0.10±0.05
A2	1.00±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

P48GA-50-HAA

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

<R> μPD78F1022K8-5B4-AX, 78F1023K8-5B4-AX, 78F1024K8-5B4-AX

48-PIN PLASTIC WQFN(7x7)



(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.05
E	7.00±0.05
D2	5.50 MAX.
E2	5.50 MAX.
A	0.75±0.05
b	$0.25^{+0.05}_{-0.07}$
e	0.50
Lp	0.40±0.10
x	0.05
y	0.05

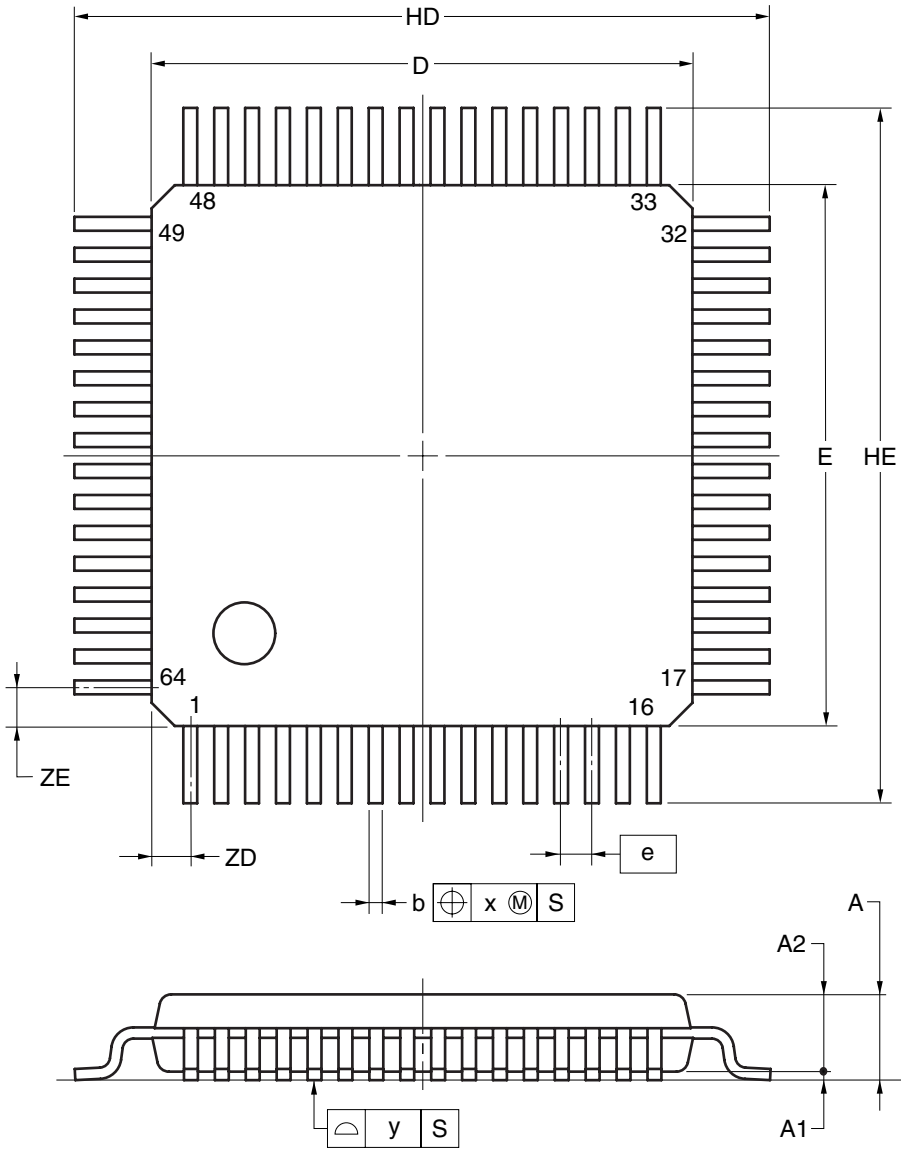
P48K8-50-5B4-2

NOTE
D2 and E2 value varies depending on die size.

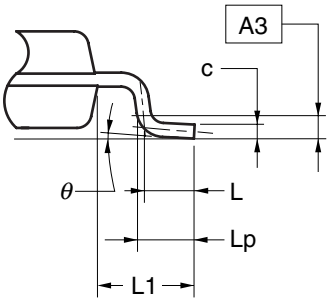
29.2 78K0R/KE3-L (64 pin product)

μPD78F1025GA-HAB-AX, 78F1026GA-HAB-AX

64-PIN PLASTIC TQFP (FINE PITCH) (7x7)



detail of lead end



(UNIT:mm)

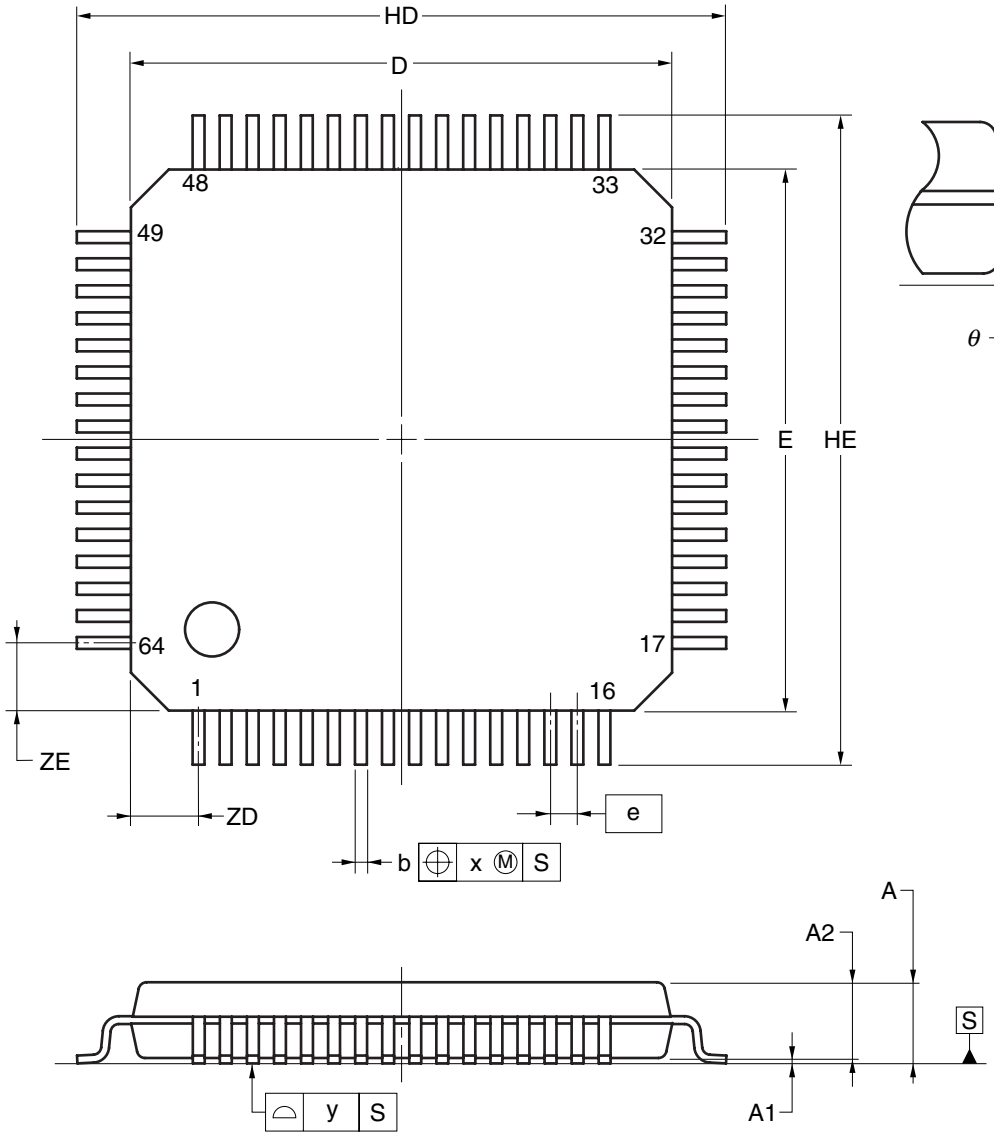
ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.20 MAX.
A1	0.10±0.05
A2	1.00±0.05
A3	0.25
b	$0.16^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.40
x	0.07
y	0.08
ZD	0.50
ZE	0.50

P64GA-40-HAB

NOTE
Each lead centerline is located within 0.07mm of its true position at maximum material condition.

μPD78F1025GB-GAH-AX, 78F1026GB-GAH-AX

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)



(UNIT:mm)

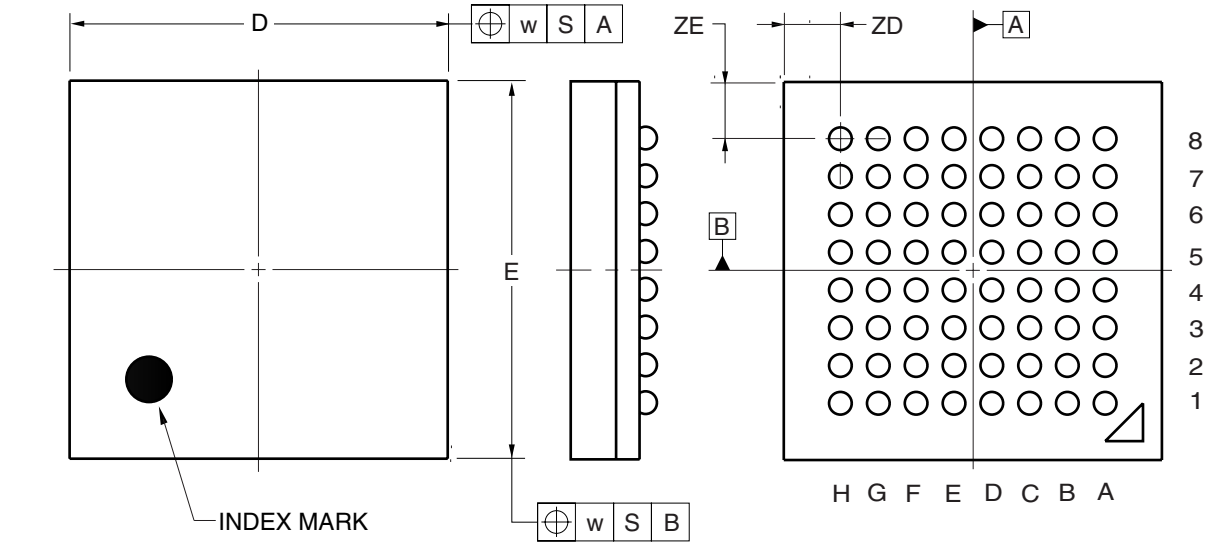
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

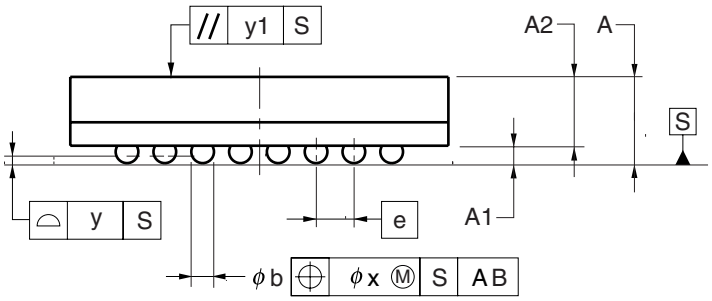
P64GB-50-GAH

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

μPD78F1025F1-AN9-A, 78F1026F1-AN9-A

64-PIN PLASTIC FBGA (5x5)





(UNIT:mm)

ITEM	DIMENSIONS
D	5.00±0.10
E	5.00±0.10
w	0.20
A	1.13±0.10
A1	0.22±0.05
A2	0.91
e	0.50
b	0.31±0.05
x	0.05
y	0.08
y1	0.20
ZD	0.75
ZE	0.75

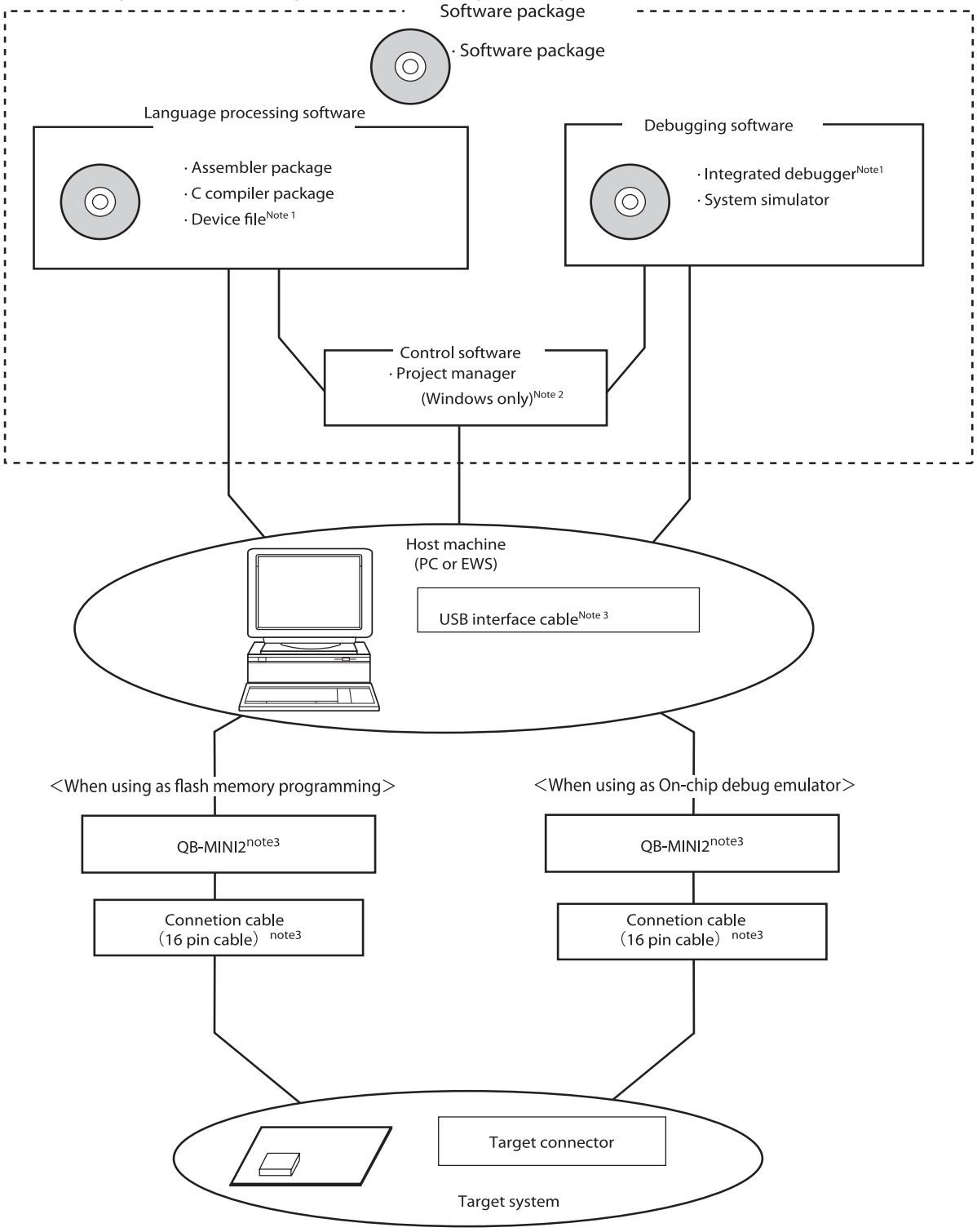
P64F1-50-AN9

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0R/KC3-L, KE3-L. Figure A-1 shows the development tool configuration.

Figure A-1. Development Tool Configuration

When using the on-chip debug emulator with programming function QB-MINI2



Notes

<R> **1.** Download device file (DF781026) and the integrated debugger (ID78K0R-QB) from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

<R> **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. Any other products are sold separately. In addition, 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.1 Software Package

SP78K0R 78K0R Series software package	Development tools (software) common to the 78K0R microcontrollers are combined in this package.
	Part number: μS××××SP78K0R

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0R

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

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 (DF781026). **<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.
	Part number: μS××××RA78K0R
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 (both sold separately). **<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.
	Part number: μS××××CC78K0R
DF781026 Note Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used.
	Part number: μS××××DF781026

Note The DF781026 can be used in common with the RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB.

Remark ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0R
μS××××CC78K0R

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

μS××××DF781026

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	

A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0R). It can only be used in Windows.

A.4 Flash Memory Programming Tools

A.4.1 When using flash memory programmers PG-FP5, FL-PR5

PG-FP5, FL-PR5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-xxxx Note Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use.

Note The part numbers of the flash memory programming adapter and the packages of the target device are described below.

<R>

Package		Flash Memory Programming Adapter
78K0R/KC3-L	48-pin plastic TQFP(Fine pitch)(7x7)	FA-78F1024GA-HAA-RX
	48 pin Plastic WQFN(7x7)	FA-78F1024K8-5B4-RX
78K0R/KE3-L	64 pin Plastic LQFP(Fine pitch)(10x10)	FA-78F1026GB-GAH-RX
	64 pin Plastic TQFP(Fine pitch)(7x7)	FA-78F1026GA-HAB-RX
	64 pin Plastic FBGA(5x5)	FA-78F1026F1-AN9-RX

Remark **1.**The FL-PR5 and FA-xxxx are the products of Naito Densei Machida Mfg. Co., Ltd.
Enquiries: Naito Densei Machida Mfg. Co., Ltd. (http://www.ndk-m.co.jp/) (TEL (042)750-4172)
2. Please use latest flash memory programming adapter.

A.4.2 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 also available as an on-chip debug emulator which serves to debug hardware and software while developing application systems using the 78K0R/KC3-L, KE3-L. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/KC3-L, KE3-L, use USB interface cable and 16-pin connection cable.

<R> **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).

A.5 Debugging Tools (Hardware)

A.5.1 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/KC3-L, KE3-L microcontrollers. It is also available as a flash memory programmer dedicated to microcontrollers with on-chip flash memory. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/Kx3-L use USB interface cable and 16-pin connection cable.

<R> **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).

A.6 Debugging Tools (Software)

SM+ for 78K0R System simulator	SM+ for 78K0R 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 SM+ for 78K0R 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. SM+ for 78K0R should be used in combination with the device file (DF781026).
	Part number: μS××××SM781000
ID78K0R-QB Integrated debugger	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. ID78K0R-QB should be used in combination with the device file (DF781026).
	Part number: μS××××ID78K0R-QB

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM781000
μS××××ID78K0R-QB

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX B REVISION HISTORY

B.1 Major Revisions in This Edition

(1/3)

Page	Description	Classification
Major Revisions in Rev.2.01		
CHAPTER 1 OUTLINE		
p.17	Change of description of Power supply voltage	(a)
CHAPTER 4 PORT FUNCTIONS		
p.150	Change of **Figure 4-30. Format of Port Mode Register (78K0R/KC3-L)**	(b)
CHAPTER 13 USB FUNCTION CONTROLLER (USBF)		
p.578	Change of **Caution** in **13.1 Overview**	(c)
Major Revisions in Rev.2.00		
Throughout		
Throughout	Change of status of μPD78F1022, 78F1023, 78F1024, 78F1025, and 78F1026 from under development to mass production	(d)
Throughout	Addition of 48 pin Plastic WQFN (7x7)	(d)
Throughout	Change URL of Renesas Electronics website	–
Throughout	Deletion of target from the capacitor (0.47 to 1 μF) connected to the REGC pin	(b)
CHAPTER 1 OUTLINE		
p.20	Addition of 64 pin Plastic FBGA(5x5)	(d)
p.26	Change of item of reset in **1.8 Outline of Functions**	(b)
CHAPTER 3 CPU ARCHITECTURE		
pp.52 to 54	Addition of **Remark** to **Figure 3-1 Memory Map (μPD78F1022)** to **Figure 3-3 Memory Map (μPD78F1024, 78F1026)**	(c)
p.67	Addition of **Caution** to **3.2.2 General-purpose registers**	(c)
pp.77, 78	Change of **Table 3-6. Extended SFR (2nd SFR) List (1/21)** and **Table 3-6. Extended SFR (2nd SFR) List (2/21)**	(a)
CHAPTER 4 PORT FUNCTIONS		
p.157	Addition of **Caution** to **Figure 4-38. Format of A/D Port Configuration Register (ADPC)**	(c)
CHAPTER 5 CLOCK GENERATOR		
p.170	Addition of **Note** to **Figure 5-3. Format of System Clock Control Register (CKC)**	(c)
p.172	Addition of **Note** to **Figure 5-4. Format of Clock Operation Status Control Register (CSC)**	(c)
p.176	Change of **Caution** of **Figure 5-6 Format of Oscillation Stabilization Time Select Register (OSTS)**	(c)
p.177	Addition of **Caution** to **Figure 5-7. Format of 20 MHz Internal High-Speed Oscillation Control Register (DSCCTL)**	(c)
p.180	Change of **(8) Operation speed mode control register (OSMC)** of **5.3 Registers Controlling Clock Generator**	(c)
pp.189 to 192	Change of **Figure 5-15. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))** and **Figure 5-16. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))**	(c)
p.202	Change of **Figure 5-17. CPU Clock Status Transition Diagram** and addition of **Caution**	(c)
pp.203, 204, 207	Change of **Table 5-4 CPU Clock Transition and SFR Register Setting Examples (1/6)**, **Table 5-4 CPU Clock Transition and SFR Register Setting Examples (2/6), and Table 5-4 CPU Clock Transition and SFR Register Setting Examples (5/6),**	(c)
p.212	Change of **Table 5-9. Maximum Number of Clocks Required for $f_{MAIN} \leftrightarrow f_{SUB}$**	(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 7 REAL-TIME COUNTER		
p.308	Addition of **Note** to **Figure 7-18. Procedure for Starting Operation of Real-Time Counter**	(c)
p.313	Addition of **Caution** to **Figure 7-23. 32.768 kHz Output Setting Procedure** and **Figure 7-24. 512 Hz, 16.384 kHz output Setting Procedure**	(c)
CHAPTER 8 WATCHDOG TIMER		
p.323	Change of **Table 8-3. Setting of Overflow Time of Watchdog Timer**	(c)
pp.324, 325	Change of **8.4.3 Setting window open period of watchdog timer**	(b)
CHAPTER 10 A/D CONVERTER		
p.341	Addition of **Caution** to **Figure 10-9. Format of A/D Port Configuration Register (ADPC)**	(c)
p.344	Change of **Figure 10-11. Basic Operation of A/D Converter**	(b)
p.354	Change of **(9) Conversion results just after A/D conversion start** of **10.6 Cautions for A/D Converter**	(b)
p.355	Change of **Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	(b)
p.355	Addition of **(12) Starting the A/D converter** to **10.6 Cautions for A/D Converter**	(c)
CHAPTER 11 SERIAL ARRAY UNIT		
p.372	Addition of **Note** to **Figure 11-8. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)**	(c)
p.373	Addition of **Caution** to **Figure 11-9. Format of Serial Data Register mn (SDRmn)**	(c)
CHAPTER 12 SERIAL INTERFACE IICA		
p.509	Change of **Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (3/4)**	(c)
p.511	Change of Caution of **(3) IICA status register (IICS)** in **12.3 Registers Controlling Serial Interface IICA**	(c)
p.512	Change of **Figure 12-7. Format of IICA Status Register (IICS) (2/3)**	(c)
p.519	Change of **12.4.2 Setting transfer clock by using IICWL and IICWH registers**	(c)
CHAPTER 13 USB FUNCTION CONTROLLER (USBF)		
p.561	Change of **13.3.2 Connection configuration**	(b), (c)
CHAPTER 16 INTERRUPT FUNCTIONS		
p.796	Change of **16.4.4 Interrupt request hold**	(c)
CHAPTER 18 STANDBY FUNCTION		
pp.809, 810	Change of **Figure 18-4. HALT Mode Release by Reset**	(c)
pp.812 to 814	Change of **Figure 18-5. STOP Mode Release by Interrupt Request Generation** and **Figure 18-6. STOP Mode Release by Reset**	(c)
CHAPTER 19 RESET FUNCTION		
p.815	Addition of description of internal reset by a reset processing check error to **CHAPTER 19 RESET FUNCTION**	(c)
p.816	Change of **Figure 19-1. Block Diagram of Reset Function**	(c)
p.817	Change of **Figure 19-2. Timing of Reset by $\overline{\text{RESET}}$ Input**	(c)
p.817	Change of **Figure 19-3. Timing of Reset Due to Watchdog Timer Overflow**	(c)
p.818	Change of **Figure 19-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	(c)
p.823	Change of **Table 19-2. Hardware Statuses after Reset Acknowledgment (4/4)**	(c)
p.825	Change of **Figure 19-5. Format of Reset Control Flag Register (RESF)**	(c)
p.826	Change of **Table 19-3. RESF Status when Reset Request Is Generated**	(c)
CHAPTER 20 POWER-ON-CLEAR CIRCUIT		
pp.829, 830	Change of **Figure 20-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector**	(c)
pp.831, 832	Change of **Figure 20-3. Example of Software Processing After Reset Release**	(c)
CHAPTER 21 LOW-VOLTAGE DETECTOR		
pp.852, 853	Change of **Figure 21-11. Example of Software Processing After Reset Release**	(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

(3/3)

Page	Description	Classification
CHAPTER 23 OPTION BYTE		
p.859	Change of **Figure 23-1. Format of User Option Byte (000C0H/010C0H)**	(c)
CHAPTER 24 FLASH MEMORY		
p.867	Addition of **Note** to **24.3 Communication Mode**	(c)
p.872	Addition of **Note** to **24.6.3 Selecting communication mode**	(c)
p.876	Addition of **Caution** and **Remark** to **24.8 Flash Memory Programming by Self-Programming**	(c)
p.877	Change of **Figure 24-11. Flow of Self Programming (Rewriting Flash Memory)**	(c)
CHAPTER 25 ON-CHIP DEBUG FUNCTION		
p.882	Change of **Table 25-1. Differences Between 1-Line Mode and 2-Line Mode**	(c)
CHAPTER 28 ELECTRICAL SPECIFICATIONS		
p.911	Change of **Absolute Maximum Ratings (T_A = 25°C)**	(b)
p.913	Change of **Internal Oscillator Characteristics**	(b)
pp.920, 921	Change of supply current of **DC Characteristics (7/9)** and **DC Characteristics (8/9)**	(b)
p.914	Addition of **Recommended oscillator circuit constants**	(b)
p.923	Addition of USB operating current and **Caution** to **DC Characteristics (9/9)**	(c)
APPENDIX A DEVELOPMENT TOOLS		
p.964	Addition of flash memory programming adapter and the packages	(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

78K0R/KC3-L, 78K0R/KE3-L (On-Chip USB Controller)
User's Manual: Hardware

Publication Date: Rev.0.01 Jan 15, 2010
Rev.2.01 Sep 22, 2011

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

© 2011 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0

78K0R/KC3-L, 78K0R/KE3-L
(On-Chip USB Controller)

Renesas Electronics Corporation

R01UH0019EJ0201