User's Manual

8

78K0/Kx2-C

User's Manual: Hardware

8-Bit Single-Chip Microcontrollers

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 78K0/Kx2-C and design and develop application systems and programs for these devices.
The target products are as follows.

- 78K0/KC2-C: μPD78F0760, 78F0761, 78F0762
- 78K0/KE2-C: μPD78F0763, 78F0764, 78F0765

Purpose

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

Organization

The 78K0/Kx2-C manual is separated into two parts: this manual and the instructions edition (common to the 78K0 Microcontroller).

78K0/Kx2-C User's Manual (This Manual)	**78K/0 Series 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 RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.
- To know details of the 78K0 Microcontroller instructions:
  → Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

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.
78K0/Kx2-C User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E
78K0 Microcontrollers Self Programming Library Type01 User's Manual	U18274E
78K0 Microcontrollers EEPROM™ Emulation Library Type01 User's Manual[Note]	U18275E

Note This document is classified under engineering management. Contact a Renesas Electronics sales representative.

Documents Related to Development Tools (Software)

Document Name		Document No.
RA78K0 Ver.3.80 Assembler Package User's Manual	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
CC78K0 Ver.3.70 C Compiler User's Manual	Operation	U17201E
	Language	U17200E
ID78K0-QB Ver.3.00 Integrated Debugger User's Manual	Operation	U18492E
PM+ Ver.6.30 User's Manual		U18416E
CubeSuite+ V1.00.00 Integrated Development Environment [Note]	Start	R20UT0545E
	78K0 Design	R20UT0546E
	78K0 Coding	R20UT0551E
	78K0 Build	R20UT0555E
	78K0 Debug	R20UT0559E
	Analysis	R20UT0563E
	Message	R20UT0407E

Note Confirm the latest information of CubeSuite+ by referring to the following website.
http://www.renesas.com/cubesuite+

Documents Related to Development Tools (Hardware)

Document Name	Document No.
QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual	R20UT0449E
QB-78K0KX2C In-Circuit Emulator	U19841E

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

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R02UT0008E

Other Documents

Document Name	Document No.
RENESAS MICROCOMPUTER GENERAL CATALOG	R01CS0001E
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

Note See the "Semiconductor Device Mount Manual" website (http://www.renesas.com/prod/package/manual/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.

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 (0.1 μs: @ 20 MHz operation with high-speed system clock) to ultra low-speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM, RAM capacities

Flash ROM[Note]	RAM[Note]	78K0/KC2-C	78K0/KE2-C
		48 Pins	64 Pins
60 KB	3 KB	μPD78F0762	μPD78F0765
48 KB	2 KB	μPD78F0761	μPD78F0764
32 KB	1 KB	μPD78F0760	μPD78F0763

Note The internal flash memory, internal high-speed RAM capacities, and internal expansion RAM capacities can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS). For IMS and IXS, see **27.1 Internal Memory Size Switching Register** and **27.2 Internal Expansion RAM Size Switching Register**.

- On-chip internal high-speed oscillation clocks
  - 8 MHz Internal high-speed oscillation clock: 8 MHz±5 %
- On-chip single-power-supply flash memory
- Self-programming (with boot swap 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 ÷ 16 bits)
- On-chip key interrupt function
- On-chip clock output/buzzer output controller
- I/O ports
  78K0/KC2-C: 41 (N-ch open drain: 10)
  78K0/KE2-C: 55 (N-ch open drain: 12)
- Timer: 9 channels
  - 16-bit timer/event counter: 3 channels[Note]
  - 8-bit timer/event counter: 2 channels
  - 8-bit timer: 2 channels
  - Real-time counter: 1 channel
  - Watchdog timer: 1 channel

Note The TM01 and TM02 of 78K0/KC2-C are not provided with the timer I/O pins.

- Serial interface
  - CSI: 1 channel/UART: 1 channel
  - CSI: 1 channel[Note 1]
  - UART (supporting LIN (Local Interconnect Network) -bus): 1 channel
  - UART: 1 channel[Note 2]
  - I^2C: 3 channels
- 10-bit resolution A/D converter (AV_{REF} = 2.3 to 5.5 V): 8 channels
- CEC transmission/reception circuit
- Remote controller receiver
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = –40 to +85°C

Notes 1. Only CSI11 of the 78K0/KE2-C supports SPI.
2. 78K0/KE2-C only

1.2 Applications

- Digital audio visual equipment

1.3 Ordering Information

- **Flash memory version**

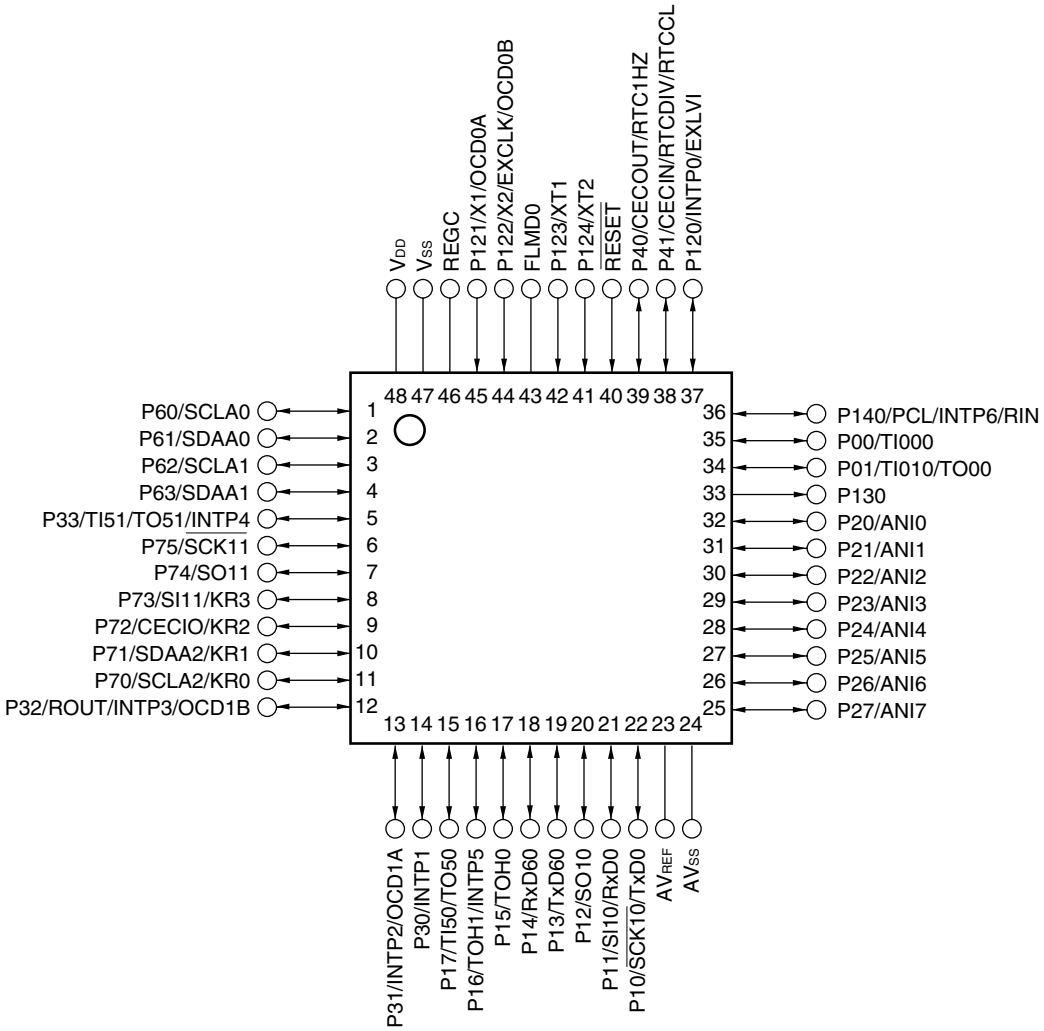
78K0/Kx2-C Microcontroller	Package	Part Number
78K0/KC2-C	48-pin plastic LQFP (fine pitch) (7 × 7)	μPD78F0760GA-GAM-AX, 78F0761GA-GAM-AX, 78F0762GA-GAM-AX
78K0/KE2-C	64-pin plastic LQFP (fine pitch) (10 × 10)	μPD78F0763GB-GAH-AX, 78F0764GB-GAH-AX, 78F0765GB-GAH-AX

Caution The 78K0/Kx2-C 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)

1.4.1 78K0/KC2-C

- 48-pin plastic LQFP (fine pitch) (7 × 7)



Cautions 1. Make AVSS the same potential as VSS.

2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF: recommended).

3. ANI0/P20 to ANI7/P27 are set in the analog input mode after release of reset.

Remark INTP4/INTDA, INTP5/INTCE/INTERR: If CEC transmission/reception circuit is used, INTP4 and INTP5 cannot be used. If INTP4 or INTP5 is used, CEC transmission/reception circuit cannot be used.

INTP6/INTRIN/PCL: Only one of INTP6, remote controller receiver, and clock output (PCL) can be used.

INTCSI10/INTST0: Only one of CSI10 and UART0 can be used.

INTRTC/INTRTCI: Only one of fixed-cycle signal of real-time counter/alarm match detection and interval signal detection of real-time counter can be used.

INTKR/INTIICA2: Only one of key interrupt function and IICA02 can be used.

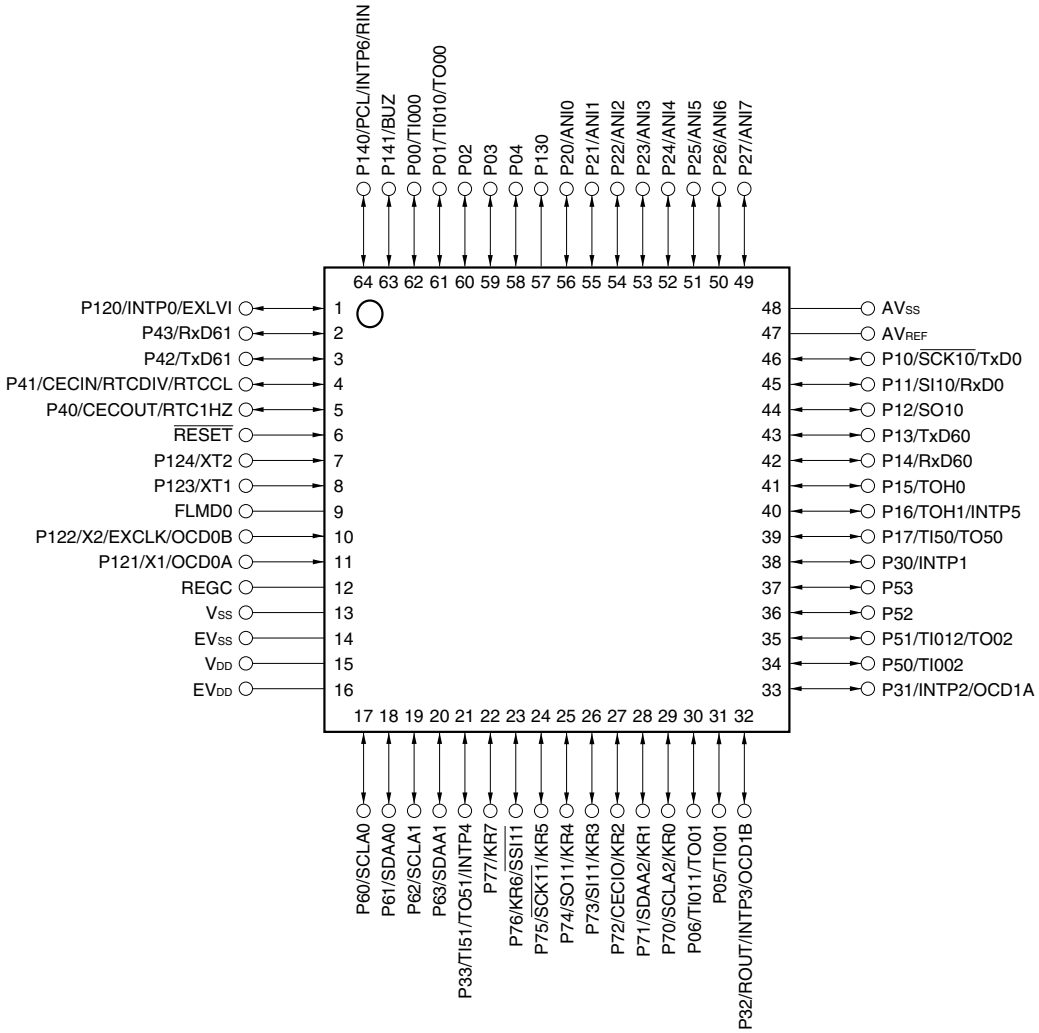
INTTM012/INTDMU: If INTTM012 occurs for TM02, only one of TM02 and multiplier/divider can be used.

Pin Identification

ANI0 to ANI7:	Analog input	P140:	Port 14
AV_{REF}:	Analog reference voltage	PCL:	Programmable clock output
AV_{SS}:	Analog ground	REGC:	Regulator capacitance
CECIN:	Consumer electronics control input	$\overline{RESET}$:	Reset
		RIN:	Remote control input
CECIO:	Consumer electronics control input/output	ROUT:	Remote control output
		RTC1HZ:	Real-time counter correction clock (1 Hz) output
CECOUT:	Consumer electronics control output	RTCCL:	Real-time counter clock (32 kHz original oscillation) output
EXCLK:	External clock input (main system clock)	RTCDIV:	Real-time counter clock (32 kHz divided frequency) output
EXLVI:	External potential input for low-voltage detector	RxD0, RxD60:	Receive data
FLMD0:	Flash programming mode	$\overline{SCK10}$, $\overline{SCK11}$:	Serial clock input/output
INTP0 to INTP6:	External interrupt input	SCLA0, SCLA1, SCLA2:	Serial clock input/output
KR0 to KR3:	Key return	SDAA0, SDAA1, SDAA2:	Serial data input/output
OCD0A, OCD0B, OCD1A, OCD1B	On chip debug input/output	SI10, SI11:	Serial data input
		SO10, SO11:	Serial data output
P02, P01:	Port 0	TI000, TI010, TI50, TI51:	Timer input
P10 to P17:	Port 1	TO00, TO50, TO51, TOH0, TOH1:	Timer output
P20 to P27:	Port 2		
P30 to P33:	Port 3	TxD0, TxD60:	Transmit data
P40, P41:	Port 4	V_{DD}:	Power supply
P60 to P63:	Port 6	V_{SS}:	Ground
P70 to P75:	Port 7	X1, X2:	Crystal oscillator (main system clock)
P120 to P124:	Port 12		
P130:	Port 13	XT1, XT2:	Crystal oscillator (subsystem clock)

1.4.2 78K0/KE2-C

- 64-pin plastic LQFP (fine pitch) (10 × 10)



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

2. Make EVDD the same potential as VDD.

3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF: recommended).

4. ANI0/P20 to ANI7/P27 are set in the analog input mode after release of reset.

Remark INTP4/INTDA, INTP5/INTCE/INTERR: If CEC transmission/reception circuit is used, INTP4 and INTP5 cannot be used. If INTP4 or INTP5 is used, CEC transmission/reception circuit cannot be used.

INTP6/INTRIN/PCL: Only one of INTP6, remote controller receiver, and clock output (PCL) can be used.

INTCSI10/INTST0: Only one of CSI10 and UART0 can be used.

INTRTC/INTRTCI: Only one of fixed-cycle signal of real-time counter/alarm match detection and interval signal detection of real-time counter can be used.

INTKR/INTIICA2: Only one of key interrupt function and IICA02 can be used.

INTTM001/INTSR6/INTSRE6: If INTTM001 occurs for TM01, only one of TM01 and receiving of data using UART61 can be used.

INTTM011/INTST6: If INTTM011 occurs for TM01, only one of TM01 and transmitting of data using UART61 can be used.

INTTM012/INTDMU: If INTTM012 occurs for TM02, only one of TM02 and multiplier/divider can be used.

Pin Identification

ANI0 to ANI7:	Analog input	P140, P141:	Port 14
AV_{REF}:	Analog reference voltage	PCL:	Programmable clock output
AV_{SS}:	Analog ground	REGC:	Regulator capacitance
BUZ:	Buzzer output	$\overline{RESET}$:	Reset
CECIN:	Consumer electronics control input	RIN:	Remote control input
		ROUT:	Remote control output
CECIO:	Consumer electronics control input/output	RTC1HZ:	Real-time counter correction clock (1 Hz) output
CECOUT:	Consumer electronics control output	RTCCL:	Real-time counter clock (32 kHz original oscillation) output
EV_{DD}:	Power supply for port	RTCDIV:	Real-time counter clock (32 kHz divided frequency) output
EV_{SS}:	Ground for port		
EXCLK:	External clock input (main system clock)	RxD0, RxD60, RxD61:	Receive data
		$\overline{SCK10}$, $\overline{SCK11}$:	Serial clock input/output
EXLVI:	External potential input for low-voltage detector	SCLA0, SCLA1, SCLA2:	Serial clock input/output
		SDAA0, SDAA1, SDAA2:	Serial data input/output
FLMD0:	Flash programming mode	SI10, SI11:	Serial data input
INTP0 to INTP6:	External interrupt input	SO10, SO11:	Serial data output
KR0 to KR7:	Key return	$\overline{SSI11}$:	Serial Interface ohip select input
OCD0A, OCD0B, OCD1A, OCD1B	On chip debug input/output	TI000, TI001, TI002, TI010, TI011, TI012, TI50, TI51:	Timer input
P00 to P06:	Port 0		
P10 to P17:	Port 1	TO00, TO01, TO02, TO50, TO51, TOH0, TOH1:	Timer output
P20 to P27:	Port 2		
P30 to P33:	Port 3		
P40 to P43:	Port 4	TxD0, TxD60, TxD61:	Transmit data
P50 to P53:	Port 5	V_{DD}:	Power supply
P60 to P63:	Port 6	V_{SS}:	Ground
P70 to P77:	Port 7	X1, X2:	Crystal oscillator (main system clock)
P120 to P124:	Port 12		
P130:	Port 13	XT1, XT2:	Crystal oscillator (subsystem clock)

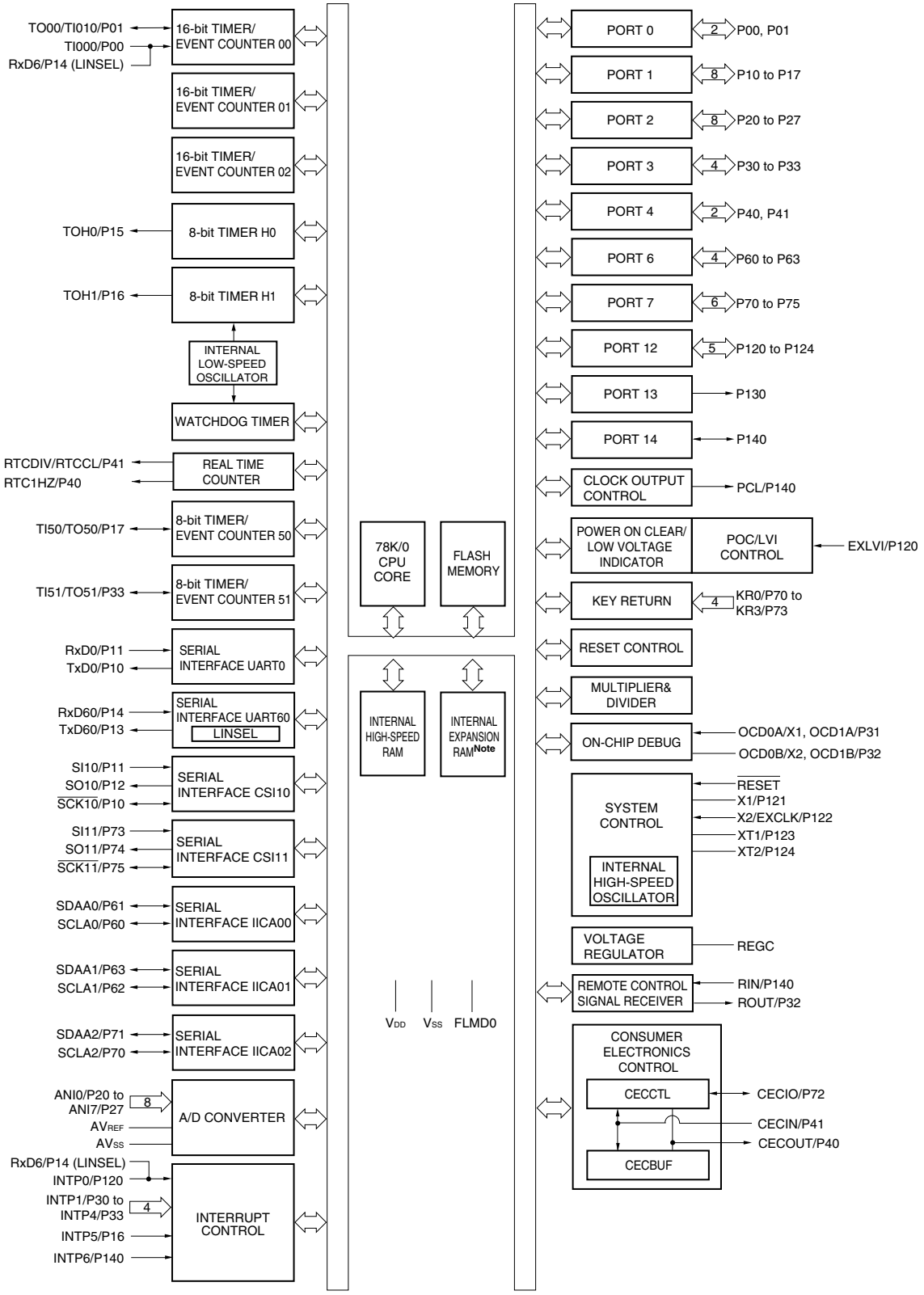
1.5 78K0/Kx2-C Microcontroller Lineup

ROM[Note]	RAM[Note]	78K0/KC2-C	78K0/KE2-C
		48 Pins	64 Pins
60 KB	3 KB	μPD78F0762	μPD78F0765
48 KB	2 KB	μPD78F0761	μPD78F0764
32 KB	1 KB	μPD78F0760	μPD78F0763

Note The internal flash memory, internal high-speed RAM capacities, and internal expansion RAM capacities can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS). For IMS and IXS, see **27.1 Internal Memory Size Switching Register** and **27.2 Internal Expansion RAM Size Switching Register**.

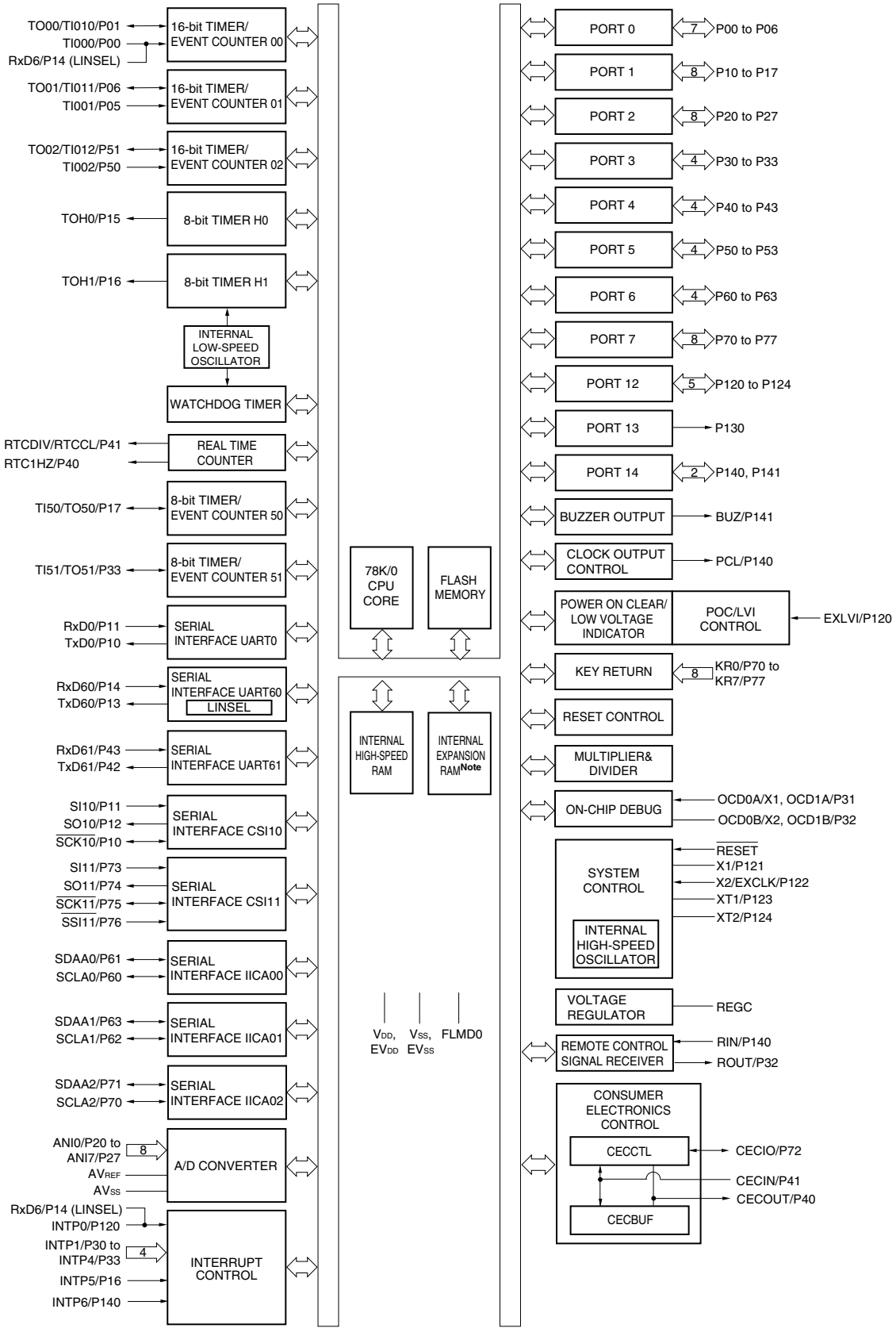
1.6 Block Diagram

1.6.1 78K0/KC2-C



Note The products with flash memory 48 KB and 60 KB only.

1.6.2 78K0/KE2-C



Note The products with flash memory 48 KB and 60 KB only.

1.7 Outline of Functions

Item			78K0/KC2-C			78K0/KE2-C		
			μPD78F0760	μPD78F0761	μPD78F0762	μPD78F0763	μPD78F0764	μPD78F0765
Flash memory (KB)			32	48	60	32	48	60
High-Speed RAM (KB)			1	1	1	1	1	1
Expansion RAM (KB)			–	1	2	–	1	2
Power supply voltage			V_{DD} = 1.8 to 5.5 V					
Regulator			Provided					
Minimum instruction execution time			0.1 μs (20 MHz)					
Clock	Main	High-speed system	2 to 20 MHz					
		Internal high-speed oscillation	8 MHz±5 %					
	Subsystem		32.768 kHz (TYP.)					
	Internal low-speed oscillation		240 kHz±10%					
Port	Total		41			55		
	N-ch O.D. (6 V tolerance)		10			12		
Timer	16 bits (TM0)		3 ch[Note]					
	8 bits (TM5)		2 ch					
	8 bits (TMH)		2 ch					
	Real-time counter (RTC)		1 ch					
	Watchdog timer (WDT)		1 ch					
Serial interface	3-wire CSI/UART		1 ch					
	3-wire CSI		1 ch			1 ch (supporting SPI)		
	UART supporting LIN-bus		1 ch					
	UART		–			1 ch		
	I^2C bus		3 ch					
CEC transmission/reception circuit			Provided					
Remote controller receiver			Provided					
10-bit A/D			8 ch					
Interrupt	External		8					
	Internal		27					
Key interrupt			4 ch			8 ch		
Reset	$\overline{\text{RESET}}$ pin		Provided					
	POC		1.59 V ±0.15 V					
	LVI		The detection level of the supply voltage is selectable.					
	WDT		Provided					
Clock output/buzzer output			Clock output only			Provided		
Multiplier/divider			Provided					
On-chip debug function			Provided					
Operating ambient temperature			T_A = –40 to +85°C					
Package			48-pin plastic LQFP (fine pitch) (7 × 7)			64-pin plastic LQFP (fine pitch) (10 × 10)		

Note The TM01 and TM02 of 78K0/KC2-C are not provided with the timer I/O pins.

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters 00, 01, and 02			8-Bit Timer/ Event Counters 50 and 51		8-Bit Timers H0 and H1		Watchdog Timer
		TM00	TM01 Note 1	TM02 Note 1	TM50	TM51	TMH0	TMH1	
Function	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	–
	External event counter	1 channel	1 channel Note 2	1 channel Note 2	1 channel	1 channel	–	–	–
	PPG output	1 output	1 output Note 2	1 output Note 2	–	–	–	–	–
	PWM output	–	–	–	1 output	1 output	1 output	1 output	–
	Pulse width measurement	2 inputs	2 inputs Note 2	2 output Note 2	–	–	–	–	–
	Square-wave output	1 output	1 output Note 2	1 output Note 2	1 output	1 output	1 output	1 output	–
	Carrier generator	–	–		–	–	–	1 output Nore 3	–
	Watchdog timer	–	–		–	–	–	–	1 channel
Interrupt source		2	2	2	1	1	1	1	–

Notes 1. The TM01 and TM02 of 78K0/KC2-C are not provided with the timer I/O pins.

2. 78K0/KE2-C only.

3. TM51 and TMH1 can be used in combination as a carrier generator mode.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

Pin I/O buffer power supplies depend on the product. The relationship between these power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies (AV_{REF}, V_{DD})

- 78K0/KC2-C: 48-pin plastic LQFP (fine pitch) (7x7)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27
V_{DD}	Pins other than P20 to P27

Table 2-2. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD}, V_{DD})

- 78K0/KE2-C: 64-pin plastic LQFP (fine pitch) (10x10)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27
EV_{DD}	Port pins other than P20 to P27 and P121 to P124
V_{DD}	• P121 to P124 • Pins other than port

2.1.1 78K0/KC2-C

(1) Port functions (1/2): 78K0/KC2-C

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI000
P01				TI010/TO00
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{\text{SCK10}}$/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD60
P14				RxD60
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 to P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0 to ANI7
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1
P31				INTP2/OCD1A
P32				ROUT/INTP3/ OCD1B
P33				TI51/TO51/INTP4
P40	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	CECOUT/RTC1HZ
P41				CECIN/RTCDIV/ RTCCL
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0
P61				SDAA0
P62				SCLA1
P63				SDAA1
P70	I/O	Port 7. 6-bit I/O port. Output of P70 to P75 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. Only for P72, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA2/KR0
P71				SDAA2/KR1
P72				CECIO/KR2
P73				SI11/KR3
P74				SO11
P75				$\overline{\text{SCK11}}$
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. Only for P120, input/output can be specified. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2

(1) Port functions (2/2): 78K0/KC2-C

Function Name	I/O	Function	After Reset	Alternate Function
P130	Output	Port 13. 1-bit output-only port.	Output port	–
P140	I/O	Port 14. 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	PCL/INTP6/RIN

(2) Non-port functions (1/2): 78K0/KC2-C

Function Name	I/O	Function	After Reset	Alternate Function
ANI0 to ANI7	Input	A/D converter analog input	Analog input	P20 to P27
CECIN	Input	Serial data input for CEC	Input port	P41/RTCDIV/ RTCCL
CECIO	I/O	Serial data I/O for CEC	Input port	P72/KR2
CECOUT	Output	Serial data output for CEC	Input port	P40/RTC1HZ
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
FLMD0	–	Flash memory programming mode setting	–	–
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P120/EXLVI
INTP1				P30
INTP2				P31/OCD1A
INTP3				P32/ROUT/ OCD1B
INTP4				P33/TI51/TO51
INTP5				P16/TOH1
INTP6				P140/PCL/RIN
KR0	Input	Key interrupt input	Input port	P70/SCLA2
KR1				P71/SDAA2
KR2				P72/CECIO
KR3				P73/SI11
PCL	Output	Clock output (for trimming of high-speed system clock, subsystem clock)	Input port	P140/INTP6/RIN
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF: recommended).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RIN	Input	Remote control reception data input	Input port	P140/PCL/INTP6
ROUT	Output	Remote control reception data output	Input port	P32/INTP3/ OCD1B
RTC1HZ	Output	Real-time counter clock (1 Hz) output	Input port	P40/CECOUT
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P41/CECIN/ RTCDIV
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P41/CECIN/ RTCCL
RxD0	Input	Serial data input to UART0	Input port	P11/SI10
RxD60	Input	Serial data input to UART60	Input port	P14
$\overline{\text{SCK10}}$	I/O	Clock input/output for CSI10	Input port	P10/TxD0
$\overline{\text{SCK11}}$	I/O	Clock input/output for CSI11	Input port	P75
SCLA0	I/O	Clock input/output for IICA00	Input port	P60
SCLA1	I/O	Clock input/output for IICA01	Input port	P62
SCLA2	I/O	Clock input/output for IICA02	Input port	P70/KR0
SDAA0	I/O	Serial data I/O for IICA00	Input port	P61
SDAA1	I/O	Serial data I/O for IICA01	Input port	P63
SDAA2	I/O	Serial data I/O for IICA02	Input port	P71/KR1

(2) Non-port functions (2/2): 78K0/KC2-C

Function Name	I/O	Function	After Reset	Alternate Function
SI10	Input	Serial data input to CSI10	Input port	P11/RxD0
SI11	Input	Serial data input to CSI11	Input port	P73/KR3
SO10	Output	Serial data output from CSI10	Input port	P12
SO11	Output	Serial data output from CSI11	Input port	P74
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input port	P00
TI010	Input	Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00	Input port	P01/TO00
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input port	P17/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P33/TO51/INTP4
TO00	Output	16-bit timer/event counter 00 output	Input port	P01/TI010
TO50	Output	8-bit timer/event counter 50 output	Input port	P17/TI50
TO51		8-bit timer/event counter 51 output		P33/TI51/INTP4
TOH0	Output	8-bit timer H0 output	Input port	P15
TOH1		8-bit timer H1 output		P16/INTP5
TxD0	Output	Serial data output from UART0	Input port	P10/$\overline{\text{SCK10}}$
TxD60	Output	Serial data output from UART60	Input port	P13
X1	–	Connecting resonator for main system clock	Input port	P121/OCD0A
X2	–		Input port	P122/EXCLK/ OCD0B
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/OCD0B
XT1	–	Connecting resonator for subsystem clock	Input port	P123
XT2	–		Input port	P124
V_{DD}	–	Positive power supply for pins other than P20 to P27	–	–
AV_{REF}	–	A/D converter reference voltage input and positive power supply for P20 to P27 and A/D converter	–	–
V_{SS}	–	Ground potential for pins other than P20 to P27	–	–
AV_{SS}	–	A/D converter ground potential. Make the same potential as V_{SS}.	–	–
OCD0A	Input	Connection for on-chip debug mode setting pins	Input port	P121/X1
OCD1A				P31/INTP2
OCD0B	–			P122/X2/EXCLK
OCD1B				P32/ROUT/ INTP3

2.1.2 78K0/KE2-C

(1) Port functions (1/2): 78K0/KE2-C

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 7-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	TI000
P01				TI010/TO00
P02				–
P03				–
P04				–
P05				TI001
P06				TI011/TO01
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{\text{SCK10}}$/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD60
P14				RxD60
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 to P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0 to ANI7
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1
P31				INTP2/OCD1A
P32				ROUT/INTP3/ OCD1B
P33				TI51/TO51/INTP4
P40	I/O	Port 4. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	CECOUT/RTC1HZ
P41				CECIN/RTCDIV/ RTCCL
P42				TxD61
P43				RxD61
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	TI002
P51				TI012/TO02
P52, P53				–
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0
P61				SDAA0
P62				SCLA1
P63				SDAA1

(2) Port functions (2/2): 78K0/KE2-C

Function Name	I/O	Function	After Reset	Alternate Function
P70	I/O	Port 7. 8-bit I/O port. Output of P70 to P77 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. Only for P72, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA2/KR0
P71				SDAA2/KR1
P72				CECIO/KR2
P73				SI11/KR3
P74				SO11/KR4
P75				$\overline{\text{SCK11}}$/KR5
P76				KR6/$\overline{\text{SSI11}}$
P77				KR7
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. Only for P120, input/output can be specified. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2
P130	Output	Port 13. 1-bit output-only port.	Output port	–
P140	I/O	Port 14. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCL/INTP6/RIN
P141				BUZ

(2) Non-port functions (1/3): 78K0/KE2-C

Function Name	I/O	Function	After Reset	Alternate Function
ANI0 to ANI7	Input	A/D converter analog input	Analog input	P20 to P27
BUZ	Output	Buzzer output	Input port	P141
CECIN	Input	Serial data input for CEC	Input port	P41/RTCDIV/ RTCCL
CECIO	I/O	Serial data I/O for CEC	Input port	P72/KR2
CECOUT	Output	Serial data output for CEC	Input port	P40/RTC1HZ
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
FLMD0	–	Flash memory programming mode setting	–	–
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P120/EXLVI
INTP1				P30
INTP2				P31/OCD1A
INTP3				P32/ROUT/ OCD1B
INTP4				P33/TI51/TO51
INTP5				P16/TOH1
INTP6				P140/PCL/RIN
KR0	Input	Key interrupt input	Input port	P70/SCLA2
KR1				P71/SDAA2
KR2				P72/CECIO
KR3				P73/SI11
KR4				P74/SO11
KR5				P75/$\overline{\text{SCK11}}$
KR6				P76/$\overline{\text{SSI11}}$
KR7				P77
PCL	Output	Clock output (for trimming of high-speed system clock, subsystem clock)	Input port	P140/INTP6/RIN
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF: recommended).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RIN	Input	Remote control reception data input	Input port	P140/PCL/INTP6
ROUT	Output	Remote control reception data output	Input port	P32/INTP3/ OCD1B
RTC1HZ	Output	Real-time counter clock (1 Hz) output	Input port	P40/CECOUT
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P41/CECIN/ RTCDIV
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P41/CECIN/ RTCCL
RxD0	Input	Serial data input to UART0	Input port	P11/SI10
RxD60		Serial data input to UART60		P14
RxD61		Serial data input to UART61		P43

(2) Non-port functions (2/3): 78K0/KE2-C

Function Name	I/O	Function	After Reset	Alternate Function
$\overline{\text{SCK10}}$	I/O	Clock input/output for CSI10	Input port	P10/TxD0
$\overline{\text{SCK11}}$		Clock input/output for CSI11		P75/KR5
SCLA0	I/O	Clock input/output for IICA00	Input port	P60
SCLA1	I/O	Clock input/output for IICA01	Input port	P62
SCLA2	I/O	Clock input/output for IICA02	Input port	P70/KR0
SDAA0	I/O	Serial data I/O for IICA00	Input port	P61
SDAA1	I/O	Serial data I/O for IICA01	Input port	P63
SDAA2	I/O	Serial data I/O for IICA02	Input port	P71/KR1
SI10	Input	Serial data input to CSI10	Input port	P11/RxD0
SI11		Serial data input to CSI11		P73/KR3
SO10	Output	Serial data output from CSI10	Input port	P12
SO11		Serial data output from CSI11		P74/KR4
$\overline{\text{SSI11}}$	Input	Chip select input to CSI11	Input port	P76/KR6
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input port	P00
TI001		External count clock input to 16-bit timer/event counter 01 Capture trigger input to capture registers (CR001, CR011) of 16-bit timer/event counter 01		P05
TI002		External count clock input to 16-bit timer/event counter 02 Capture trigger input to capture registers (CR002, CR012) of 16-bit timer/event counter 02		P50
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TI011		Capture trigger input to capture register (CR001) of 16-bit timer/event counter 01		P06/TO01
TI012		Capture trigger input to capture register (CR002) of 16-bit timer/event counter 02		P51/TO02
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input port	P17/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P33/TO51/INTP4
TO00	Output	16-bit timer/event counter 00 output	Input port	P01/TI010
TO01		16-bit timer/event counter 01 output		P06/TI011
TO02		16-bit timer/event counter 02 output		P51/TI012
TO50	Output	8-bit timer/event counter 50 output	Input port	P17/TI50
TO51		8-bit timer/event counter 51 output		P33/TI51/INTP4
TOH0	Output	8-bit timer H0 output	Input port	P15
TOH1		8-bit timer H1 output		P16/INTP5
TxD0	Output	Serial data output from UART0	Input port	P10/$\overline{\text{SCK10}}$
TxD60		Serial data output from UART60		P13
TxD61		Serial data output from UART61		P42

(2) Non-port functions (3/3): 78K0/KE2-C

Function Name	I/O	Function	After Reset	Alternate Function
X1	Input	Connecting resonator for main system clock	Input port	P121/OCD0A
X2	–			P122/EXCLK/ OCD0B
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/OCD0B
XT1	Input	Connecting resonator for subsystem clock	Input port	P123
XT2	–		Input port	P124
V_{DD}	–	Positive power supply for P121 to P124 and other than ports	–	–
EV_{DD}	–	Positive power supply for ports other than P20 to P27 and P121 to P124. Make EV_{DD} the same potential as V_{DD}.	–	–
AV_{REF}	Input	A/D converter reference voltage input and positive power supply for P20 to P27 and A/D converter	–	–
V_{SS}	–	Ground potential for P121 to P124 and other than ports	–	–
EV_{SS}	–	Ground potential for ports other than P20 to P27 and P121 to P124. Make EV_{SS} the same potential as V_{SS}.	–	–
AV_{SS}	–	A/D converter ground potential. Make the same potential as V_{SS}.	–	–
OCD0A	Input	Connection for on-chip debug mode setting pins	Input port	P121/X1
OCD1A				P31/INTP2
OCD0B	–			P122/X2/EXCLK
OCD1B				P32/ROUT/ INTP3

2.2 Description of Pin Functions

Remark The pins mounted depend on the product. See **1.4 Ordering Information** and **2.1 Pin Function List**.

2.2.1 P00 to P06 (port 0)

P00 to P06 function as an I/O port. These pins also function as timer I/O.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P00/TI000	√	√
P01/TI010/TO00	√	√
P02	–	√
P03	–	√
P04	–	√
P05/TI001	–	√
P06/TI011/TO01	–	√

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P00 to P06 function as an I/O port. P00 to P06 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

P00 to P06 function as timer I/O.

(a) TI000, TI001

These are the pins for inputting an external count clock to 16-bit timer/event counters 00 and 01 and are also for inputting a capture trigger signal to the capture registers (CR000, CR010 or CR001, CR011) of 16-bit timer/event counters 00 and 01.

(b) TI010, TI011

These are the pins for inputting a capture trigger signal to the capture register (CR000 or CR001) of 16-bit timer/event counters 00 and 01.

(c) TO00, TO01

These are timer output pins of 16-bit timer/event counters 00 and 01.

2.2.2 P10 to P17 (port 1)

P10 to P17 function as an I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P10/$\overline{\text{SCK10}}$/TxD0	√	
P11/SI10/RxD0	√	
P12/SO10	√	
P13/TxD60	√	
P14/RxD60	√	
P15/TOH0	√	
P16/TOH1/INTP5	√	
P17/TI50/TO50	√	

Remark √: Mounted

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

(1) Port mode

P10 to P17 function as an I/O port. P10 to P17 can be set to input or output port in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P17 function as external interrupt request input, serial interface data I/O, clock I/O, and timer 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) RxD0

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

(e) RxD60

This is a serial data input pin of serial interface UART60.

(f) TxD0

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

(g) TxD60

This is a serial data output pin of serial interface UART60.

(h) TI50

This is the pin for inputting an external count clock to 8-bit timer/event counter 50.

(i) TO50

This is a timer output pin of 8-it timer/event counter 50.

(j) TOH0, TOH1

These are the timer output pins of 8-bit timers H0 and H1.

(k) 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.

2.2.3 P20 to P27 (port 2)

P20 to P27 function as an I/O port. These pins also function as pins for A/D converter analog input.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P20/ANI0	√	
P21/ANI1	√	
P22/ANI2	√	
P23/ANI3	√	
P24/ANI4	√	
P25/ANI5	√	
P26/ANI6	√	
P27/ANI7	√	

Remark √: Mounted

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

(1) Port mode

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

(2) Control mode

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, see **(5) ANI0/P20 to ANI7/P27** in **12.6 Cautions for A/D Converter**.

Caution ANI0/P20 to ANI7/P27 are set in the analog input mode after release of reset.

2.2.4 P30 to P33 (port 3)

P30 to P33 function as an I/O port. These pins also function as pins for external interrupt request input, remote control receive data output, and timer I/O.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P30/INTP1	√	
P31/INTP2/OCD1A	√	
P32/ROUT/INTP3/OCD1B	√	
P33/INTP4/TI51/TO51	√	

Remark √: Mounted

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

(1) Port mode

P30 to P33 function as an I/O port. P30 to P33 can be set to input or output port in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30 to P33 function as external interrupt request input, remote control receive data output, and timer I/O.

(a) INTP1 to 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) ROUT

This is a remote control receive data output pin.

(c) TI51

This is an external count clock input pin to 8-bit timer/event counter 51.

(d) TO51

This is a timer output pin from 8-bit timer/event counter 51.

Cautions 1. Be sure to pull the P31/INTP2/OCD1A pin down before a reset release, to prevent malfunction.

2. Process the P31/INTP2/OCD1A pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P31/INTP2/OCD1A
Flash memory programmer connection		Connect to EV_{SS}[Note] via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Input: Connect to EV_{DD}[Note] or EV_{SS}[Note] via a resistor. Output: Leave open.

Note With 78K0/KC2-C without an EV_{SS} pin, connect them to V_{SS}. With 78K0/KC2-C without an EV_{DD} pin, connect them to V_{DD}.

Remark P31 and P32 of the product can be used as on-chip debug mode setting pins (OCD1A and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

2.2.5 P40 to P43 (port 4)

P40 to P43 function as an I/O port. These pins also function as pins for serial data I/O for CEC, real-time counter correction clock output, real-time counter clock output, and serial interface data I/O.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P40/CECOUT/RTC1HZ	√	√
P41/CECIN/RTCDIV/RTCCL	√	√
P42/TxD61	–	√
P43/RxD61	–	√

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P40 to P43 function as an I/O port. P40 to P43 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).

(2) Control mode

P40 to P43 function as serial data I/O for CEC, real-time counter correction clock output, real-time counter clock output, and serial interface data I/O.

(a) CECIN

This is a serial data input pin for CEC

(b) CECOUT

This is a serial data output pin for CEC

(c) RTC1HZ

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

(d) RTCDIV

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

(e) RTCCL

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

(f) RxD61

This is a serial data input pin of serial interface UART61.

(g) TxD61

This is a serial data output pin of serial interface UART61.

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

2.2.6 P50 to P53 (port 5)

P50 to P53 function as an I/O port. These pins also function as pins for timer I/O.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P50/TI002	–	√
P51/TI012/TO02	–	√
P52	–	√
P53	–	√

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P50 to P53 function as an I/O port. P50 to P53 can be set to input or output port in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

(2) Control mode

P50 and P51 function as timer I/O.

(a) TI002

This is a pin for inputting an external count clock to 16-bit timer/event counter 02 and are also for inputting a capture trigger signal to the capture registers (CR002, CR012) of 16-bit timer/event counter 02.

(b) TI012

This is a pin for inputting a capture trigger signal to the capture register (CR002) of 16-bit timer/event counter 02.

(c) TO02

This is timer output pin of 16-bit timer/event counters 02.

2.2.7 P60 to P63 (port 6)

P60 to P63 function as an I/O port. These pins also function as pins for serial interface data I/O and clock I/O.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P60/SCLA0	√	
P61/SDAA0	√	
P62/SCLA1	√	
P63/SDAA1	√	

Remark √: Mounted

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

(1) Port mode

P60 to P63 function as an I/O port. P60 to P63 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

Output of P60 to P63 is N-ch open-drain output (6 V tolerance).

(2) Control mode

P60 to P67 function as serial interface data I/O and clock I/O.

(a) SDAA0, SDAA1

These are the serial data I/O pins for serial interfaces IICA00 and IICA01.

(b) SCLA0, SCLA1

These are the serial clock I/O pins for serial interfaces IICA00 and IICA01.

2.2.8 P70 to P77 (port 7)

P70 to P77 function as an I/O port. These pins also function as serial interface data I/O and clock I/O, chip select input, serial data I/O for CEC, and key interrupt input pins.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P70/SCLA2/KR0	√	√
P71/SDAA2/KR1	√	√
P72/CECIO/KR2	√	√
P73/SI11/KR3	√	√
P74/SO11/KR4	P74/SO11 Note	√
P75/$\overline{\text{SCK11}}$/KR5	P75/$\overline{\text{SCK11}}$ Note	√
P76//KR6/$\overline{\text{SSI11}}$	–	√
P77/KR7	–	√

Note The 78K0/KC2-C is not provided with the KR4 and KR5 pins.

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P70 to P77 function as an I/O port. P70 to P77 can be set to input or output port in 1-bit units using port mode register 7 (PM7). Only for P72, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

P70 to P77 function as serial interface data I/O and clock I/O, chip select input, serial data I/O for CEC, and key interrupt input pins.

(a) SDAA2

This is a serial data I/O pin for serial interface IICA02.

(b) SCLA2

This is a serial clock I/O pin for serial interface IICA02.

(c) KR0 to KR7

These are the key interrupt input pins.

(d) CECIO

This is a serial data I/O pin for CEC.

(e) SI11

This is a serial data input pin of serial interface CSI11.

(f) SO11

This is a serial data output pin of serial interface CSI11.

(g) $\overline{\text{SCK11}}$

This is a serial clock I/O pin of serial interface CSI11.

(h) $\overline{\text{SSI11}}$

This is a chip select input pin of serial interface CSI11.

2.2.9 P120 to P124 (port 12)

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

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P120/INTP0/EXLVI	√	
P121/X1/OCD0A	√	
P122/X2/EXCLK/OCD0B	√	
P123/XT1	√	
P124/XT2	√	

Remark √: Mounted

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

(1) Port mode

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

(2) Control mode

P120 to P124 function as pins 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.

(a) INTP0

This functions as an external interrupt request input (INTP0) 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.

Caution Process the P121/X1/OCD0A pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P121/X1/OCD0A
Flash memory programmer connection		Connect to V_{SS} via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.

Remark X1 and X2 can be used as on-chip debug mode setting pins (OCD0A and OCD0B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

2.2.10 P130 (port 13)

P130 functions as an output-only port.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P130	√	

Remarks 1. 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 **4.2.10 Port 13**).

2. √: Mounted

2.2.11 P140, P141 (port 14)

P140 and P141 function as an I/O port. These pins also function as external interrupt request input, clock output, buzzer output, and remote control receive data input.

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P140/PCL/INTP6/RIN	√	√
P141/BUZ	–	√

Remark √: Mounted, –: Not mounted

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

(1) Port mode

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

(2) Control mode

P140 to P145 function as external interrupt request input, clock output, buzzer output, and remote control receive data input.

(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) PCL

This is a clock output pin.

(c) BUZ

This is a buzzer output pin.

(d) RIN

This is a remote control receive data input pin.

2.2.12 AV_{REF}, AV_{SS}, V_{DD}, EV_{DD}, V_{SS}, EV_{SS}

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
AV_{REF}	√	√
AV_{SS}	√	√
V_{DD}	√	√
EV_{DD}	–	√
V_{SS}	√	√
EV_{SS}	–	√

Remark √: Mounted, –: Not mounted

(a) AV_{REF}

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

When the A/D converter is not used, connect this pin directly to EV_{DD} or V_{DD}[Note].

Note Make the AV_{REF} pin the same potential as the V_{DD} pin when port 2 is used as a digital port.

(b) AV_{SS}

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

(c) V_{DD} and EV_{DD}

V_{DD} is the positive power supply pin for P121 to P124 and other than ports[Note].

EV_{DD} is the positive power supply pin for ports other than P20 to P27 and P121 to P124.

Always make EV_{DD} the same potential as V_{DD}.

Note With 78K0/KC2-C that are not mounted with an EV_{DD} pin, use V_{DD} as a positive power supply pin other than P20 to P27.

(d) V_{SS} and EV_{SS}

V_{SS} is the ground potential pin for P121 to P124 and other than ports.

EV_{SS} is the ground potential pin for ports other than P20 to P27 and P121 to P124.

Always make EV_{SS} the same potential as V_{SS}.

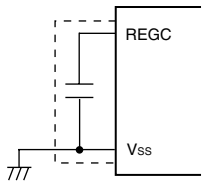
Note With 78K0/KC2-C that are not mounted with an EV_{SS} pin, use V_{SS} as a ground potential pin other than P20 to P27.

2.2.13 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

2.2.14 REGC

This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF: recommended).



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

2.2.15 FLMD0

This is a pin for setting flash memory programming mode.

Connect FLMD0 to EV_{SS} or V_{SS} in the normal operation mode.

In flash memory programming mode, connect this pin to the flash memory programmer.

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

2.3.1 78K0/KC2-C

Table 2-3 shows the types of pin I/O circuits and the recommended connections of unused pins.

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

Table 2-3. Pin I/O Circuit Types (78K0/KC2-C) (1/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/TI010/TO00			
P10/$\overline{SCK10}$/TxD0			
P11/SI10/RxD0			
P12/SO10	5-AG		
P13/TxD60			
P14/RxD60	5-AH		
P15/TOH0	5-AG		
P16/TOH1/INTP5	5-AH		
P17/TI50/TO50			
ANI0/P20 to ANI7/P27[Note]	11-G		< Digital input setting and analog input setting> Independently connect to AV_{REF} or AV_{SS} via a resistor. <Digital output setting> Leave open.

Note ANI0/P20 to ANI7/P27 are set in the analog input mode after release of reset.

Table 2-3. Pin I/O Circuit Types (78K0/KC2-C) (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P30/INTP1	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P31/INTP2/OCD1A[Note 1]			
P32/ROUT/INTP3/OCD1B			
P33/TI51/TO51/INTP4			
P40/CECOUT/RTC1HZ			
P41/CECIN/RTCDIV/ RTCCL			
P60/SCLA0	13-AI		Input: Independently connect to V_{DD} or V_{SS} via a resistor, or connect directly to V_{SS}. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P61/SDAA0			
P62/SCLA1			
P63/SDAA1			
P70/SCLA2/KR0			
P71/SDAA2/KR1			
P72/CECIO/KR2	13-AJ		
P73/SI11/KR3	13-AI		
P74/SO11			
P75/$\overline{SCK11}$			
P120/INTP0/EXLVI	5-AH		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Notes 1, 2]	37-A	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK/ OCD0B[Notes 2]			
P123/XT1[Note 2]			
P124/XT2[Note 2]			

Notes 1. Process the P31/INTP2/OCD1A and P121/X1/OCD0A pins as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P31/INTP2/OCD1A	P121/X1/OCD0A
Flash memory programmer connection		Connect to V_{SS} via a resistor.	
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset		
	During reset released	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.	

2. Use recommended connection above in I/O port mode (see **Figure 5-2 Format of Clock Operation Mode Select Register (OSCCTL)**) when these pins are not used.

Table 2-3. Pin I/O Circuit Types (78K0/KC2-C) (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P130	3-C	Output	Leave open.
P140/PCL/INTP6/RIN	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
AV_{REF}	–	–	Connect directly to V_{DD}[Note 1].
AV_{SS}	–	–	Connect directly to V_{SS}.
FLMD0	38-A	–	Connect to V_{SS}[Note 2].
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).
$\overline{\text{RESET}}$	2	Input	Connect directly to V_{DD} or via a resistor.

Notes 1. Make the same potential as the V_{DD} pin when port 2 is used as a digital port.

2. FLMD0 is a pin that is used to write data to the flash memory. To rewrite the data of the flash memory on-board, connect this pin to V_{SS} via a resistor (10 kΩ: recommended). The same applies when executing on-chip debugging.

2.3.2 78K0/KE2-C

Table 2-4 shows the types of pin I/O circuits and the recommended connections of unused pins.

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

Table 2-4. Pin I/O Circuit Types (78K0/KE2-C) (1/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01/TI010/TO00			
P02	5-AG		
P03			
P04			
P05/TI001	5-AH		
P06/TI011/TO01			
P10/$\overline{SCK10}$/TxD0			
P11/SI10/RxD0			
P12/SO10	5-AG		
P13/TxD60			
P14/RxD60	5-AH		
P15/TOH0	5-AG		
P16/TOH1/INTP5	5-AH		
P17/TI50/TO50			
ANI0/P20 to ANI7/P27[Note]	11-G		< Digital input setting and analog input setting> Independently connect to AV_{REF} or AV_{SS} via a resistor. <Digital output setting> Leave open.

Note ANI0/P20 to ANI7/P27 are set in the analog input mode after release of reset.

Table 2-4. Pin I/O Circuit Types (78K0/KE2-C) (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P30/INTP1	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31/INTP2/OCD1A[Note 1]			
P32/ROUT/INTP3/OCD1B			
P33/TI51/TO51/INTP4			
P40/CECOUT/RTC1HZ	5-AG		
P41/CECIN/RTCDIV/ RTCCL	5-AH		
P42/TxD61	5-AG		
P43/RxD61	5-AH		
P50/TI002			
P51/TI012/TO02			
P52	5-AG		
P53			
P60/SCLA0	13-AI		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor, or connect directly to EV_{SS}. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P61/SDAA0			
P62/SCLA1			
P63/SDAA1			
P70/SCLA2/KR0			
P71/SDAA2/KR1			
P72/CECIO/KR2	13-AJ		
P73/SI11/KR3	13-AI		
P74/SO11/KR4			
P75/$\overline{SCK11}$/KR5			
P76/KR6/$\overline{SSI11}$			
P77/KR7			
P120/INTP0/EXLVI	5-AH		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Notes 1, 2]	37-A	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK/ OCD0B[Notes 2]			
P123/XT1[Note 2]			
P124/XT2[Note 2]			

Notes 1. Process the P31/INTP2/OCD1A and P121/X1/OCD0A pins as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P31/INTP2/OCD1A	P121/X1/OCD0A
Flash memory programmer connection		Connect to EV_{SS} via a resistor.	Connect to V_{SS} via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset		
	During reset released	Input: Connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.	Input: Connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.

2. Use recommended connection above in I/O port mode (see **Figure 5-2 Format of Clock Operation Mode Select Register (OSCCTL)**) when these pins are not used.

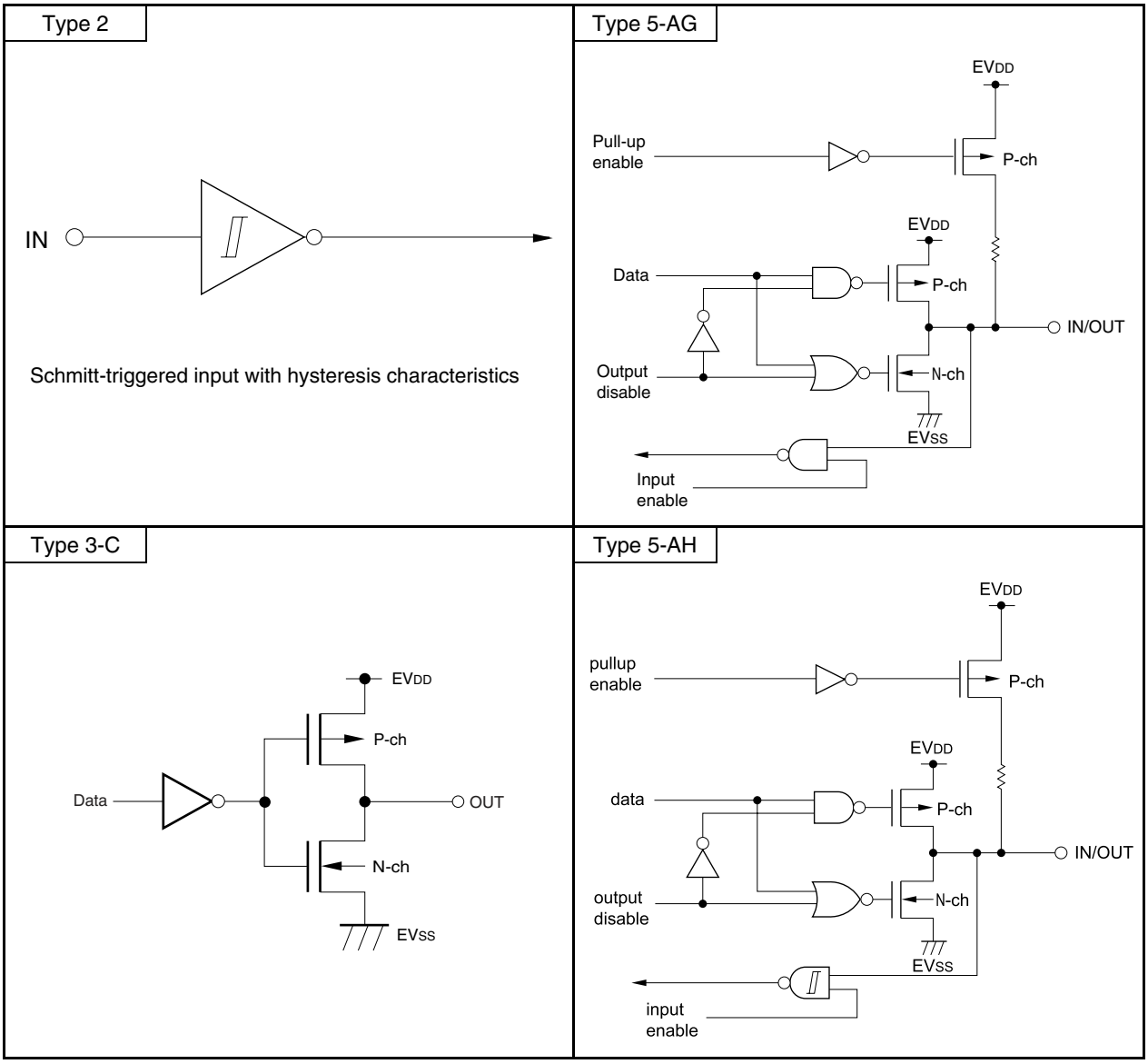
Table 2-4. Pin I/O Circuit Types (78K0/KE2-C) (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P130	3-C	Output	Leave open.
P140/PCL/INTP6/RIN	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P141/BUZ	5-AG		
AV_{REF}	–	–	Connect directly to EV_{DD} or V_{DD}[Note 1].
AV_{SS}	–	–	Connect directly to EV_{SS} or V_{SS}.
FLMD0	38-A	–	Connect to EV_{SS} or V_{SS}[Note 2].
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).
$\overline{\text{RESET}}$	2	Input	Connect directly to V_{DD} or via a resistor.

Notes 1. Make the same potential as the V_{DD} pin when port 2 is used as a digital port.

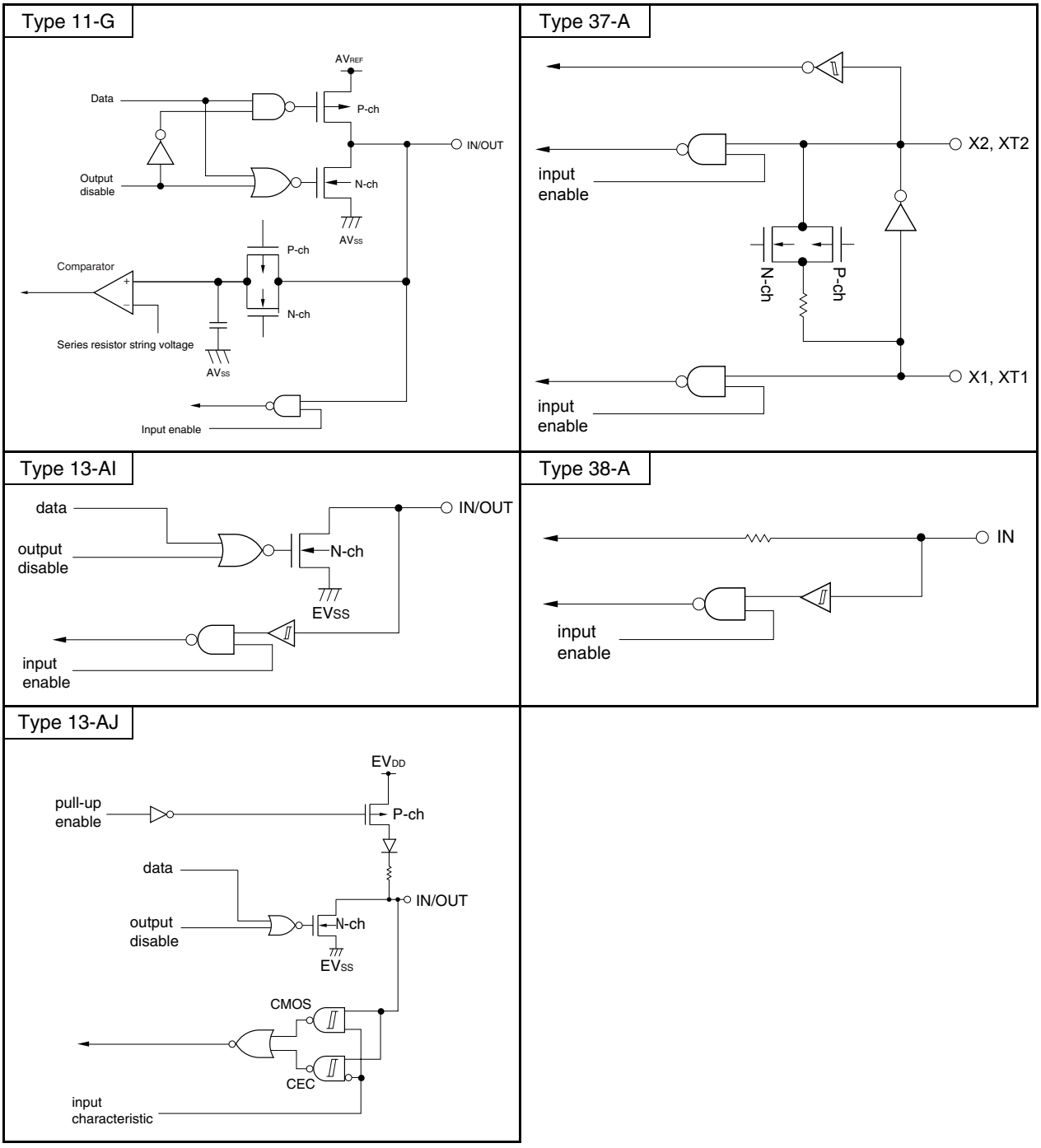
2. FLMD0 is a pin that is used to write data to the flash memory. To rewrite the data of the flash memory on-board, connect this pin to EV_{SS} or V_{SS} via a resistor (10 kΩ: recommended). The same applies when executing on-chip debugging.

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



Remark With 78K0/KC2-C not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

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



Remark With 78K0/KC2-C not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Each products in the 78K0/Kx2-C can access a 64 KB memory space. Figures 3-1 to 3-3 show the memory maps.

Cautions 1. Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products in the 78K0/Kx2-C microcontrollers are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

2. To set the memory size, set IMS and then IXS. Set the memory size so that the internal ROM and internal expansion RAM areas do not overlap.

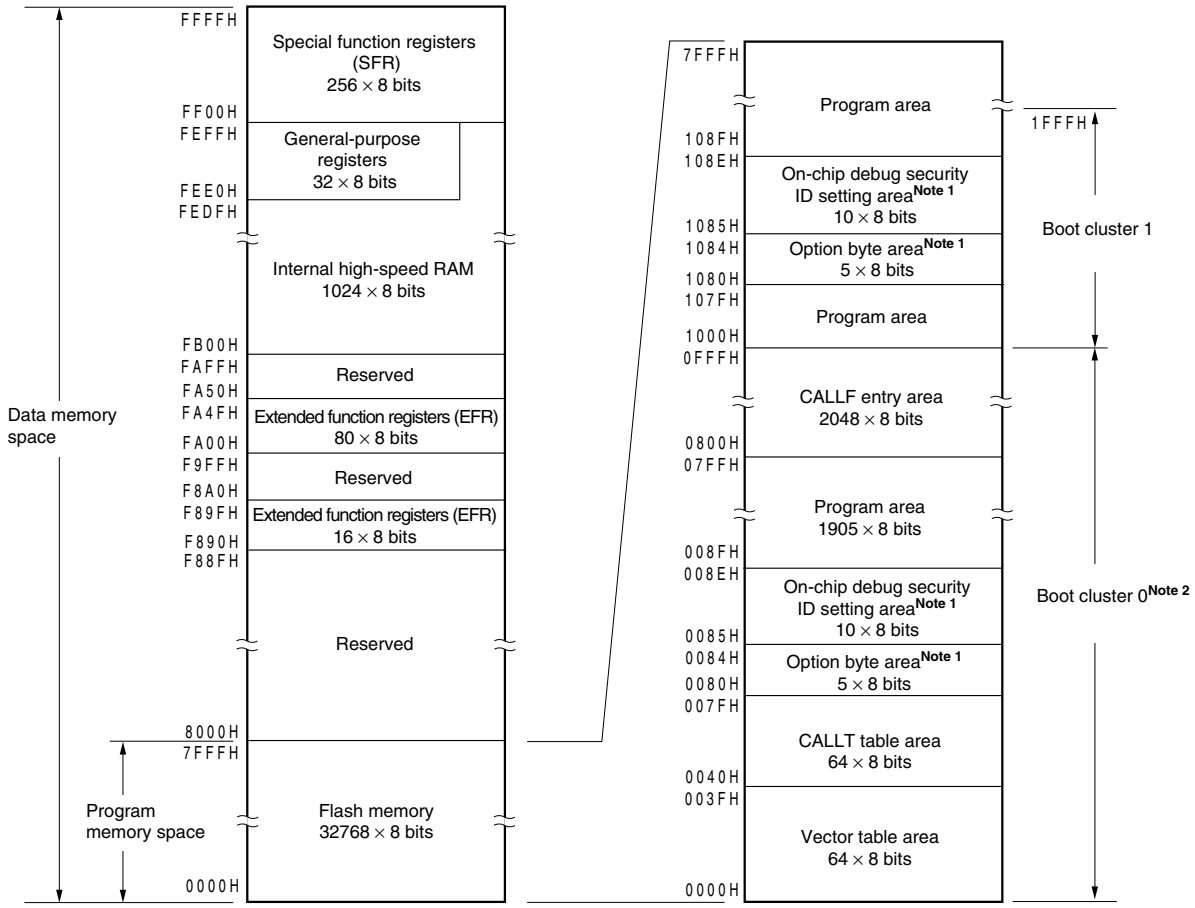
Table 3-1. Set Values of Internal Memory Size Switching Register (IMS) and Internal Expansion RAM Size Switching Register (IXS)

78K0/KC2-C	78K0/KE2-C	IMS	IXS	ROM Capacity	Internal High-Speed RAM Capacity	Internal Expansion RAM Capacity
μPD78F0760 Note 1	μPD78F0763 Note 1	C8H	0CH	32 KB	1 KB	–
μPD78F0761 Note 2	μPD78F0764 Note 2	CCH	0AH	48 KB		1 KB
μPD78F0762 Note 2	μPD78F0765 Note 2	CFH		60 KB		2 KB

Notes 1. A product that does not have an internal expansion RAM is not provided with IXS.

2. The ROM and RAM capacities of the products with the on-chip debug function can be debugged according to the debug target products. Set IMS and IXS according to the debug target products.

Figure 3-1. Memory Map (μPD78F0760, 78F0763)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **27.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

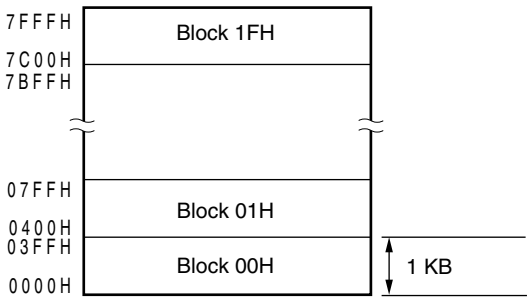
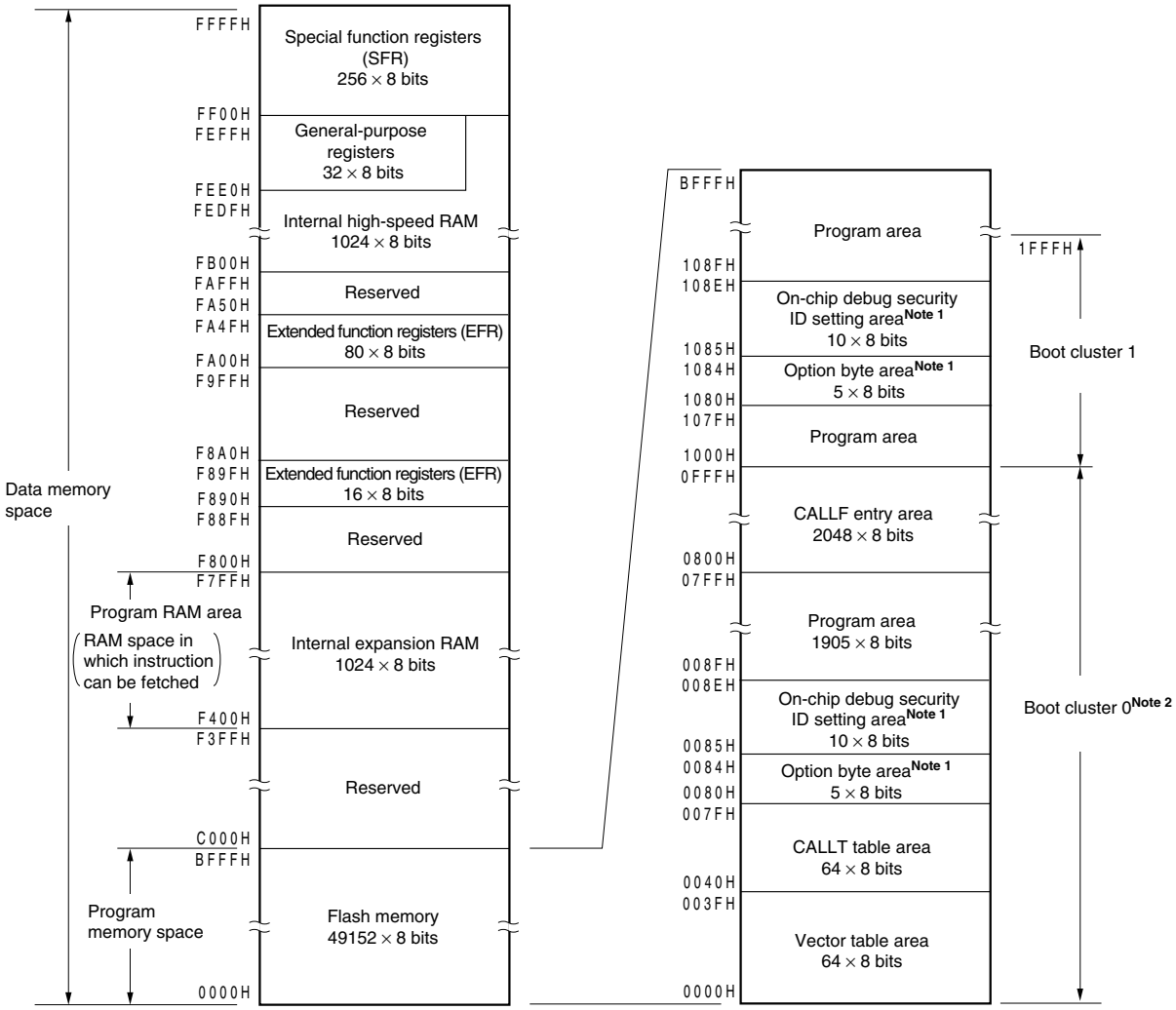


Figure 3-2. Memory Map (μPD78F0761, 78F0764)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **27.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

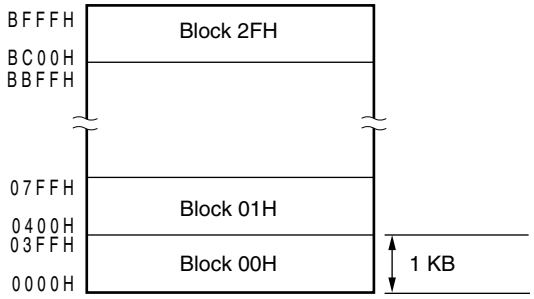
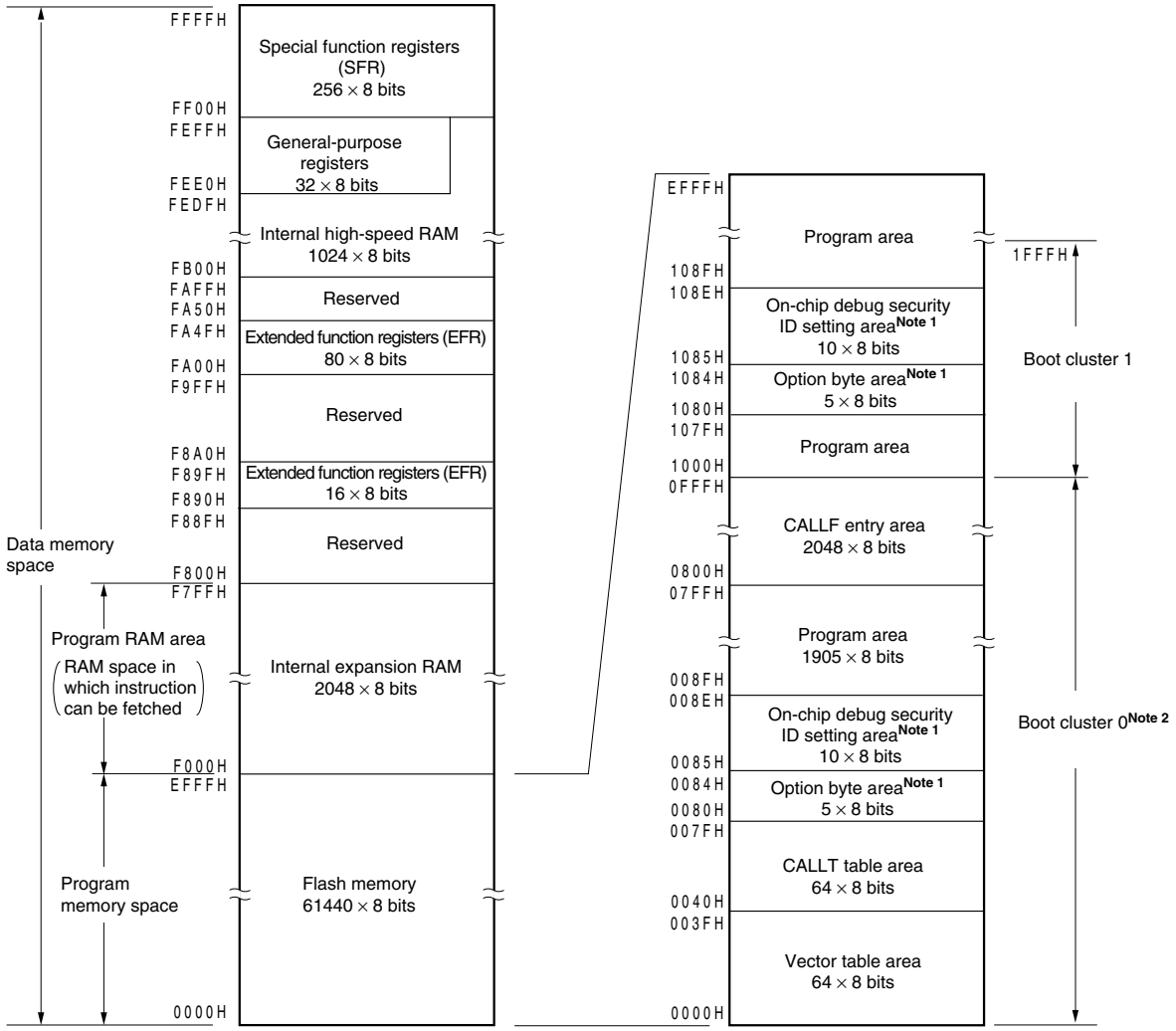


Figure 3-3. Memory Map (μPD78F0762, 78F0765)

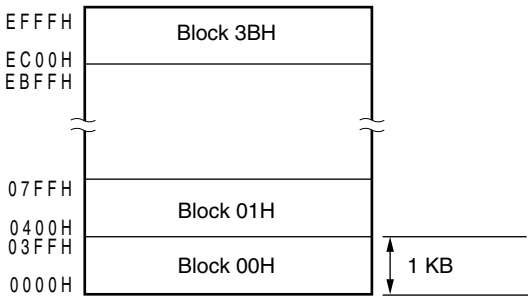


Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **27.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 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 3-2. 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
0000H to 03FFH	00H	4000H to 43FFH	10H	8000H to 83FFH	20H	C000H to C3FFH	30H
0400H to 07FFH	01H	4400H to 47FFH	11H	8400H to 87FFH	21H	C400H to C7FFH	31H
0800H to 0BFFH	02H	4800H to 4BFFH	12H	8800H to 8BFFH	22H	C800H to CBFFH	32H
0C00H to 0FFFH	03H	4C00H to 4FFFH	13H	8C00H to 8FFFH	23H	CC00H to CFFFH	33H
1000H to 13FFH	04H	5000H to 53FFH	14H	9000H to 93FFH	24H	D000H to D3FFH	34H
1400H to 17FFH	05H	5400H to 57FFH	15H	9400H to 97FFH	25H	D400H to D7FFH	35H
1800H to 1BFFH	06H	5800H to 5BFFH	16H	9800H to 9BFFH	26H	D800H to DBFFH	36H
1C00H to 1FFFH	07H	5C00H to 5FFFH	17H	9C00H to 9FFFH	27H	DC00H to DFFFH	37H
2000H to 23FFH	08H	6000H to 63FFH	18H	A000H to A3FFH	28H	E000H to E3FFH	38H
2400H to 27FFH	09H	6400H to 67FFH	19H	A400H to A7FFH	29H	E400H to E7FFH	39H
2800H to 2BFFH	0AH	6800H to 6BFFH	1AH	A800H to ABFFH	2AH	E800H to EBFFH	3AH
2C00H to 2FFFH	0BH	6C00H to 6FFFH	1BH	AC00H to AFFFH	2BH	EC00H to EFFFH	3BH
3000H to 33FFH	0CH	7000H to 73FFH	1CH	B000H to B3FFH	2CH		
3400H to 37FFH	0DH	7400H to 77FFH	1DH	B400H to B7FFH	2DH		
3800H to 3BFFH	0EH	7800H to 7BFFH	1EH	B800H to BBFFH	2EH		
3C00H to 3FFFH	0FH	7C00H to 7FFFH	1FH	BC00H to BFFFH	2FH		

Remark μPD78F0760, 78F0763: Block numbers 00H to 1FH
μPD78F0761, 78F0764: Block numbers 00H to 2FH
μPD78F0762, 78F0765: Block numbers 00H to 3BH

3.1.1 Internal program memory space

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

78K0/Kx2-C products incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F0760, 78F0763	Flash memory	32768 × 8 bits (0000H to 7FFFH)
μPD78F0761, 78F0764		49152 × 8 bits (0000H to BFFFH)
μPD78F0762, 78F0765		61440 × 8 bits (0000H to EFFFH)

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

(1) Vector table area

The 64-byte area 0000H to 003FH 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.

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 3-4. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	0020H	INTTM000
0002H	INTCK2	0022H	INTTM010
	INTNMI	0024H	INTAD
0004H	INTLVI	0026H	INTSR0/INTSRE0
0006H	INTP0	0028H	INTRTC/INTRTCI
0008H	INTP1	002AH	INTTM51
000AH	INTP2	002CH	INTIICA0
000CH	INTP3	002EH	INTIICA1
000EH	INTP4/INTDA	0030H	INTKR/INTIICA2
0010H	INTP5/INTCE/INTERR	0032H	INTCSI11
0012H	INTP6/INTRIN	0034H	INTTM001/INTSR61/INTSRE61
0014H	INTSR60/INTSRE60	0036H	INTTM011/INTST61
0016H	INTST60	0038H	INTTM002
0018H	INTCSI10/INTST0	003AH	INTTM012/INTDMU
001AH	INTTMH1	003CH	INTRERR/INTGP/INTREND/ INTDFULL
001CH	INTTMH0		
001EH	INTTM50	003EH	BRK

(2) CALLT instruction table area

The 64-byte area 0040H to 007FH can store the subroutine entry address of a 1-byte call instruction (CALLT).

(3) Option byte area

A 5-byte area of 0080H to 0084H and 1080H to 1084H can be used as an option byte area. Set the option byte at 0080H to 0084H when the boot swap is not used, and at 0080H to 0084H and 1080H to 1084H when the boot swap is used. For details, see **CHAPTER 26 OPTION BYTE**.

(4) CALLF instruction entry area

The area 0800H to 0FFFH can perform a direct subroutine call with a 2-byte call instruction (CALLF).

(5) On-chip debug security ID setting area

A 10-byte area of 0085H to 008EH and 1085H to 108EH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 0085H to 008EH when the boot swap is not used and at 0085H to 008EH and 1085H to 108EH when the boot swap is used. For details, see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

3.1.2 Internal data memory space

78K0/Kx2-C products incorporate the following RAMs.

(1) Internal high-speed RAM

Table 3-5. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD78F0760, 78F0763	1024 × 8 bits (FB00H to FEFFH)
μPD78F0761, 78F0764	
μPD78F0762, 78F0765	

This area cannot be used as a program area in which instructions are written and executed.

The internal high-speed RAM can also be used as a stack memory.

(2) Internal expansion RAM

Table 3-6. Internal Expansion RAM Capacity

Part Number	Internal Expansion RAM
μPD78F0760, 78F0763	–
μPD78F0761, 78F0764	1024 × 8 bits (F400H to F7FFH)
μPD78F0762, 78F0765	2048 × 8 bits (F000H to F7FFH)

The internal expansion RAM can also be used as a normal data area similar to the internal high-speed RAM, as well as a program area in which instructions can be written and executed.

The internal expansion RAM cannot be used as a stack memory.

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (see **Table 3-7 Special Function Register List** in **3.2.3 Special function registers (SFRs)**).

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

3.1.4 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 78K0/Kx2-C, 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 3-4 to 3-6 show correspondence between data memory and addressing. For details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-4. Correspondence Between Data Memory and Addressing (μPD78F0760, 78F0763)

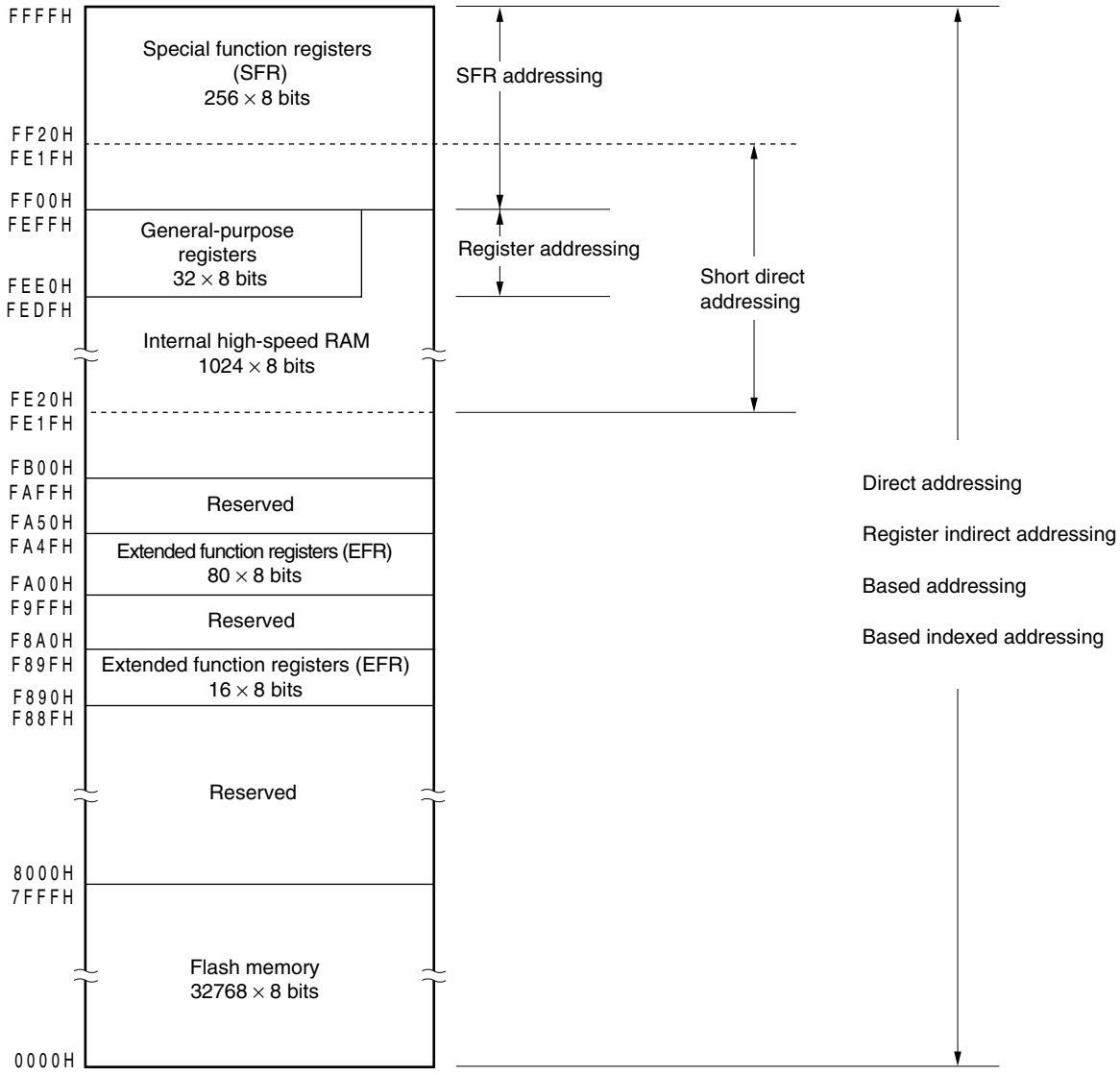


Figure 3-5. Correspondence Between Data Memory and Addressing (μPD78F0761, 78F0764)

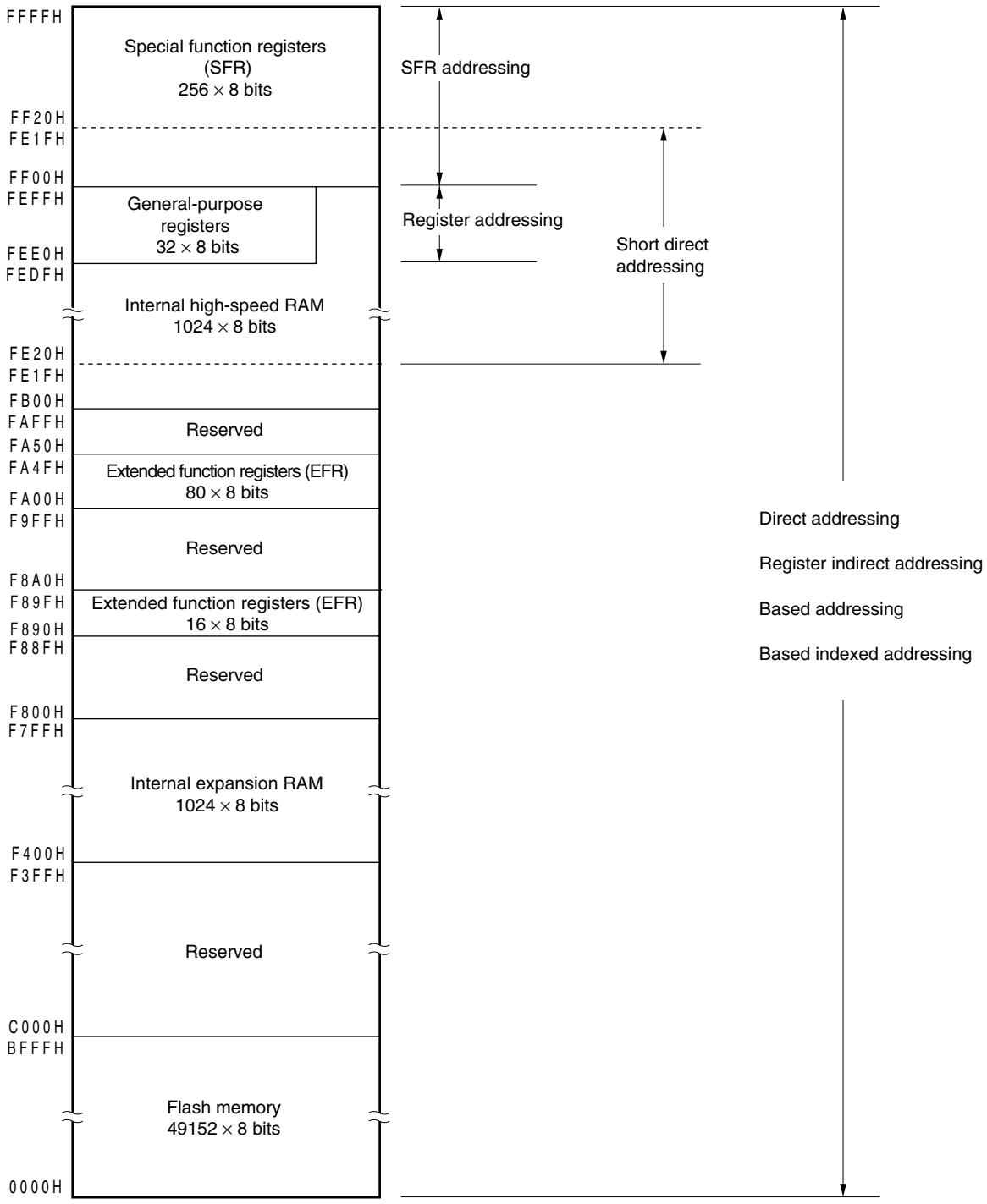
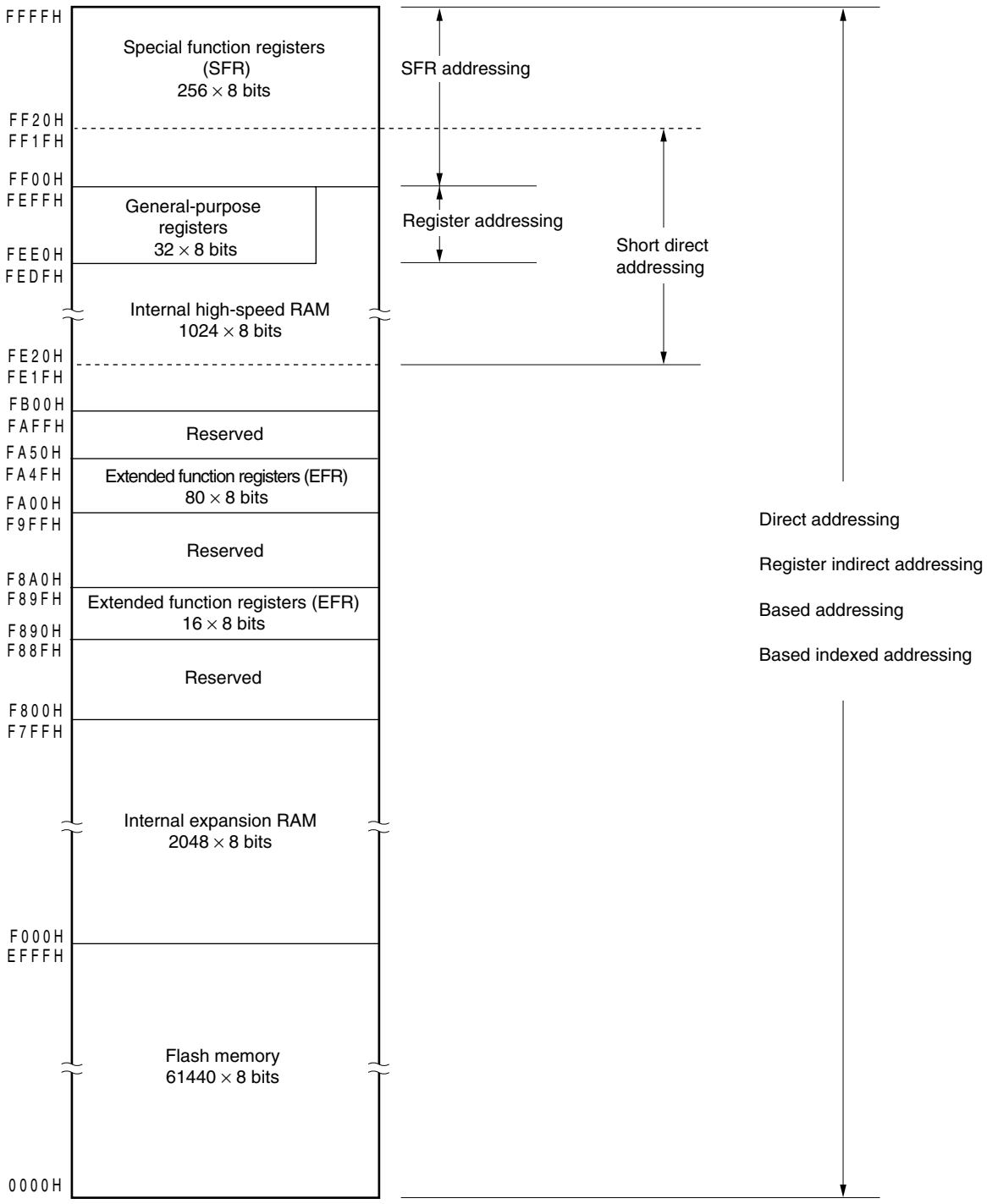


Figure 3-6. Correspondence Between Data Memory and Addressing (μPD78F0762, 78F0765)



3.2 Processor Registers

The 78K0/Kx2-C 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 16-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 3-7. Format of Program Counter

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(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 02H.

Figure 3-8. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	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 (ISP), 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 and 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 flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L, PR1H) (see **20.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

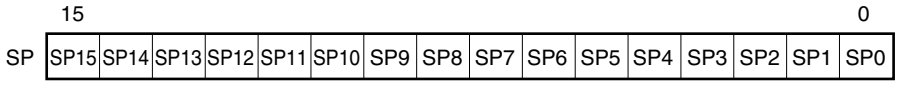
(f) Carry flag (CY)

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 high-speed RAM area can be set as the stack area.

Figure 3-9. Format of Stack Pointer



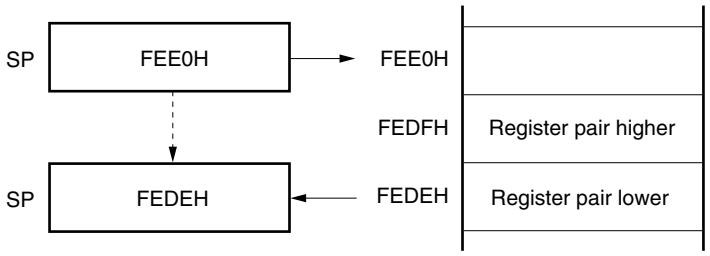
The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-10 and 3-11.

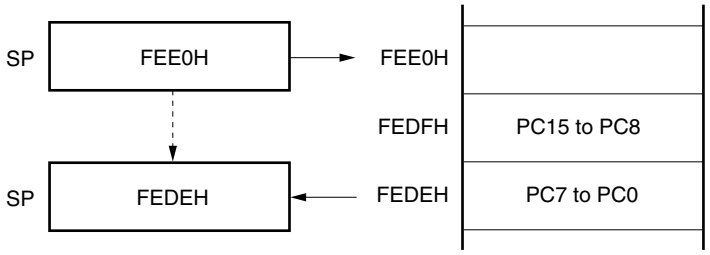
Caution Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-10. Data to Be Saved to Stack Memory

(a) PUSH rp instruction (when SP = FEE0H)



(b) CALL, CALLF, CALLT instructions (when SP = FEE0H)



(c) Interrupt, BRK instructions (when SP = FEE0H)

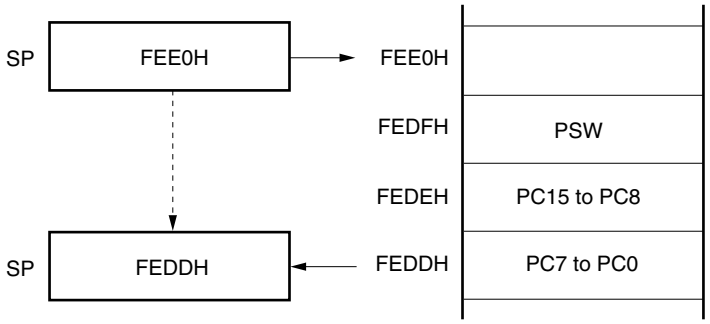
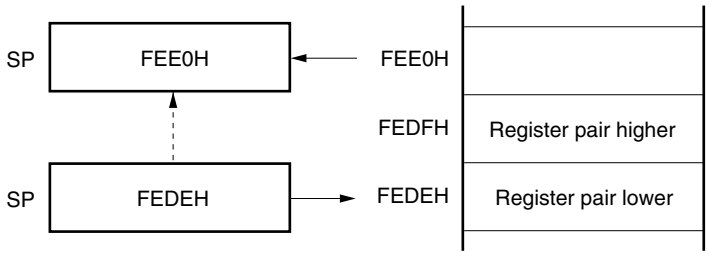
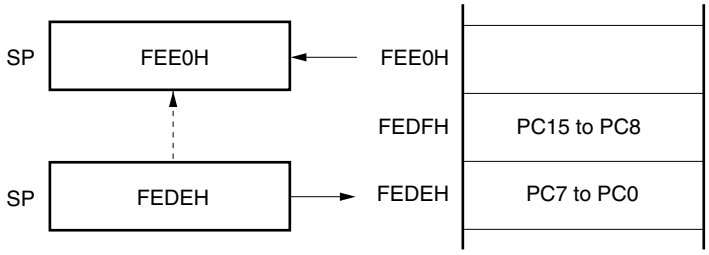


Figure 3-11. Data to Be Restored from Stack Memory

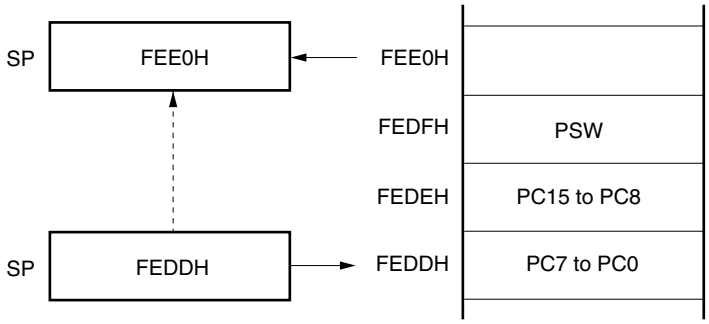
(a) POP rp instruction (when SP = FEDEH)



(b) RET instruction (when SP = FEDEH)



(c) RETI, RETB instructions (when SP = FEDDH)



3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) 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).

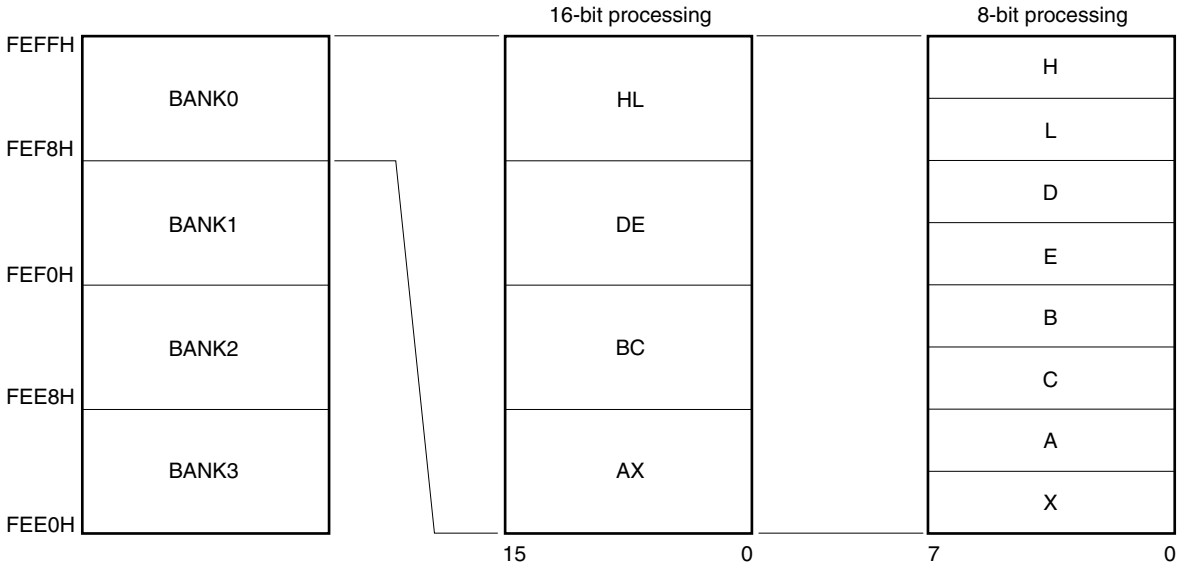
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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).

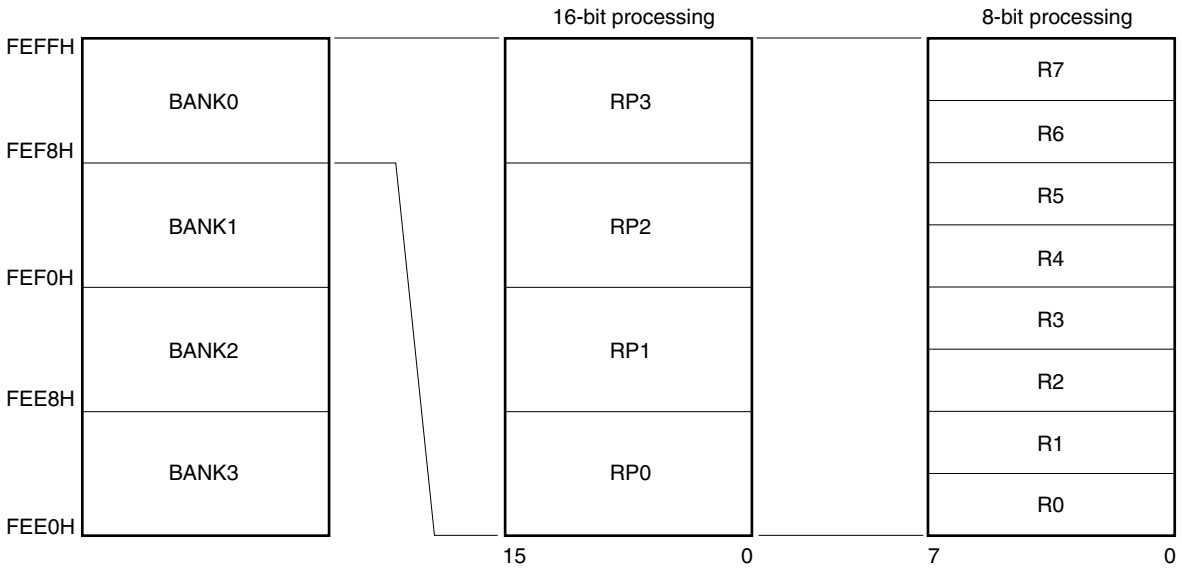
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-12. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 Special function registers (SFRs)

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

SFRs are allocated to the FF00H to FFFFH areas in the CPU.

Special function registers can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulatable bit units, 1, 8, and 16, depend on the special function register 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 3-7 gives a list of the special function registers. 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 RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0. When using the RA78K0, ID78K0-QB, and SM+, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Table 3-7. Special Function Register List (1/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
				1 Bit	8 Bits	16 Bits			
FF00H	Port register 0	P0	R/W	√	√	–	00H	√	√
FF01H	Port register 1	P1	R/W	√	√	–	00H	√	√
FF02H	Port register 2	P2	R/W	√	√	–	00H	√	√
FF03H	Port register 3	P3	R/W	√	√	–	00H	√	√
FF04H	Port register 4	P4	R/W	√	√	–	00H	√	√
FF05H	Port register 5	P5	R/W	√	√	–	00H	–	√
FF06H	Port register 6	P6	R/W	√	√	–	00H	√	√
FF07H	Port register 7	P7	R/W	√	√	–	00H	√	√
FF08H	10-bit A/D conversion result register	ADCR	R	–	–	√	0000H	√	√
FF09H	8-bit A/D conversion result register H	ADCRH	R	–	√	–	00H	√	√
FF0AH	Receive buffer register 60	RXB60	R	–	√	–	FFH	√	√
FF0BH	Transmit buffer register 60	TXB60	R/W	–	√	–	FFH	√	√
FF0CH	Port register 12	P12	R/W	√	√	–	00H	√	√
FF0DH	Port register 13	P13	R/W	√	√	–	00H	√	√
FF0EH	Port register 14	P14	R/W	√	√	–	00H	√	√
FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H	√	√
FF10H	16-bit timer counter 00	TM00	R	–	–	√	0000H	√	√
FF11H									
FF12H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√	0000H	√	√
FF13H									
FF14H	16-bit timer capture/compare register 010	CR010	R/W	–	–	√	0000H	√	√
FF15H									
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H	√	√
FF17H	8-bit timer compare register 50	CR50	R/W	–	√	–	00H	√	√
FF18H	8-bit timer H compare register 00	CMP00	R/W	–	√	–	00H	√	√
FF19H	8-bit timer H compare register 10	CMP10	R/W	–	√	–	00H	√	√
FF1AH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	00H	√	√
FF1BH	8-bit timer H compare register 11	CMP11	R/W	–	√	–	00H	√	√
FF1FH	8-bit timer counter 51	TM51	R	–	√	–	00H	√	√
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH	√	√
FF21H	Port mode register 1	PM1	R/W	√	√	–	FFH	√	√
FF22H	Port mode register 2	PM2	R/W	√	√	–	FFH	√	√
FF23H	Port mode register 3	PM3	R/W	√	√	–	FFH	√	√
FF24H	Port mode register 4	PM4	R/W	√	√	–	FFH	√	√
FF25H	Port mode register 5	PM5	R/W	√	√	–	FFH	–	√
FF26H	Port mode register 6	PM6	R/W	√	√	–	FFH	√	√
FF27H	Port mode register 7	PM7	R/W	√	√	–	FFH	√	√
FF28H	A/D converter mode register	ADM	R/W	√	√	–	00H	√	√
FF29H	Analog input channel specification register	ADS	R/W	√	√	–	00H	√	√
FF2AH	CEC local address setting register	CADR	R/W	–	–	√	0000H	√	√
FF2BH									
FF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH	√	√
FF2DH	Port input mode register 7	PIM7	R/W	√	√	–	FFH	√	√
FF2EH	Port mode register 14	PM14	R/W	√	√	–	FFH	√	√

Table 3-7. Special Function Register List (2/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
				1 Bit	8 Bits	16 Bits			
FF2FH	A/D port configuration register	ADPC	R/W	√	√	–	00H	√	√
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H	√	√
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H	√	√
FF32H	Port function register 4	PF4	R/W	√	√	–	00H	√	√
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H	√	√
FF34H	Pull-up resistor option register 4	PU4	R/W	√	√	–	00H	√	√
FF35H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H	–	√
FF36H	Port function register 7	PF7	R/W	√	√	–	00H	√	√
FF37H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H	√	√
FF38H FF39H	16-bit timer capture/compare register 001	CR001	R/W	–	–	√	0000H	√	√
FF3AH FF3BH	16-bit timer capture/compare register 011	CR011	R/W	–	–	√	0000H	√	√
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H	√	√
FF3DH	IICA shift register 02	IICA02	R/W	–	√	–	00H	√	√
FF3EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H	√	√
FF3FH	Remote controller receive control register	RMCN	R/W	√	√	–	00H	√	√
FF40H	Clock output selection register	CKS	R/W	√	√	–	00H	√	√
FF41H	8-bit timer compare register 51	CR51	R/W	–	√	–	00H	√	√
FF42H	CEC control register 0	CECCTL0	R/W	√	√	–	00H	√	√
FF43H	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H	√	√
FF44H FF45H	16-bit timer counter 01	TM01	R	–	–	√	0000H	√	√
FF47H	CEC control register 1	CECCTL1	R/W	√	√	–	00H	√	√
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H	√	√
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H	√	√
FF4AH	Serial I/O shift register 11	SIO11	R	–	√	–	00H	√	√
FF4BH	Real-time counter clock selection register	RTCCL	R/W	√	√	–	00H	√	√
FF4CH	Transmit buffer register 11	SOTB11	R/W	–	√	–	00H	√	√
FF4DH	Watch error correction register	SUBCUD	R/W	√	√	–	00H	√	√
FF4EH	Remote controller shift register receive counter register	RMSCR	R	–	√	–	00H	√	√
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H	√	√
FF50H	Asynchronous serial interface operation mode register 60	ASIM60	R/W	√	√	–	01H	√	√
FF51H	CEC transmission buffer register	CTXD	R/W	–	√	–	00H	√	√
FF52H	CEC reception buffer register	CRXD	R	–	√	–	00H	√	√
FF53H	Asynchronous serial interface reception error status register 60	ASIS60	R	–	√	–	00H	√	√

Table 3-7. Special Function Register List (3/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
					1 Bit	8 Bits	16 Bits			
FF54H	Remote controller receive data register	RMDR		R	–	√	–	00H	√	√
FF55H	Asynchronous serial interface transmission status register 60	ASIF60		R	–	√	–	00H	√	√
FF56H	Clock selection register 60	CKSR60		R/W	–	√	–	00H	√	√
FF57H	Baud rate generator control register 60	BRGC60		R/W	–	√	–	FFH	√	√
FF58H	Asynchronous serial interface control register 60	ASICL60		R/W	√	√	–	16H	√	√
FF59H	Alarm minute register	ALARMWM		R/W	–	√	–	00H	√	√
FF5AH	Alarm hour register	ALARMWH		R/W	–	√	–	12H	√	√
FF5BH	Alarm week register	ALARMWW		R/W	–	√	–	00H	√	√
FF5CH	Capture/compare control register 002	CR002		R/W	–	–	√	0000H	√	√
FF5DH										
FF5EH	Capture/compare control register 012	CR012		R/W	–	–	√	0000H	√	√
FF5FH										
FF60H	Remainder data register 0	SDR0	SDR0L	R	–	√	√	00H	√	√
FF61H			SDR0H		–	√		00H	√	√
FF62H	Multiplication/division data register A0	MDA0L	MDA0LL	R/W	–	√	√	00H	√	√
FF63H			MDA0LH		–	√		00H	√	√
FF64H		MDA0H	MDA0HL	R/W	–	√	√	00H	√	√
FF65H			MDA0HH		–	√		00H	√	√
FF66H	Multiplication/division data register B0	MDB0	MDB0L	R/W	–	√	√	00H	√	√
FF67H			MDB0H		–	√		00H	√	√
FF68H	Multiplier/divider control register 0	DMUC0		R/W	√	√	–	00H	√	√
FF69H	8-bit timer H mode register 0	TMHMD0		R/W	√	√	–	00H	√	√
FF6AH	Timer clock selection register 50	TCL50		R/W	√	√	–	00H	√	√
FF6BH	8-bit timer mode control register 50	TMC50		R/W	√	√	–	00H	√	√
FF6CH	8-bit timer H mode register 1	TMHMD1		R/W	√	√	–	00H	√	√
FF6DH	8-bit timer H carrier control register 1	TMCYC1		R/W	√	√	–	00H	√	√
FF6EH	Key return mode register	KRM		R/W	√	√	–	00H	√	√
FF6FH	Remote controller receive shift register	RMSR		R	–	√	–	00H	√	√
FF70H	Asynchronous serial interface operation mode register 0	ASIM0		R/W	√	√	–	01H	√	√
FF71H	Baud rate generator control register 0	BRGC0		R/W	–	√	–	1FH	√	√
FF72H	Receive buffer register 0	RXB0		R	–	√	–	FFH	√	√
FF73H	Asynchronous serial interface reception error status register 0	ASIS0		R	–	√	–	00H	√	√
FF74H	Transmit shift register 0	TXS0		W	–	√	–	FFH	√	√

Table 3-7. Special Function Register List (4/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
				1 Bit	8 Bits	16 Bits			
FF7AH	IICA control register 02	IICACTL02	R/W	√	√	–	00H	√	√
FF7BH	IICA control register 12	IICACTL12	R/W	√	√	–	00H	√	√
FF7CH	IICA flag register 02	IICAF02	R/W	√	√	–	00H	√	√
FF7DH	IICA status register 02	IICAS02	R	√	√	–	00H	√	√
FF7EH	IICA low-level width setting register 02	IICAWL02	R/W	–	√	–	FFH	√	√
FF7FH	IICA high-level width setting register 02	IICAWH02	R/W	–	√	–	FFH	√	√
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H	√	√
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H	√	√
FF82H FF83H	16-bit timer counter 02	TM02	R	–	–	√	0000H	√	√
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	00H	√	√
FF85H	IICA shift register 01	IICA01	R/W	–	√	–	00H	√	√
FF86H	Slave address register 01	SVA01	R/W	–	√	–	00H	√	√
FF87H	IICA flag register 01	IICAF01	R/W	√	√	–	00H	√	√
FF88H	Serial operation mode register 11	CSIM11	R/W	√	√	–	00H	√	√
FF89H	Serial clock selection register 11	CSIC11	R/W	√	√	–	00H	√	√
FF8AH	IICA control register 01	IICACTL01	R/W	√	√	–	00H	√	√
FF8BH	IICA control register 11	IICACTL11	R/W	√	√	–	00H	√	√
FF8CH	Timer clock selection register 51	TCL51	R/W	√	√	–	00H	√	√
FF90H	Asynchronous serial interface operation mode register 61	ASIM61	R/W	√	√	–	01H	–	√
FF92H	Receive buffer register 61	RXB61	R	–	√	–	FFH	–	√
FF93H	Asynchronous serial interface reception error status register 61	ASIS61	R	–	√	–	00H	–	√
FF94H	Transmit buffer register 61	TXB61	R/W	–	√	–	FFH	–	√
FF95H	Asynchronous serial interface transmission status register 61	ASIF61	R	–	√	–	00H	–	√
FF96H	Clock selection register 61	CKSR61	R/W	–	√	–	00H	–	√
FF97H	Baud rate generator control register 61	BRGC61	R/W	–	√	–	FFH	–	√
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/ 9AH Note	√	√
FF9AH	8-bit timer mode control register 01	TMC01	R/W	√	√	–	00H	√	√
FF9BH	Prescaler mode register 01	PRM01	R/W	√	√	–	00H	√	√
FF9CH	Capture/compare control register 01	CRC01	R/W	√	√	–	00H	–	√
FF9DH	16-bit timer output control register 01	TOC01	R/W	√	√	–	00H	–	√

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

Table 3-7. Special Function Register List (5/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
				1 Bit	8 Bits	16 Bits			
FF9EH	Slave address register 02	SVA02	R/W	–	√	–	00H	√	√
FF9FH	Clock operation mode select register	OSCCTL	R/W	√	√	–	00H	√	√
FFA0H	Internal oscillation mode register	RCM	R/W	√	√	–	80H Note 1	√	√
FFA1H	Main clock mode register	MCM	R/W	√	√	–	00H	√	√
FFA2H	Main OSC control register	MOC	R/W	√	√	–	80H	√	√
FFA3H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H	√	√
FFA4H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	05H	√	√
FFA5H	CEC communication error status register	CECES	R	–	√	–	00H	√	√
FFA6H	CEC communication status register	CECS	R	–	√	–	00H	√	√
FFA7H	CEC communication error flag clear trigger register	CECFC	R/W	√	√	–	00H	√	√
FFA8H	16-bit timer mode control register 02	TMC02	R/W	√	√	–	00H	√	√
FFA9H	Prescaler mode register 02	PRM02	R/W	√	√	–	00H	√	√
FFAAH	Capture/compare control register 02	CRC02	R/W	√	√	–	00H	–	√
FFABH	16-bit timer output control register 02	TOC02	R/W	√	√	–	00H	–	√
FFACH	Reset control flag register	RESF	R	–	√	–	00H Note 2	√	√
FFADH	Real-time counter control register 0	RTCC0	R/W	√	√	–	00H	√	√
FFAEH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H	√	√
FFAFH	Real-time counter control register 2	RTCC2	R/W	√	√	–	00H	√	√
FFB0H	IICA shift register 00	IICA00	R/W	–	√	–	00H	√	√
FFB1H	Slave address register 00	SVA00	R/W	–	√	–	00H	√	√
FFB2H	IICA control register 00	IICACTL00	R/W	√	√	–	00H	√	√
FFB3H	IICA control register 10	IICACTL10	R/W	√	√	–	00H	√	√
FFB4H	IICA flag register 00	IICAF00	R/W	√	√	–	00H	√	√
FFB5H	Asynchronous serial interface control register 61	ASICL61	R/W	√	√	–	16H	–	√
FFB6H	IICA status register 00	IICAS00	R	√	√	–	00H	√	√
FFB7H	Remote controller receive data output control register	RMSW	R/W	√	√	–	00H	√	√
FFB8H	IICA low-level width setting register 00	IICAWL00	R/W	–	√	–	FFH	√	√
FFB9H	IICA high-level width setting register 00	IICAWH00	R/W	–	√	–	FFH	√	√
FFBAH	16-bit timer mode control register 00	TMC00	R/W	√	√	–	00H	√	√
FFBBH	Prescaler mode register 00	PRM00	R/W	√	√	–	00H	√	√
FFBCH	Capture/compare control register 00	CRC00	R/W	√	√	–	00H	√	√
FFBDH	16-bit timer output control register 00	TOC00	R/W	√	√	–	00H	√	√

Notes 1. The value of this register is 00H immediately after a reset release but automatically changes to 80H after oscillation accuracy stabilization of high-speed internal oscillator has been waited.

2. The reset value of RESF varies depending on the reset source.

Table 3-7. Special Function Register List (6/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
					1 Bit	8 Bits	16 Bits			
FFBEH	Low-voltage detection register	LVIM		R/W	√	√	–	00H Note 1	√	√
FFBFH	Low-voltage detection level selection register	LVIS		R/W	√	√	–	00H Note 1	√	√
FFC0H	–	PFCMD Note 2		–	–	–	–	Undefined	√	√
FFC2H	–	PFS Note 2		–	–	–	–	Undefined	√	√
FFC4H	–	FLPMC Note 2		–	–	–	–	Undefined	√	√
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H	√	√
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H	√	√
FFE2H	Interrupt request flag register 1L	IF1	IF1L	R/W	√	√	√	00H	√	√
FFE3H	Interrupt request flag register 1H		IF1H	R/W	√	√		00H	√	√
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH	√	√
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH	√	√
FFE6H	Interrupt mask flag register 1L	MK1	MK1L	R/W	√	√	√	FFH	√	√
FFE7H	Interrupt mask flag register 1H		MK1H	R/W	√	√		FFH	√	√
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH	√	√
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH	√	√
FFEAH	Priority specification flag register 1L	PR1	PR1L	R/W	√	√	√	FFH	√	√
FFEBH	Priority specification flag register 1H		PR1H	R/W	√	√		FFH	√	√
FFEDH	IICA status register 01	IICAS01		R	√	√	–	00H	√	√
FFEEH	IICA low-level width setting register 01	IICAWL01		R/W	–	√	–	FFH	√	√
FFEFH	IICA high-level width setting register 01	IICAWH01		R/W	–	√	–	FFH	√	√
FFF0H	Internal memory size switching register Notes 3, 4	IMS		R/W	–	√	–	CFH	√	√
FFF4H	Internal expansion RAM size switching register Notes 3, 4	IXS		R/W	–	√	–	0CH	Note 5	Note 5
FFF9H	Remote controller receive interrupt status register	INTS		R	√	√	–	00H	√	√
FFFAH	Remote controller receive interrupt status clear register	INTC		R/W	√	√	–	00H	√	√
FFFBH	Processor clock control register	PCC		R/W	√	√	–	01H	√	√

Notes 1. The reset values of LVIM and LVIS vary depending on the reset source.

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

3. Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products in the 78K0/Kx2-C are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated in Table 3-1.

4. The ROM and RAM capacities of the products with the on-chip debug function can be debugged by setting IMS and IXS, according to the debug target products. Set IMS and IXS according to the debug target products.

5. This register is incorporated only in products with internal expansion RAM.

Remark For EFRs in the EFR area, see **Table 3-8 Extended Function Register (EFR) List**.

3.2.4 Extended function registers (EFRs)

Unlike a general-purpose register, each extended function register (EFR) has a special function.

EFRs are allocated to the F890H to F89FH and FA00H to FA4FH area. EFRs other than those in the SFR area (FF00H to FFFFH) are allocated to this area.

EFRs 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 EFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Specify the address of the EFR for the HL register and include the 1-bit manipulation instruction operand ([HL].bit).
- 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 EFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an EFR. It is a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0. When using the RA78K0, ID78K0-QB, and SM+ for 78K0, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding EFR 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 EFR is not assigned.

Remark For SFRs in the SFR area, see **3.2.3 Special function registers (SFRs)**.

Table 3-8. Extended Function Register (EFR) List (1/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
				1 Bit	8 Bits	16 Bits			
F890H F891H	Sub-count register	RSUBC	R	–	–	√	0000H	√	√
F892H	Second count register	SEC	R/W	–	√	–	00H	√	√
F893H	Minute count register	MIN	R/W	–	√	–	00H	√	√
F894H	Hour count register	HOUR	R/W	–	√	–	12H	√	√
F895H	Week count register	WEEK	R/W	–	√	–	00H	√	√
F896H	Day count register	DAY	R/W	–	√	–	01H	√	√
F897H	Month count register	MONTH	R/W	–	√	–	01H	√	√
F898H	Year count register	YEAR	R/W	–	√	–	00H	√	√
FA04H FA05H	CEC transmission start bit width setting register	STATB	R/W	–	–	√	0000H	√	√
FA06H FA07H	CEC transmission start bit low width setting register	STATL	R/W	–	–	√	0000H	√	√
FA08H FA09H	CEC transmission logical 0 low width setting register	LGC0L	R/W	–	–	√	0000H	√	√
FA0AH FA0BH	CEC transmission logical 1 low width setting register	LGC1L	R/W	–	–	√	0000H	√	√
FA0CH FA0DH	CEC transmission data bit width setting register	DATB	R/W	–	–	√	0000H	√	√
FA0EH FA0FH	CEC reception data sampling time setting register	NOMT	R/W	–	–	√	0000H	√	√
FA10H FA11H	CEC reception start bit minimum low width setting register	STATLL	R/W	–	–	√	0000H	√	√
FA12H FA13H	CEC reception start bit maximum low width setting register	STATLH	R/W	–	–	√	0000H	√	√
FA14H FA15H	CEC reception start bit minimum bit width setting register	STATBL	R/W	–	–	√	0000H	√	√
FA16H FA17H	CEC reception start bit maximum bit width setting register	STATBH	R/W	–	–	√	0000H	√	√
FA18H FA19H	CEC reception logical 0 minimum low width setting register	LGC0LL	R/W	–	–	√	0000H	√	√
FA1AH FA1BH	CEC reception logical 0 maximum low width setting register	LGC0LH	R/W	–	–	√	0000H	√	√
FA1CH FA1DH	CEC reception logical 1 minimum low width setting register	LGC1LL	R/W	–	–	√	0000H	√	√
FA1EH FA1FH	CEC reception logical 1 maximum low width setting register	LGC1LH	R/W	–	–	√	0000H	√	√
FA20H FA21H	CEC reception data bit minimum bit width setting register	DATBL	R/W	–	–	√	0000H	√	√
FA22H FA23H	CEC reception data bit maximum bit width setting register	DATBH	R/W	–	–	√	0000H	√	√

Table 3-8. Extended Function Register (EFR) List (2/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	KC2-C	KE2-C
				1 Bit	8 Bits	16 Bits			
FA24H FA25H	CEC data bit reference width setting register	NOMP	R/W	–	–	√	0000H	√	√
FA44H	Remote controller receive GPLS compare register	RMGPLS	R/W	–	√	–	00H	√	√
FA45H	Remote controller receive GPLL compare register	RMGPLL	R/W	–	√	–	00H	√	√
FA46H	Remote controller receive GPHS compare register	RMGPHS	R/W	–	√	–	00H	√	√
FA47H	Remote controller receive GPHL compare register	RMGPHL	R/W	–	√	–	00H	√	√
FA48H	Remote controller receive DLS compare register	RMDLS	R/W	–	√	–	00H	√	√
FA49H	Remote controller receive DLL compare register	RMDLL	R/W	–	√	–	00H	√	√
FA4AH	Remote controller receive DH0S compare register	RMDH0S	R/W	–	√	–	00H	√	√
FA4BH	Remote controller receive DH0L compare register	RMDH0L	R/W	–	√	–	00H	√	√
FA4CH	Remote controller receive DH1S compare register	RMDH1S	R/W	–	√	–	00H	√	√
FA4DH	Remote controller receive DH1L compare register	RMDH1L	R/W	–	√	–	00H	√	√
FA4EH	Remote controller receive end width select register	RMER	R/W	–	√	–	00H	√	√

Remark For SFRs in the SFR area, see **Table 3-7 SFR List**.

3.3 Instruction Address Addressing

An instruction address is determined by contents of the program counter (PC), and is normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to PC and branched by the following addressing (for details of instructions, refer to the **78K/0 Series Instructions User's Manual (U12326E)**).

3.3.1 Relative addressing

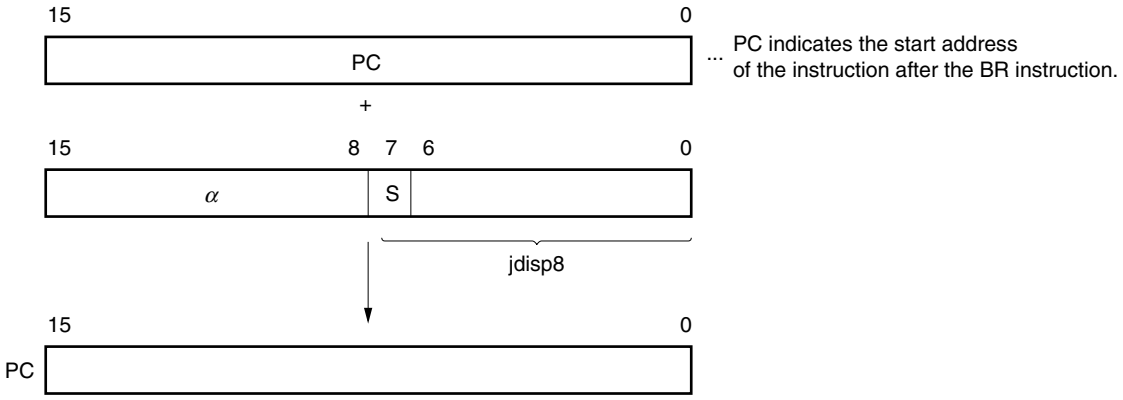
[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (−128 to +127) and bit 7 becomes a sign bit.

In other words, relative addressing consists of relative branching from the start address of the following instruction to the −128 to +127 range.

This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]



When S = 0, all bits of α are 0.
When S = 1, all bits of α are 1.

3.3.2 Immediate addressing

[Function]

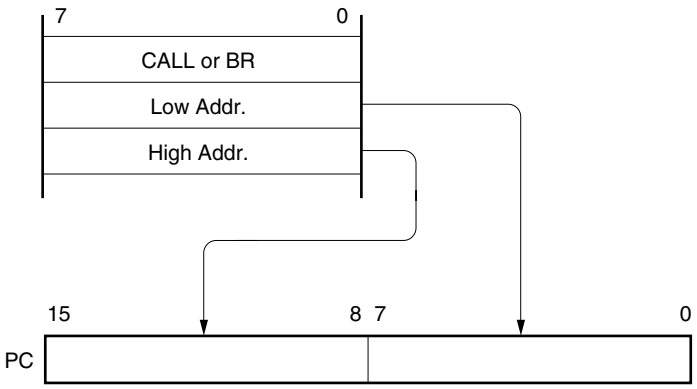
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

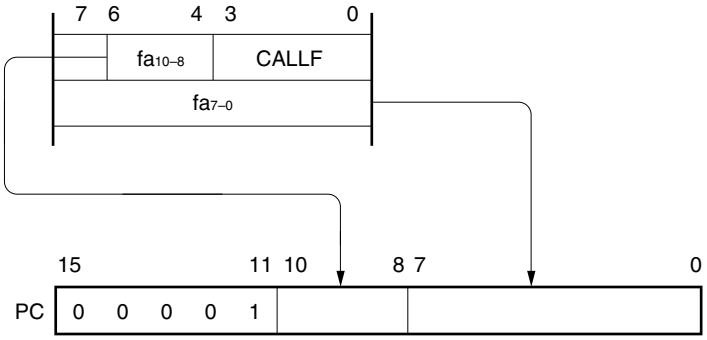
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

In the case of CALL !addr16 and BR !addr16 instructions



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

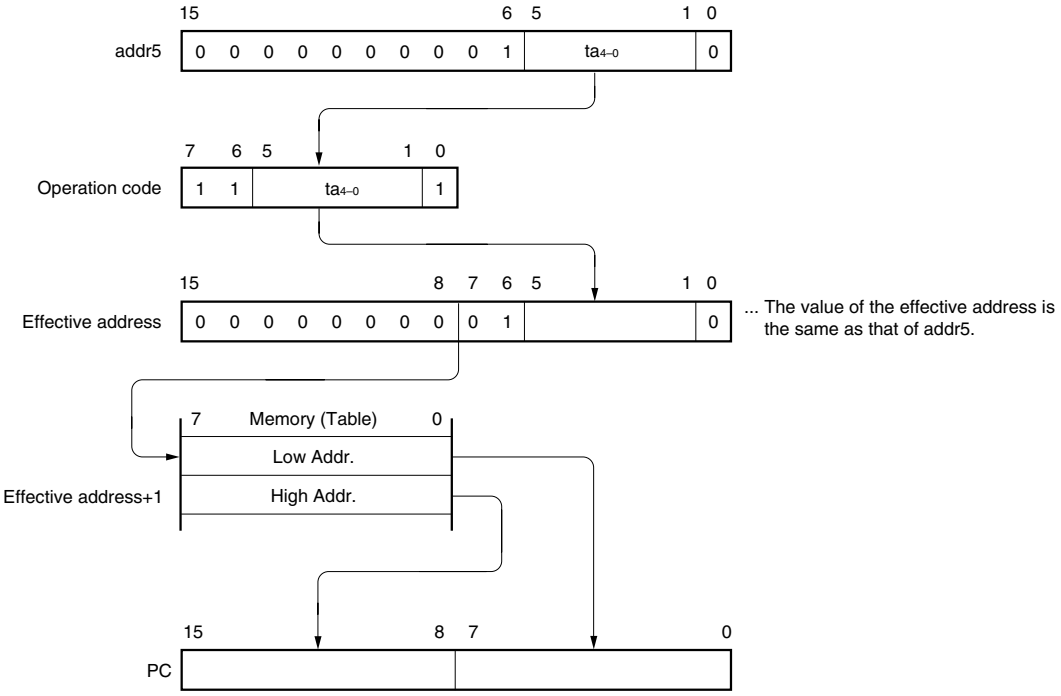
[Function]

Table contents (branch destination address) of the particular location to be addressed by bits 1 to 5 of the immediate data of an operation code are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed.

This instruction references the address stored in the memory table from 40H to 7FH, and allows branching to the entire memory space.

[Illustration]



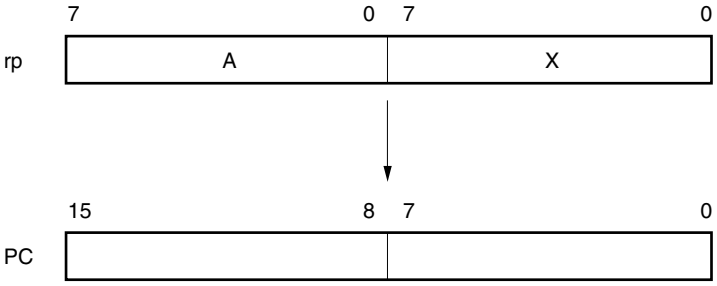
3.3.4 Register addressing

[Function]

Register pair (AX) contents to be specified with an instruction word are transferred to the program counter (PC) and branched.

This function is carried out when the BR AX instruction is executed.

[Illustration]



3.4 Operand Address Addressing

The following methods are available to specify the register and memory (addressing) to undergo manipulation during instruction execution.

3.4.1 Implied addressing

[Function]

The register that functions as an accumulator (A and AX) among the general-purpose registers is automatically (implicitly) addressed.

Of the 78K0/Kx2-C instruction words, the following instructions employ implied addressing.

Instruction	Register to Be Specified by Implied Addressing
MULU	A register for multiplicand and AX register for product storage
DIVUW	AX register for dividend and quotient storage
ADJBA/ADJBS	A register for storage of numeric values that become decimal correction targets
ROR4/ROL4	A register for storage of digit data that undergoes digit rotation

[Operand format]

Because implied addressing can be automatically determined with an instruction, no particular operand format is necessary.

[Description example]

In the case of MULU X

With an 8-bit × 8-bit multiply instruction, the product of the A register and X register is stored in AX. In this example, the A and AX registers are specified by implied addressing.

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register bank select flags (RBS0 to RBS1) and the register specify codes of an operation code.

Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

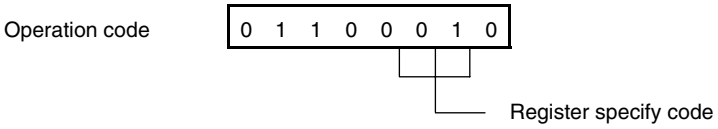
[Operand format]

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

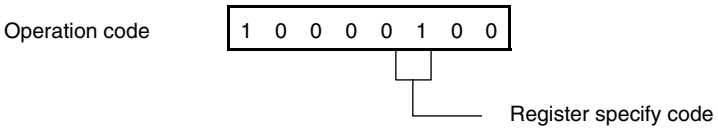
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

The memory to be manipulated is directly addressed with immediate data in an instruction word becoming an operand address.

[Operand format]

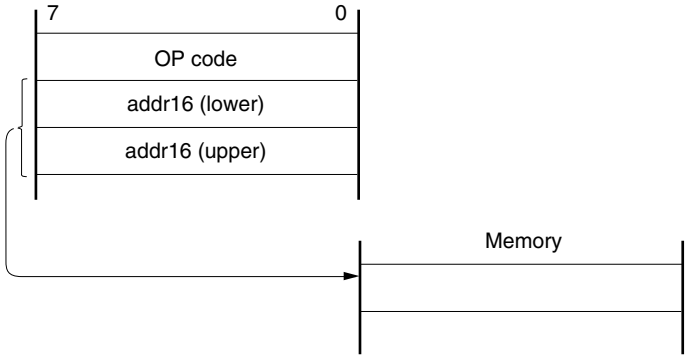
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !0FE00H; when setting !addr16 to FE00H

Operation code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

This addressing is applied to the 256-byte space FE20H to FF1FH. Internal high-speed RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to be manipulated with a small number of bytes and clocks.

When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. See the **[Illustration]** shown below.

[Operand format]

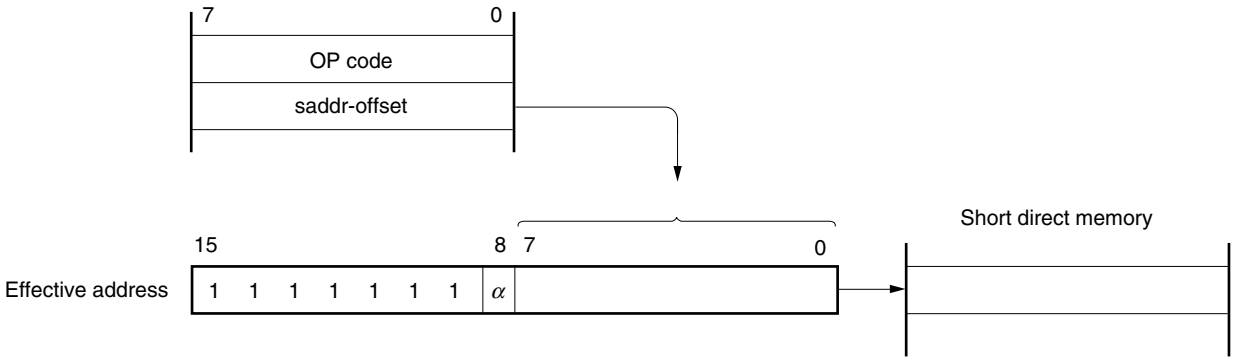
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

```
MOV 0FE30H, A      ; When transferring the value of A register to the saddr (FE30H)
```

Operation code	1 1 1 1 0 0 1 0	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)

[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$

When 8-bit immediate data is 00H to 1FH, $\alpha = 1$

3.4.5 Special function register (SFR) addressing

[Function]

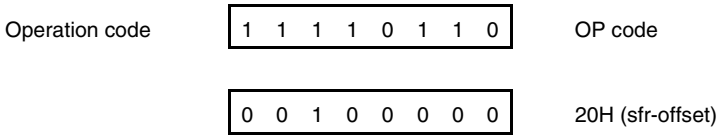
A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word. This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

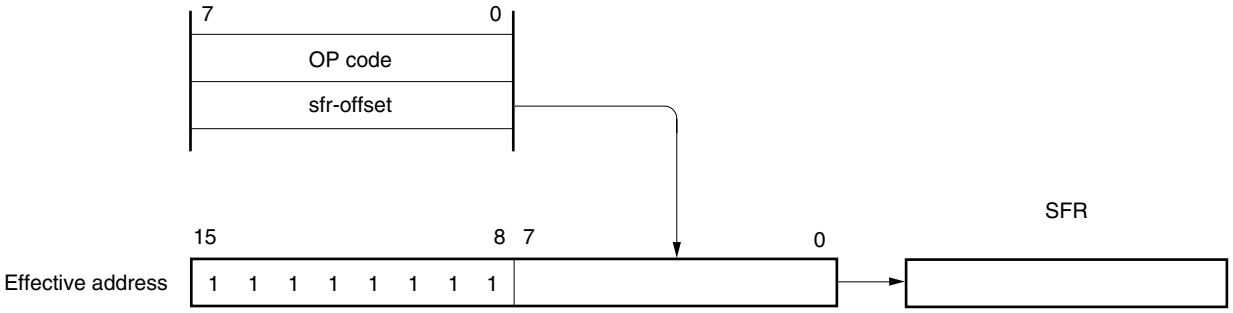
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

MOV PM0, A; when selecting PM0 (FF20H) as sfr



[Illustration]



3.4.6 Register indirect addressing

[Function]

Register pair contents specified by a register pair specify code in an instruction word and by a register bank select flag (RBS0 and RBS1) serve as an operand address for addressing the memory. This addressing can be carried out for all of the memory spaces.

[Operand format]

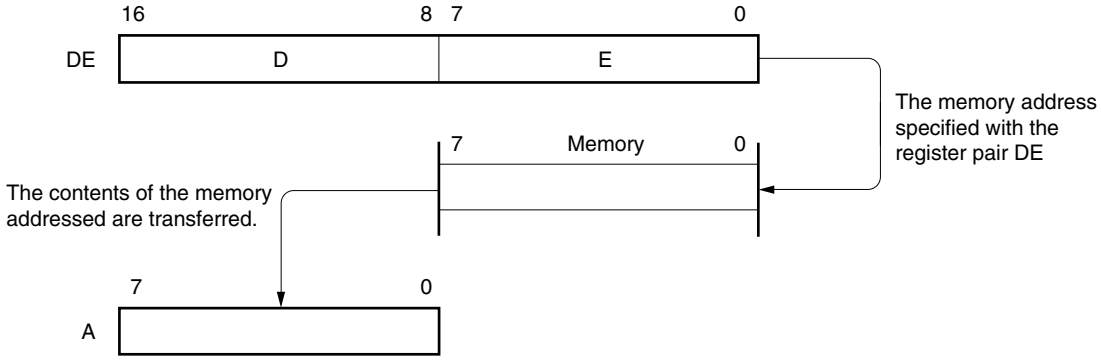
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; when selecting [DE] as register pair

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all of the memory spaces.

[Operand format]

Identifier	Description
–	[HL + byte]

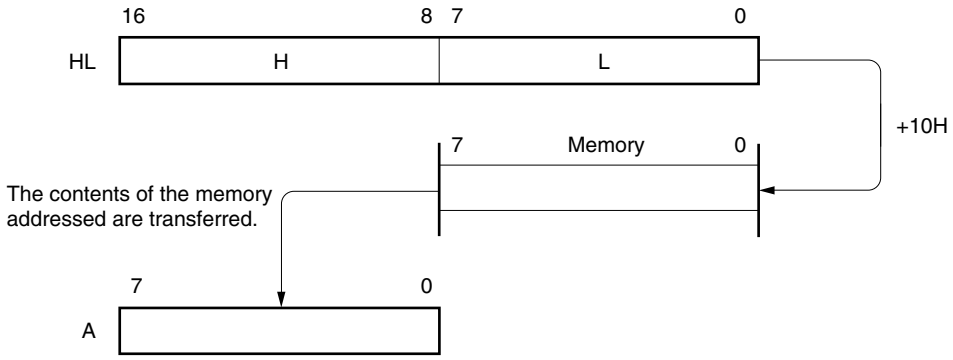
[Description example]

MOV A, [HL + 10H]; when setting byte to 10H

Operation code 1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

[Illustration]



3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction word are added to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the B or C register contents as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all of the memory spaces.

[Operand format]

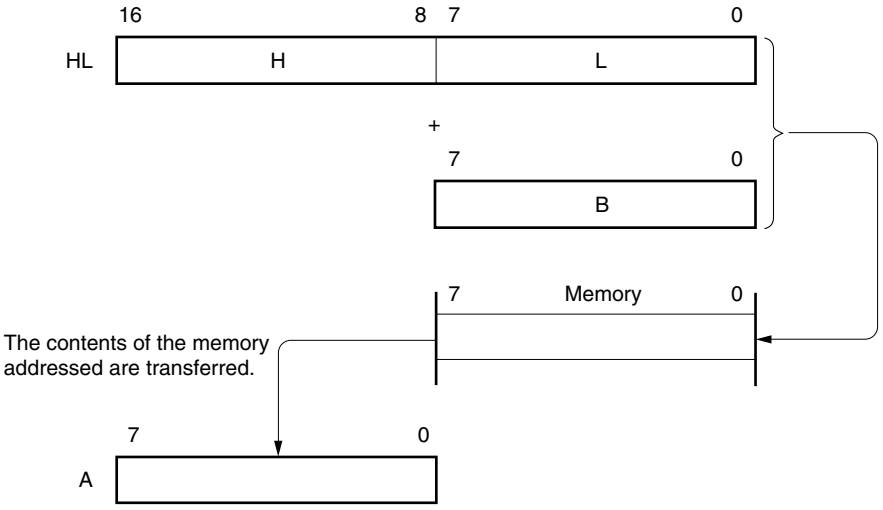
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

MOV A, [HL +B]; when selecting B register

Operation code	1	0	1	0	1	0	1	1

[Illustration]



3.4.9 Stack addressing

[Function]

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

This addressing method is automatically employed when the PUSH, POP, subroutine call and return instructions are executed or the register is saved/reset upon generation of an interrupt request.

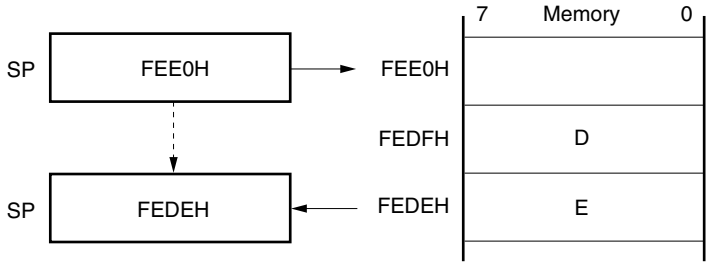
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

PUSH DE; when saving DE register

Operation code	1	0	1	1	0	1	0	1

[Illustration]



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

Pin I/O buffer power supplies depend on the product. The relationship between these power supplies and the pins is shown below.

Table 4-1. Pin I/O Buffer Power Supplies (AV_{REF}, V_{DD})

- 78K0/KC2-C: 48-pin plastic LQFP (fine pitch) (7x7)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27
V_{DD}	Pins other than P20 to P27

Table 4-2. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD}, V_{DD})

- 78K0/KE2-C: 64-pin plastic LQFP (fine pitch) (10x10)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27
EV_{DD}	Port pins other than P20 to P27 and P121 to P124
V_{DD}	• P121 to P124 • Pins other than port

78K0/Kx2-C microcontrollers are provided with digital I/O ports, which enable variety of control operations. The functions of each port are shown in Table 4-3.

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 4-3. Port Functions (1/2)

KC2-C	KE2-C	Function Name	I/O	Function	After Reset	Alternate Function
√	√	P00	I/O	Port 0. 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	TI000
√	√	P01				TI010/TO00
–	√	P02				–
–	√	P03				–
–	√	P04				–
–	√	P05				TI001
–	√	P06				TI011/TO01
√	√	P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{\text{SCK10}}$/TxD0
√	√	P11				SI10/RxD0
√	√	P12				SO10
√	√	P13				TxD60
√	√	P14				RxD60
√	√	P15				TOH0
√	√	P16				TOH1/INTP5
√	√	P17				TI50/TO50
√	√	P20 to P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0 to ANI7
√	√	P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1
√	√	P31				INTP2/OCD1A
√	√	P32				ROUT/INTP3/ OCD1B
√	√	P33				TI51/TO51/INTP4
√	√	P40	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	CECOUT/RTC1HZ
√	√	P41				CECIN/RTCDIV/ RTCCL
–	√	P42				TxD61
–	√	P43				RxD61
–	√	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	TI002
–	√	P51				TI012/TO02
–	√	P52, P53				–
√	√	P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0
√	√	P61				SDAA0
√	√	P62				SCLA1
√	√	P63				SDAA1

Remark √: Mounted, –: Not mounted

Table 4-3. Port Functions (2/2)

KC2-C	KE2-C	Function Name	I/O	Function	After Reset	Alternate Function
√	√	P70	I/O	Port 7. I/O port. Output of P70 to P77 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. Only for P72, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA2/KR0
√	√	P71				SDAA2/KR1
√	√	P72				CECIO/KR2
√	√	P73				SI11/KR3
Note	√	P74				SO11/KR4
Note	√	P75				$\overline{\text{SCK11}}$/KR5
–	√	P76				KR6/$\overline{\text{SSI11}}$
–	√	P77				KR7
√	√	P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. Only for P120, input/output can be specified. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
√	√	P121	Input			X1/OCD0A
√	√	P122				X2/EXCLK/OCD0B
√	√	P123				XT1
√	√	P124				XT2
√	√	P130	Output	Port 13. 1-bit output-only port.	Output port	–
√	√	P140	I/O	Port 14. 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	PCL/INTP6/RIN
–	√	P141				BUZ

Note The 78K0/KC2-C is not provided with the KR4 and KR5 pins.

Remark √: Mounted, –: Not mounted

4.2 Port Configuration

Ports include the following hardware.

Table 4-4. Port Configuration

Item	Configuration
Control registers	• 78K0/KC2-C Port mode register (PMxx): PM0 to PM4, PM6, PM7, PM12, PM14 Port register (Pxx): P0 to P4, P6, P7, P12 to P14 Pull-up resistor option register (PUxx): PU0, PU1, PU3, PU4, PU7, PU12, PU14 A/D port configuration register (ADPC) • 78K0/KE2-C Port mode register (PMxx): PM0 to PM7, PM12, PM14 Port register (Pxx): P0 to P7, P12 to P14 Pull-up resistor option register (PUxx): PU0, PU1, PU3 to PU5, PU7, PU12, PU14 • Common Port function register 4 (PF4) Port function register 7 (PF7) Port input mode register 7 (PIM7) A/D port configuration register (ADPC)
Port	• 78K0/KC2-C: Total: 41 (CMOS I/O: 26, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 10) • 78K0/KE2-C: Total: 55 (CMOS I/O: 38, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 12)
Pull-up resistor	• 78K0/KC2-C: Total: 19 • 78K0/KE2-C: Total: 31

4.2.1 Port 0

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P00/TI000	√	√
P01/TI010/TO00	√	√
P02	–	√
P03	–	√
P04	–	√
P05/TI001	–	√
P06/TI011/TO01	–	√

Remark √: Mounted, –: Not mounted

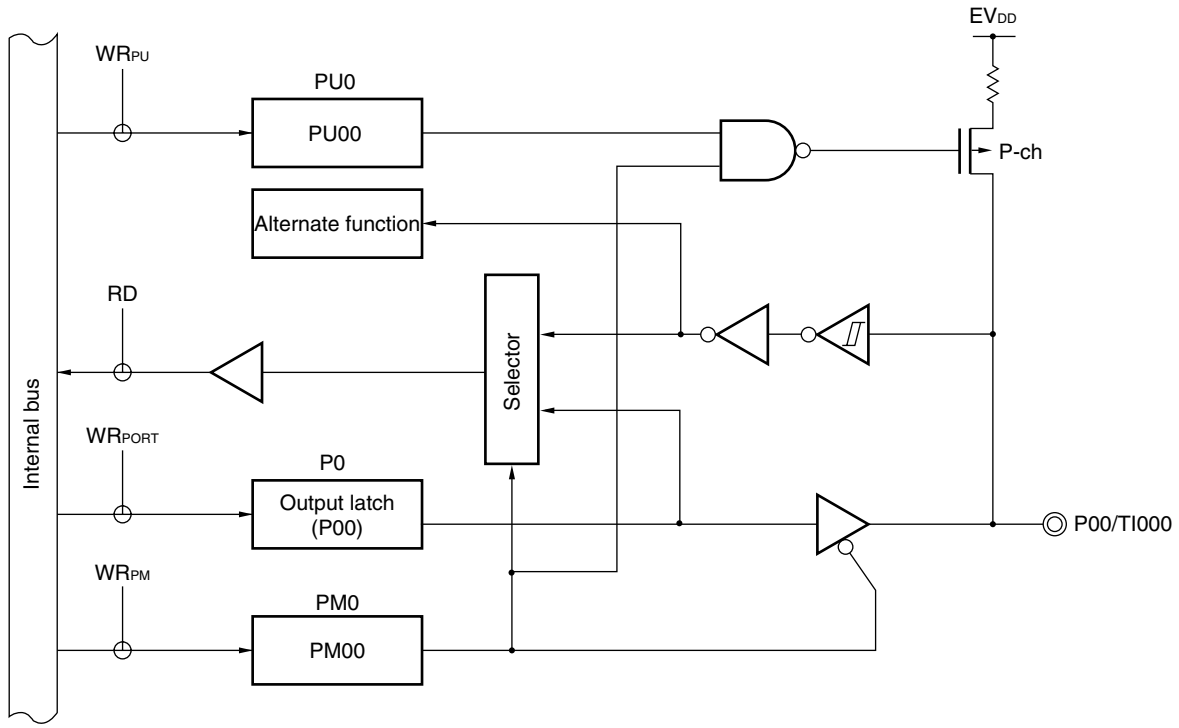
Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P06 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

This port can also be used for timer I/O.

Reset signal generation sets port 0 to input mode.

Figures 4-1 to 4-5 show block diagrams of port 0.

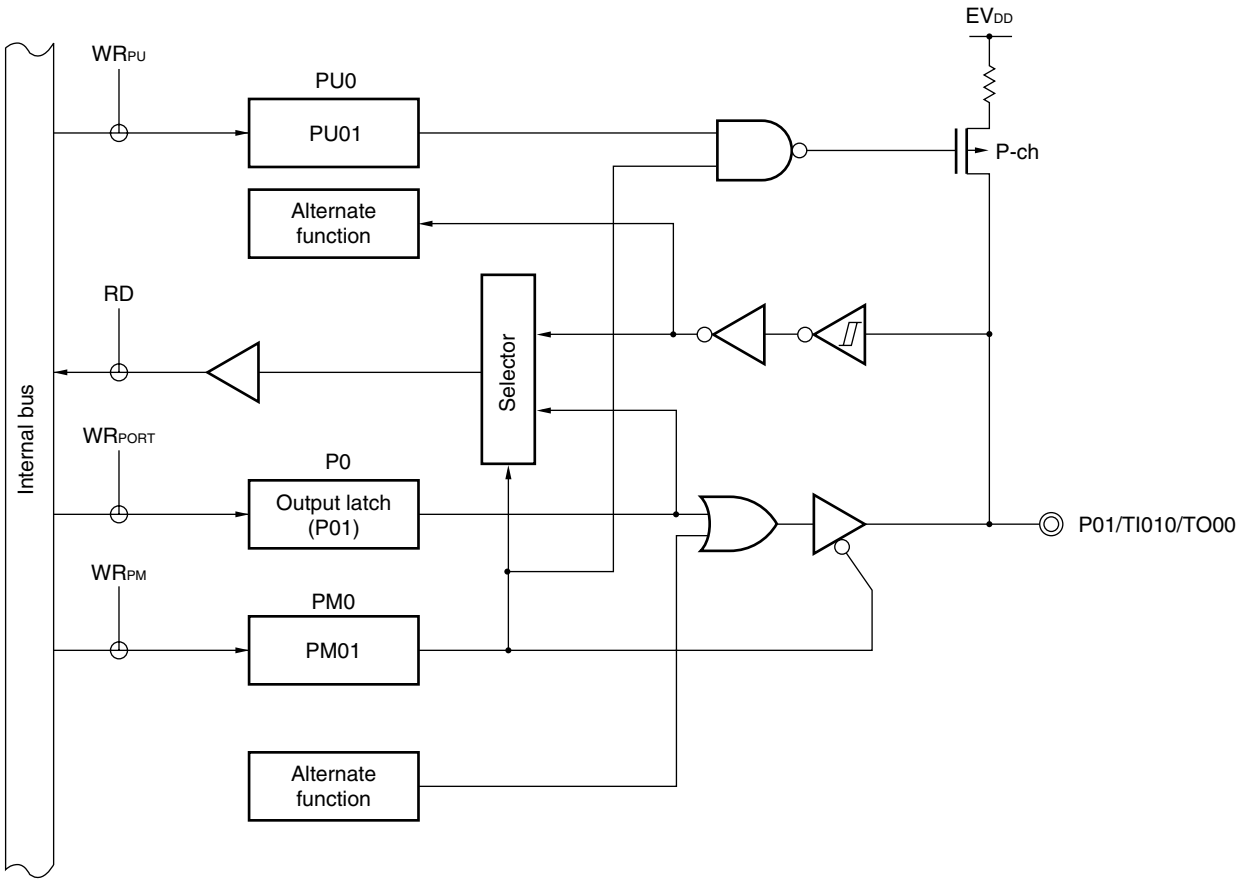
Figure 4-1. Block Diagram of P00



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

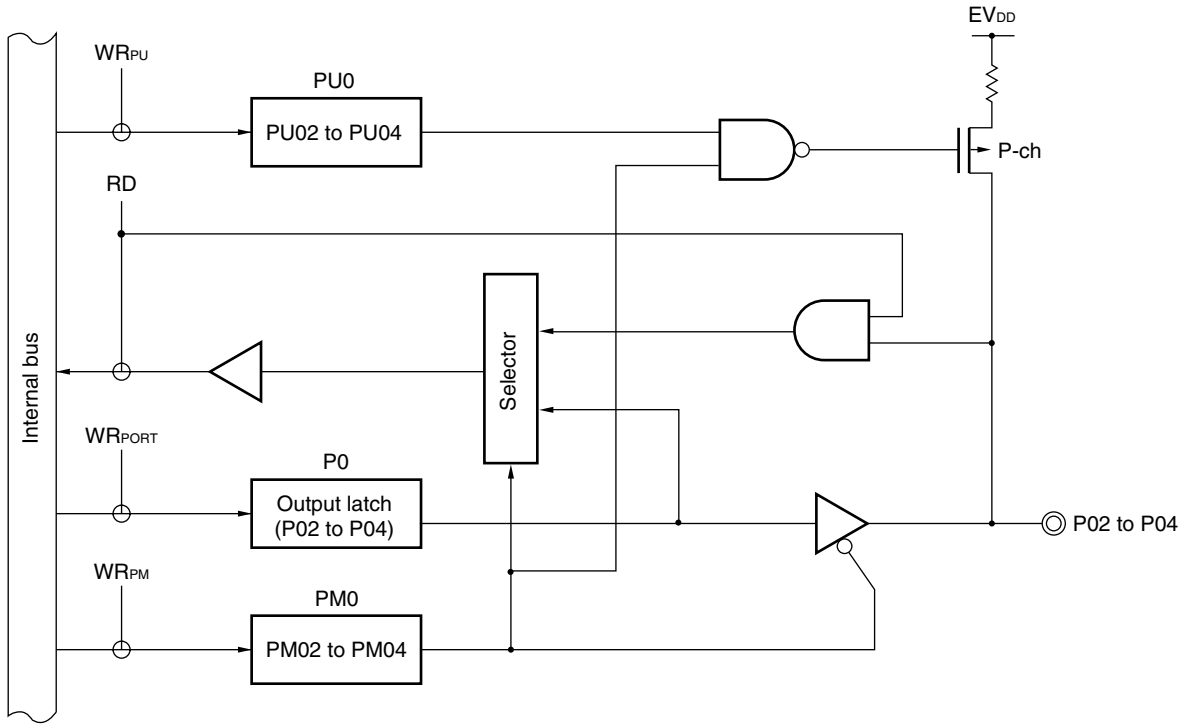
Figure 4-2. Block Diagram of P01



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

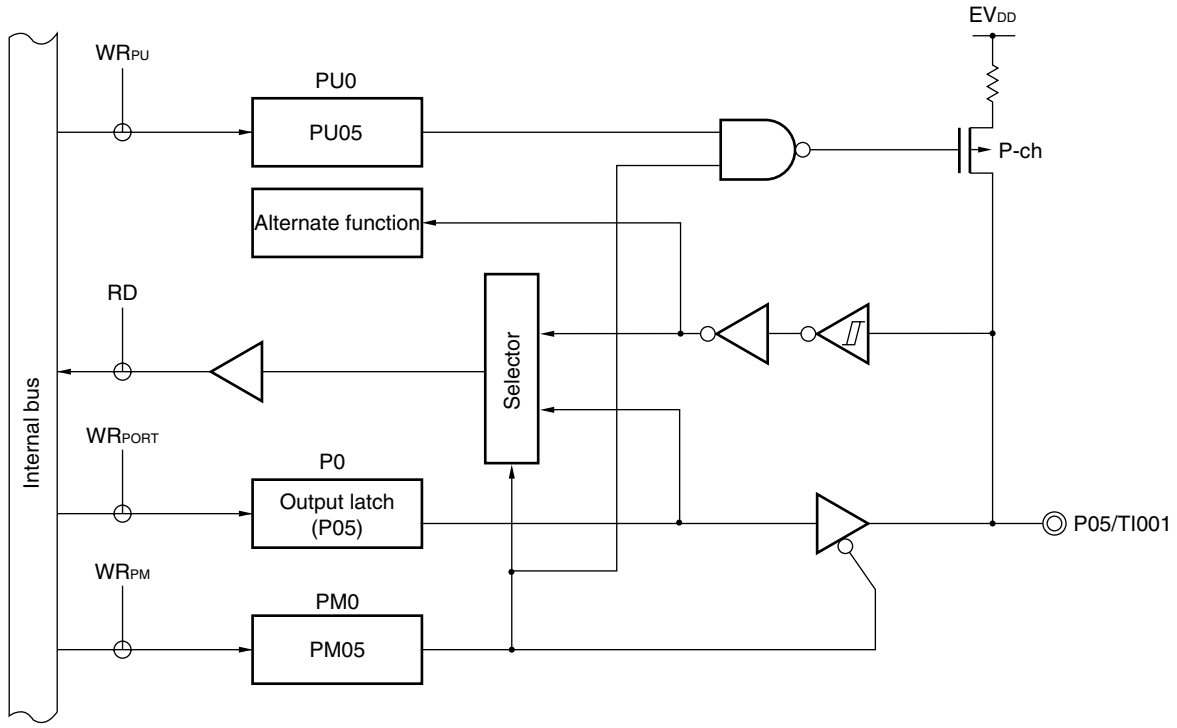
Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Figure 4-3. Block Diagram of P02 to P04



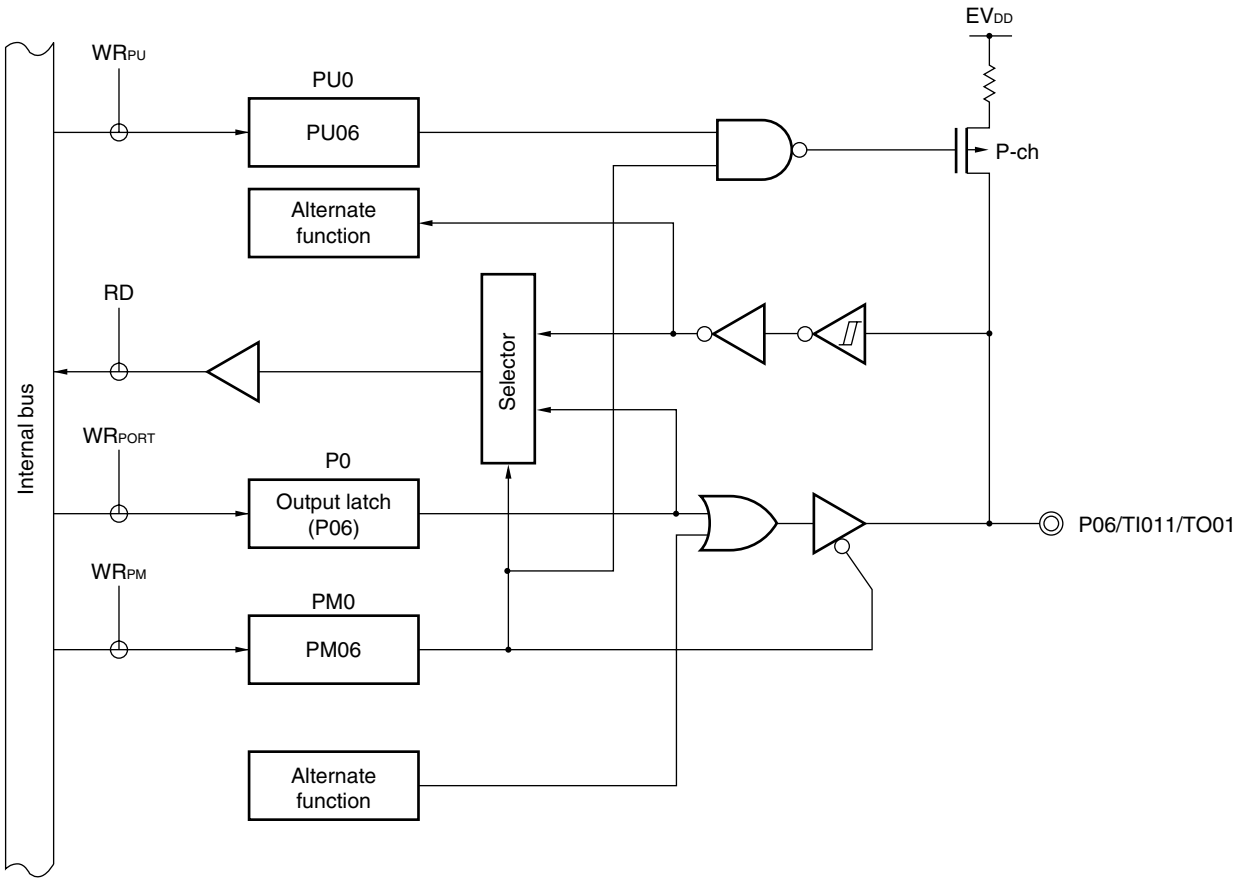
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P05



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

Figure 4-5. Block Diagram of P06



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

4.2.2 Port 1

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P10/$\overline{\text{SCK10}}$/TxD0	√	
P11/SI10/RxD0	√	
P12/SO10	√	
P13/TxD60	√	
P14/RxD60	√	
P15/TOH0	√	
P16/TOH1/INTP5	√	
P17/TI50/TO50	√	

Remark √: Mounted

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to 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).

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

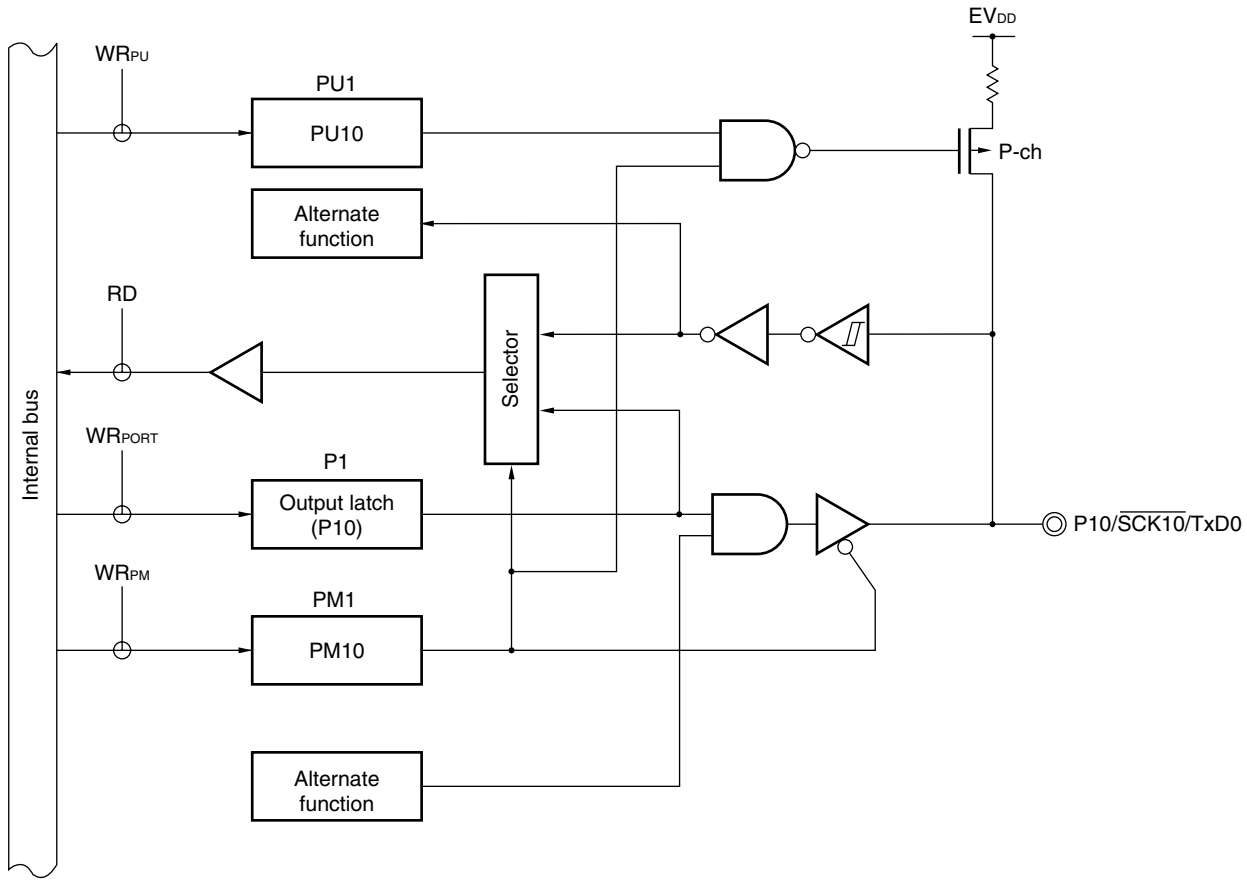
Reset signal generation sets port 1 to input mode.

Figures 4-6 to 4-10 show block diagrams of port 1.

Cautions 1. To use P10/$\overline{\text{SCK10}}$/TxD0 and P12/SO10 as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection register 10 (CSIC10) to the default status (00H).

2. To use P13/TxD60 as general-purpose port, clear bit 0 (TXDLV60) of asynchronous serial interface control register 60 (ASICL60) to 0 (normal output of TxD60).

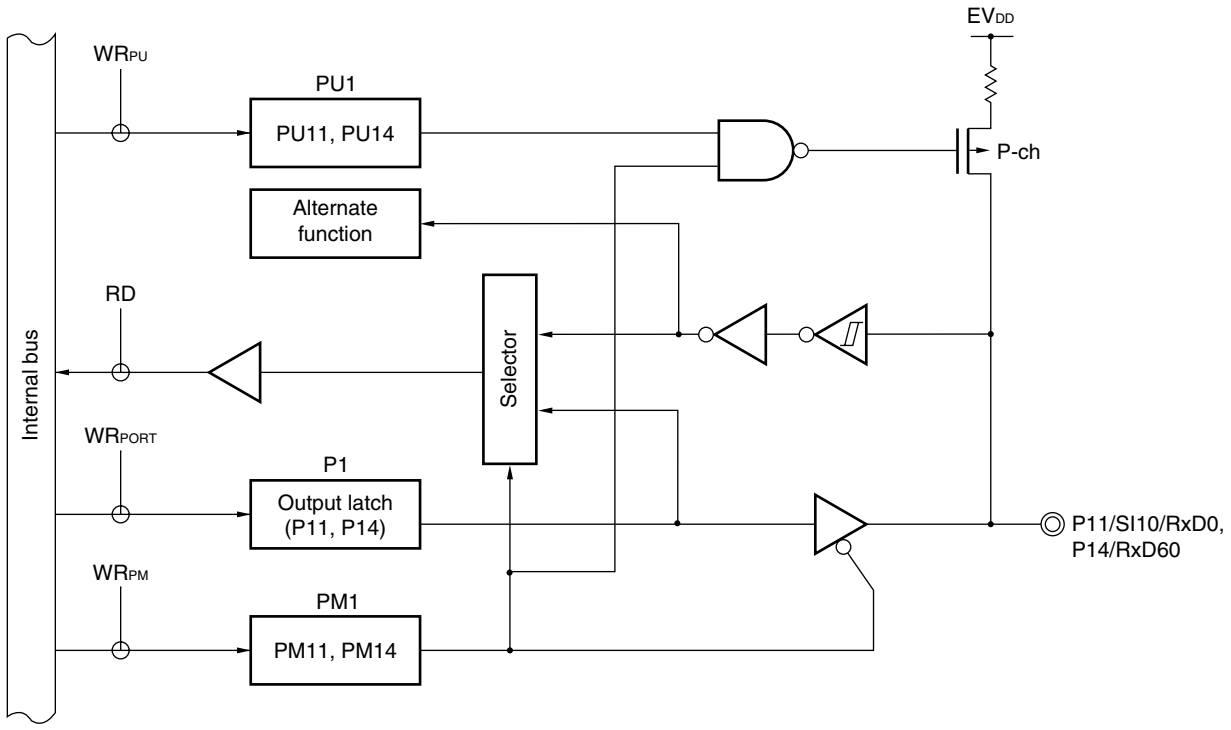
Figure 4-6. Block Diagram of P10



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

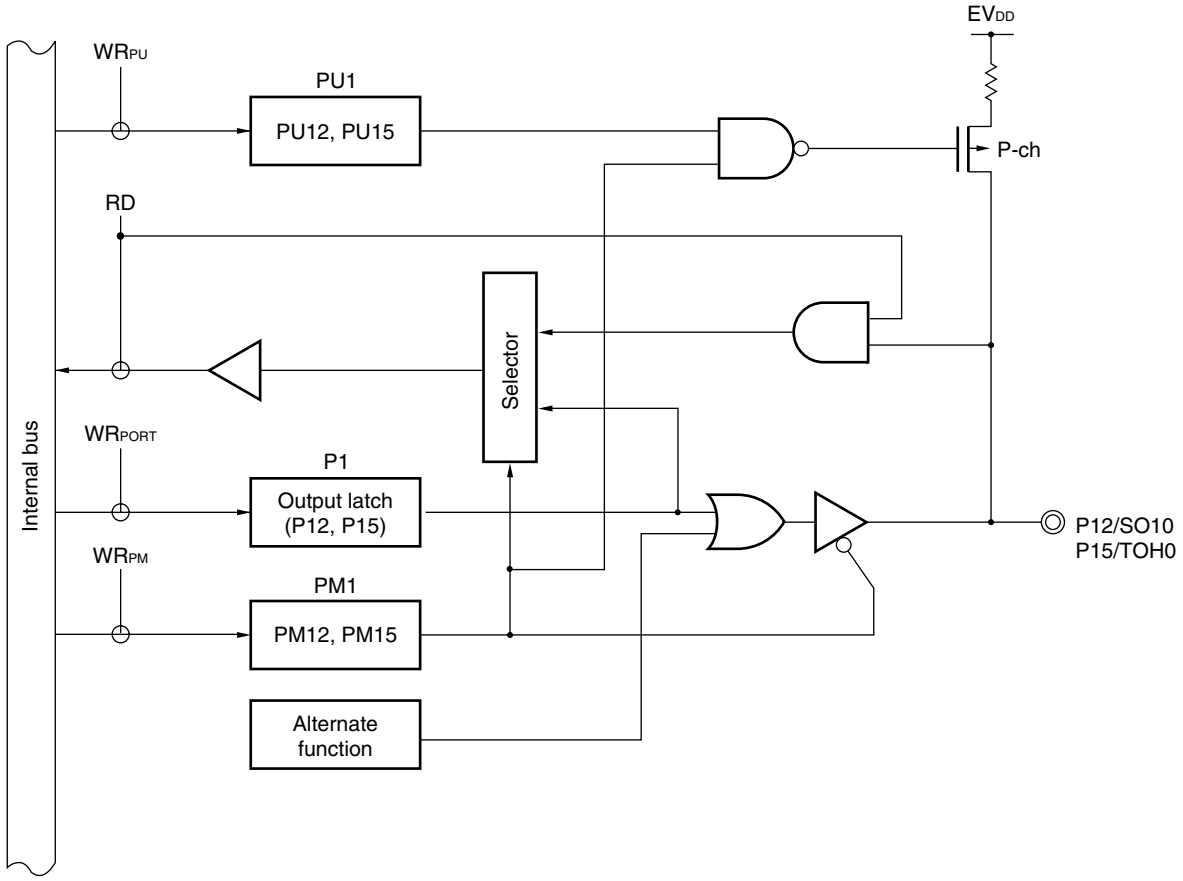
Figure 4-7. Block Diagram of P11 and P14



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

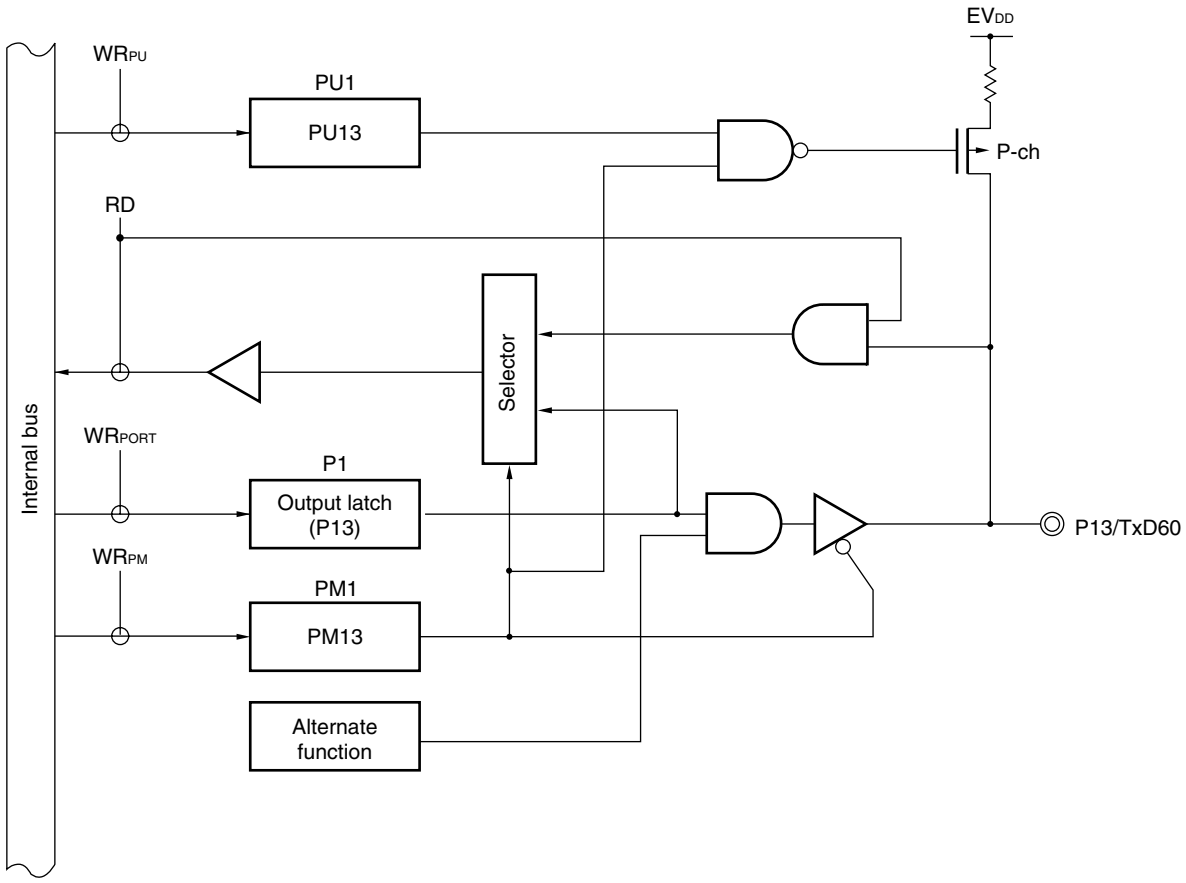
Figure 4-8. Block Diagram of P12 and P15



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

Remark With products not provided with an EVDD or EVSS pin, replace EVDD with VDD, or replace EVSS with VSS.

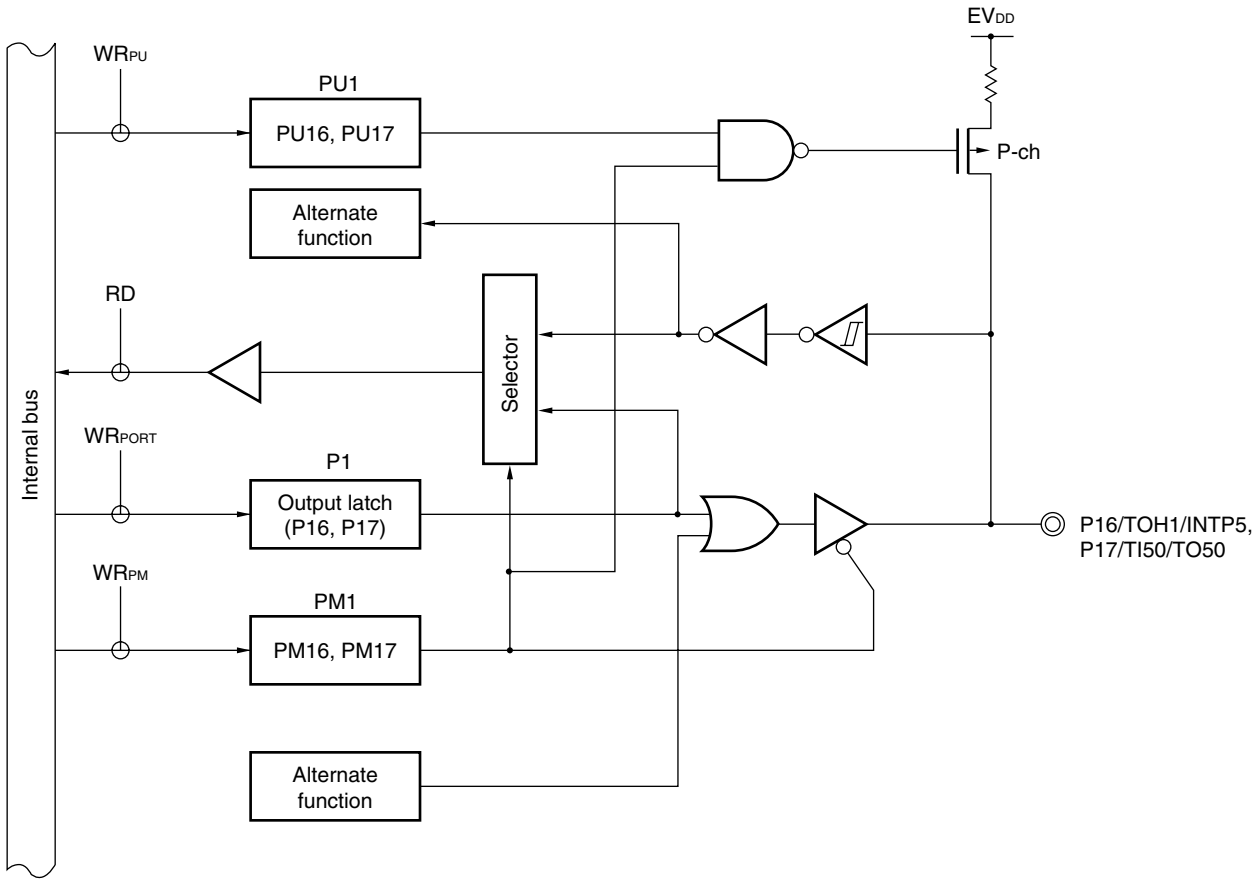
Figure 4-9. Block Diagram of P13



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

Remark With products not provided with an EVDD or EVSS pin, replace EVDD with VDD, or replace EVSS with VSS.

Figure 4-10. Block Diagram of P16 and P17



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

Remark With products not provided with an EVDD or EVSS pin, replace EVDD with VDD, or replace EVSS with VSS.

4.2.3 Port 2

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P20/ANI0	√	
P21/ANI1	√	
P22/ANI2	√	
P23/ANI3	√	
P24/ANI4	√	
P25/ANI5	√	
P26/ANI6	√	
P27/ANI7	√	

Remark √: Mounted

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

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

To use P20/ANI0 to 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 to 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.

Table 4-5. Setting Functions of P20/ANI0 to P27/ANI7 Pins

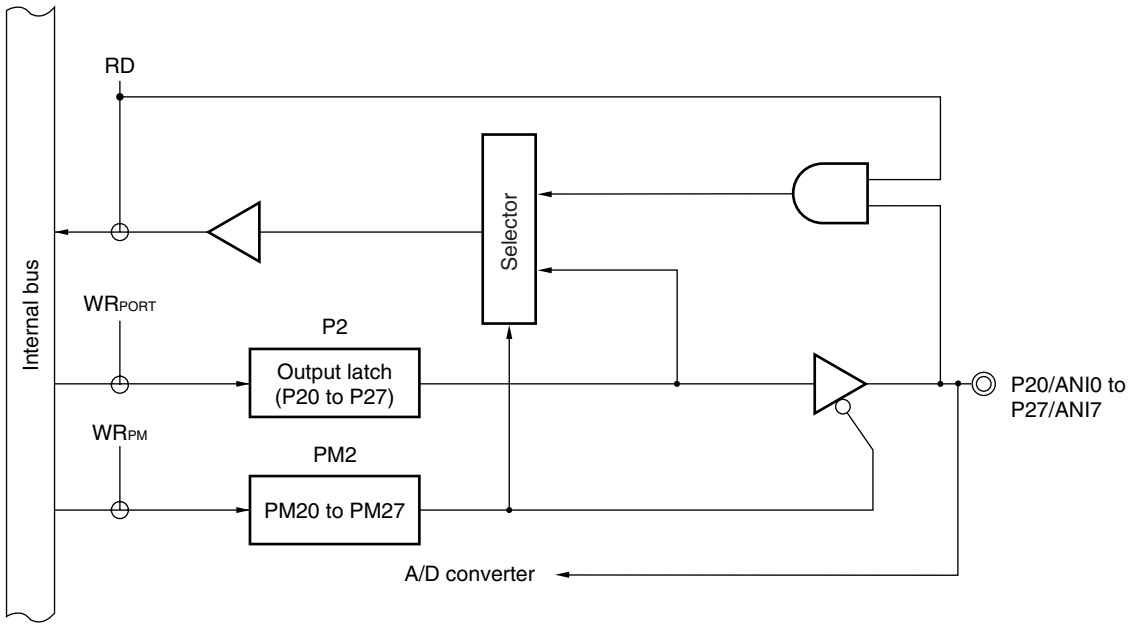
ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pin
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does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

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

Figure 4-11 shows a block diagram of port 2.

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

Figure 4-11. Block Diagram of P20 to P27



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

4.2.4 Port 3

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P30/INTP1	√	
P31/INTP2/OCD1A	√	
P32/ROUT/INTP3/OCD1B	√	
P33/INTP4/TI51/TO51	√	

Remark √: Mounted

Port 3 is a 4-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P33 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, remote control receive data output, and timer I/O.

Reset signal generation sets port 3 to input mode.

Figures 4-12 and 4-13 show block diagrams of port 3.

Cautions 1. Be sure to pull the P31/INTP2/OCD1A pin down before a reset release, to prevent malfunction.

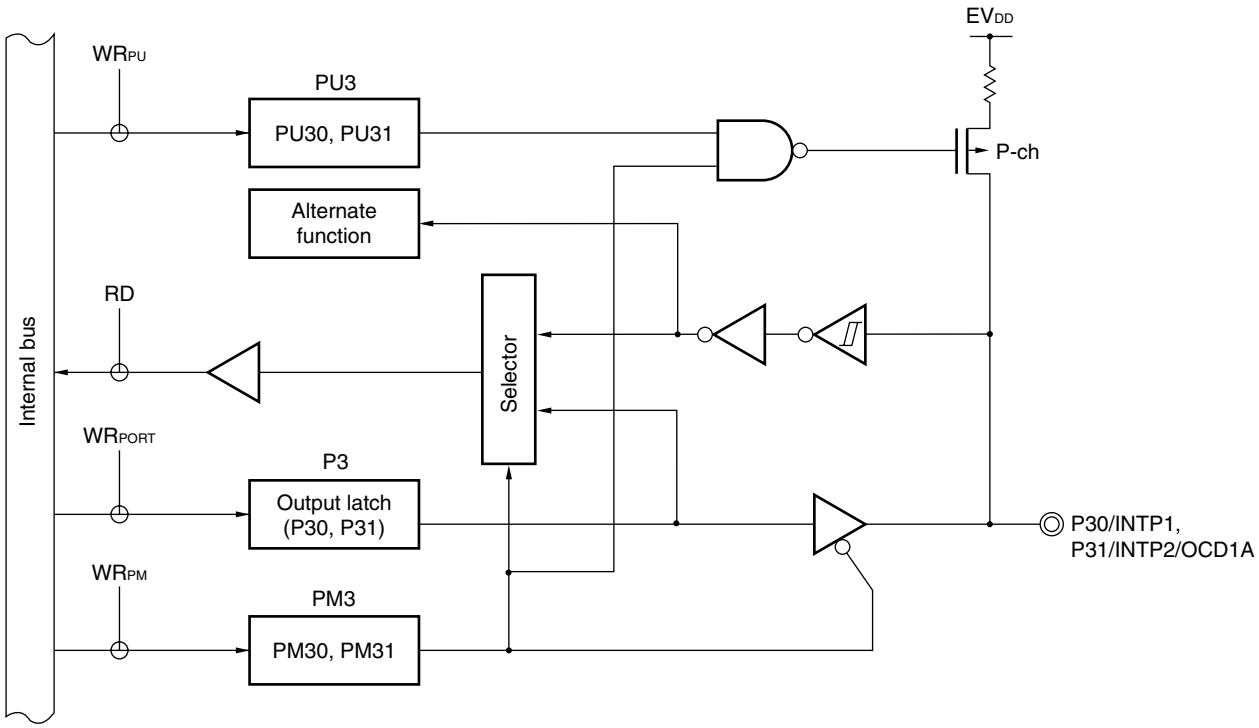
2. Process the P31/INTP2/OCD1A pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P31/INTP2/OCD1A
Flash memory programmer connection		Connect to EV_{SS}[Note] via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Input: Connect to EV_{DD}[Note] or EV_{SS}[Note] via a resistor. Output: Leave open.

Note With products without an EV_{SS} pin, connect them to V_{SS}. With products without an EV_{DD} pin, connect them to V_{DD}.

Remark P31 and P32 can be used as on-chip debug mode setting pins (OCD1A, OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 28 ON-CHIP DEBUG FUNCTION.**

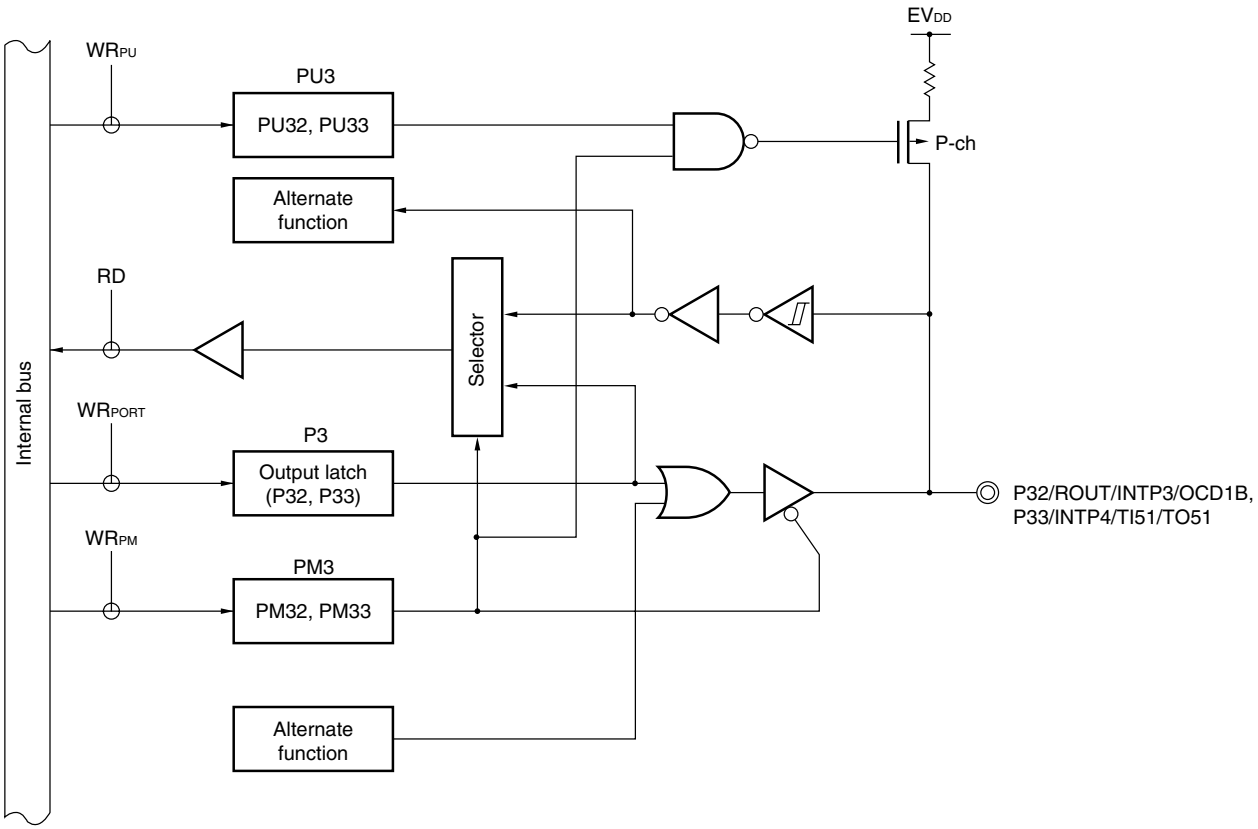
Figure 4-12. Block Diagram of P30, P31



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Figure 4-13. Block Diagram of P32, P33



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

4.2.5 Port 4

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P40/CECOUT/RTC1HZ	√	√
P41/CECIN/RTCDIV/RTCCL	√	√
P42/TxD61	–	√
P43/RxD61	–	√

Remark √: Mounted, –: Not mounted

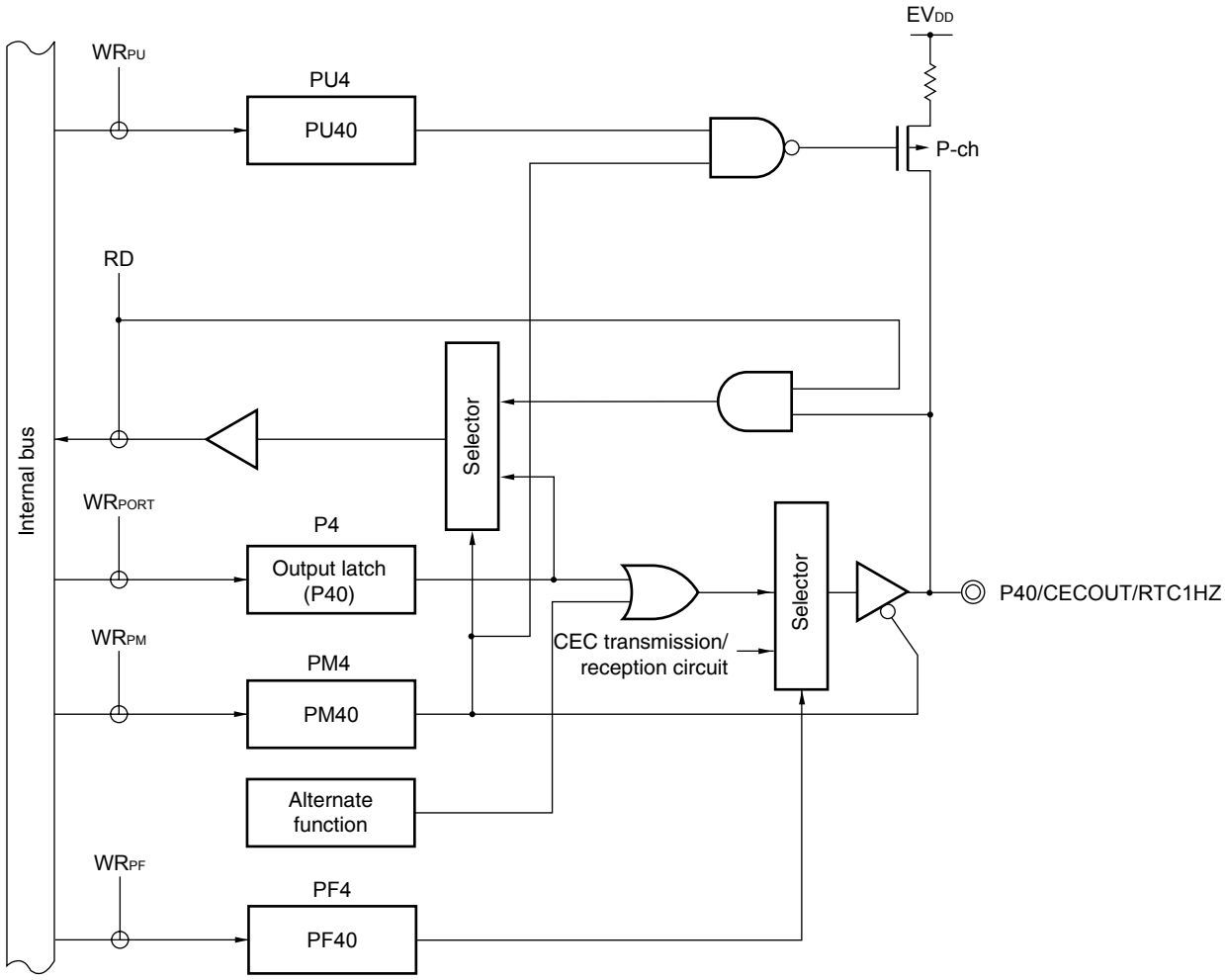
Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P43 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

This port can also be used for serial data I/O for CEC, real-time counter correction clock output, real-time counter clock output, and serial interface data I/O.

Reset signal generation sets port 4 to input mode.

Figures 4-14 to 4-17 show a block diagram of port 4.

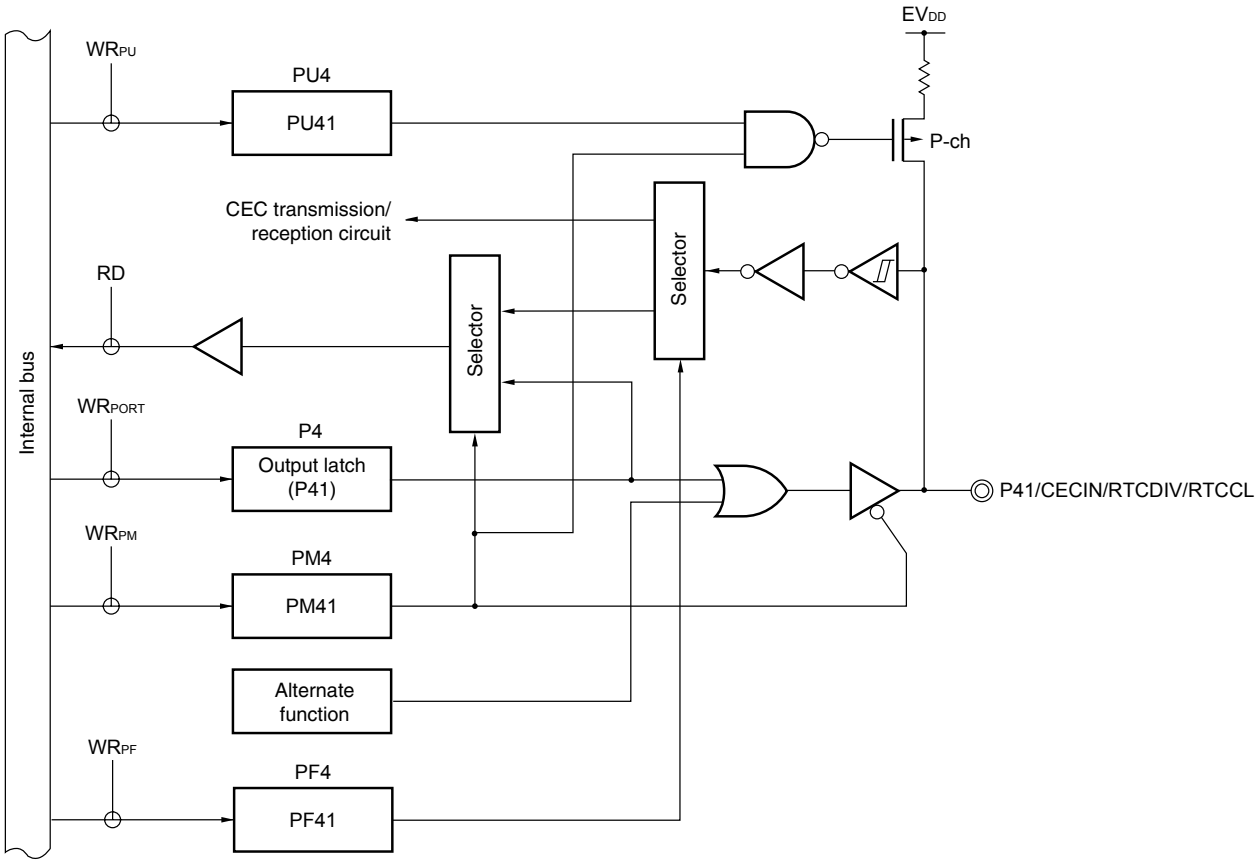
Figure 4-14. Block Diagram of P40



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PF4: Port function register 4
RD: Read signal
WR××: Write signal

Remark With products not provided with an EVDD or EVSS pin, replace EVDD with VDD, or replace EVSS with VSS.

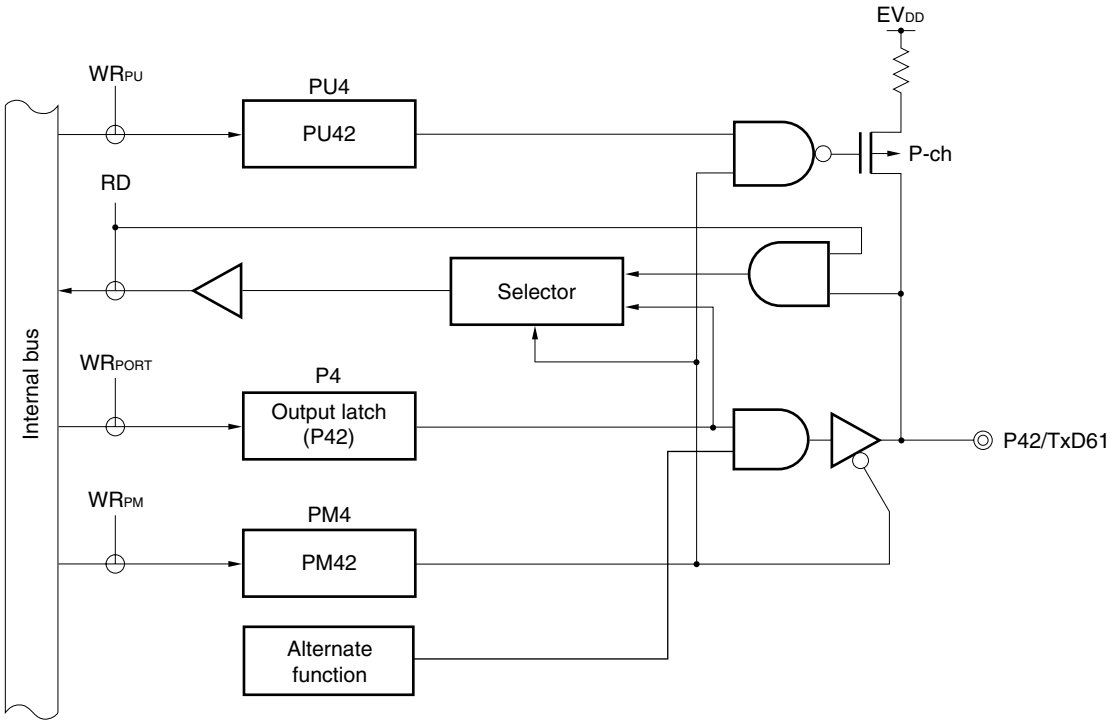
Figure 4-15. Block Diagram of P41



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PF4: Port function register 4
RD: Read signal
WR××: Write signal

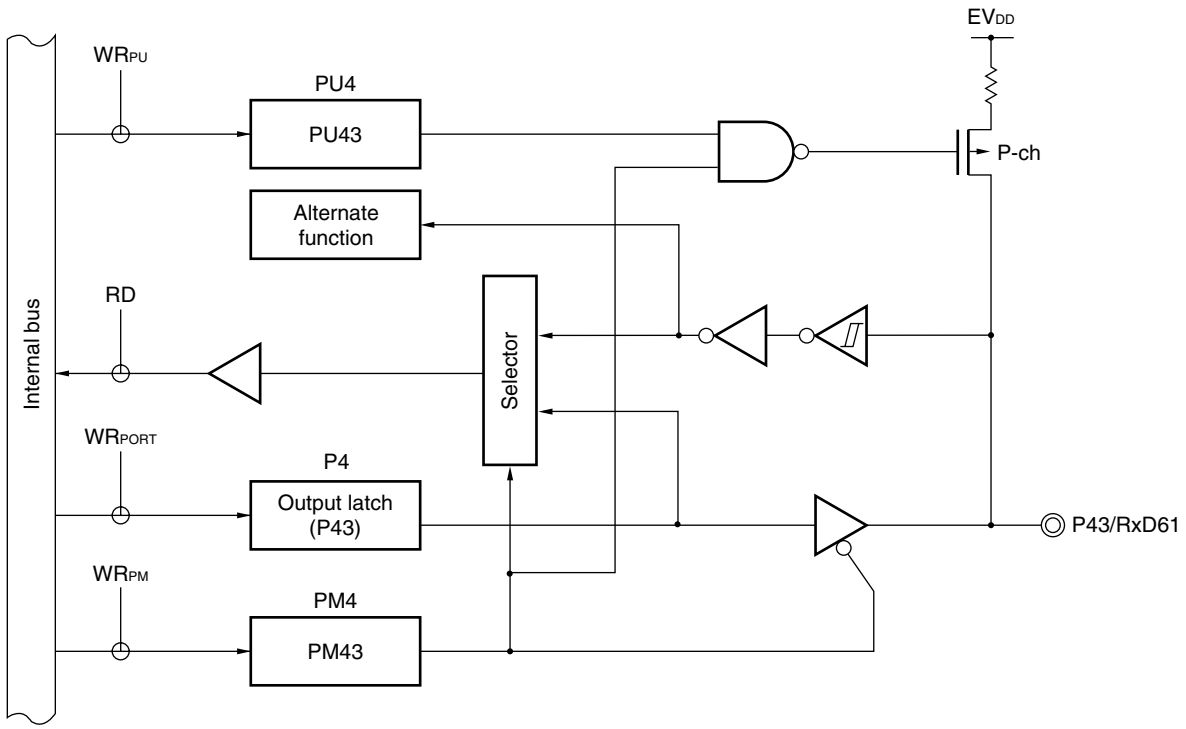
Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Figure 4-16. 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 4-17. 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

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P50/TI002	–	√
P51/TI012/TO02	–	√
P52	–	√
P53	–	√

Remark √: Mounted, –: Not mounted

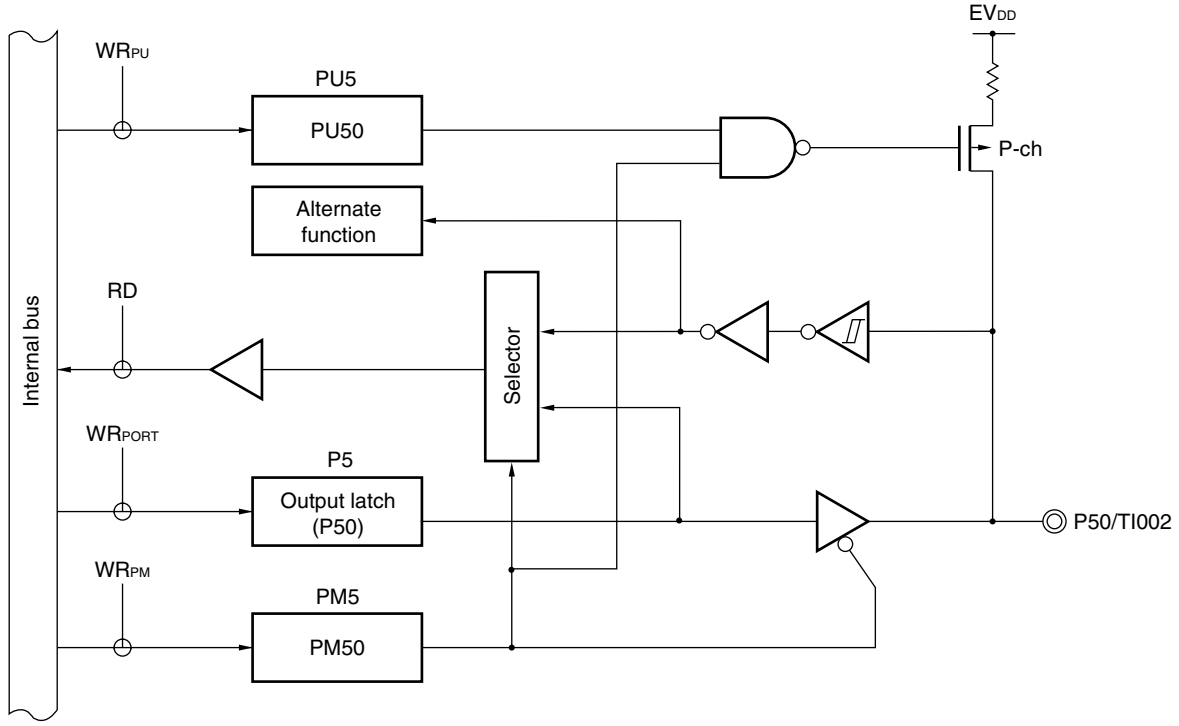
Port 5 is a 4-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P53 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

This port can also be used for timer I/O.

Reset signal generation sets port 5 to input mode.

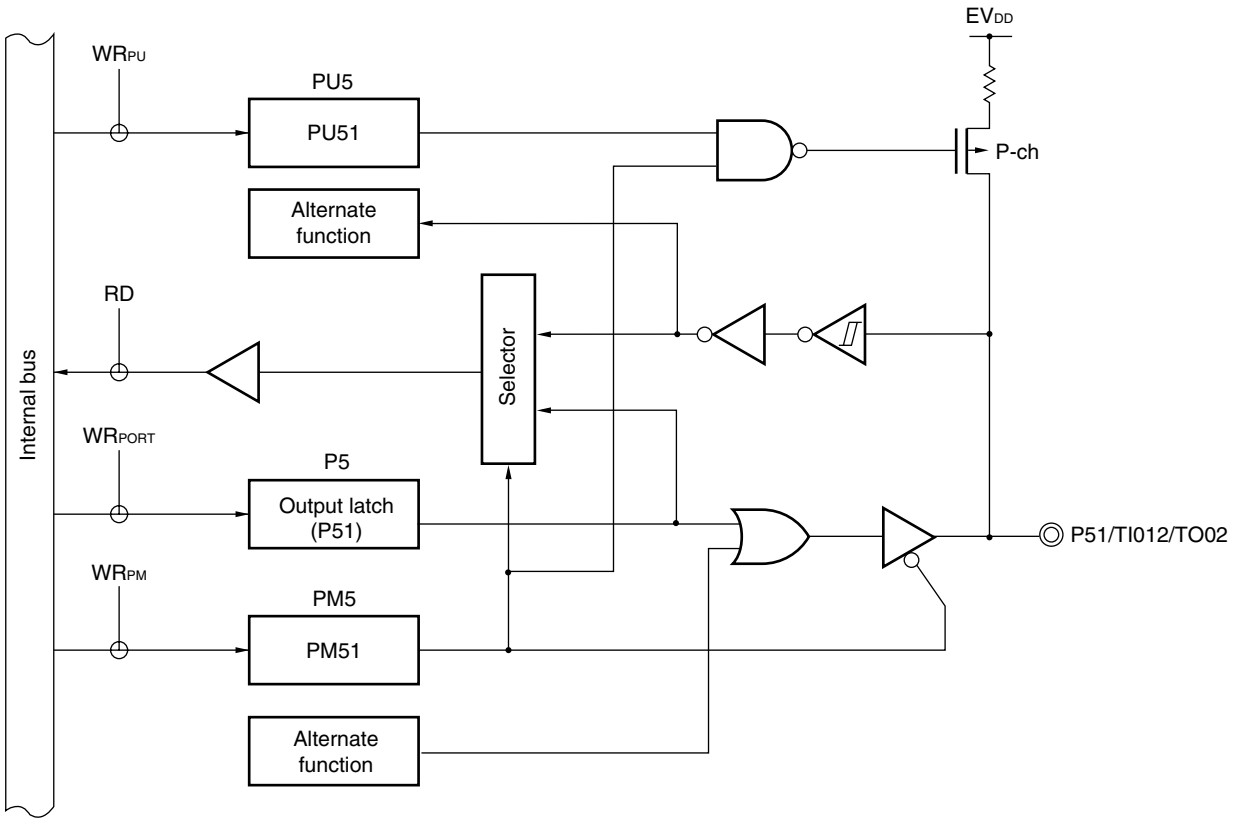
Figures 4-18 to 4-20 show a block diagram of port 5.

Figure 4-18. Block Diagram of P50



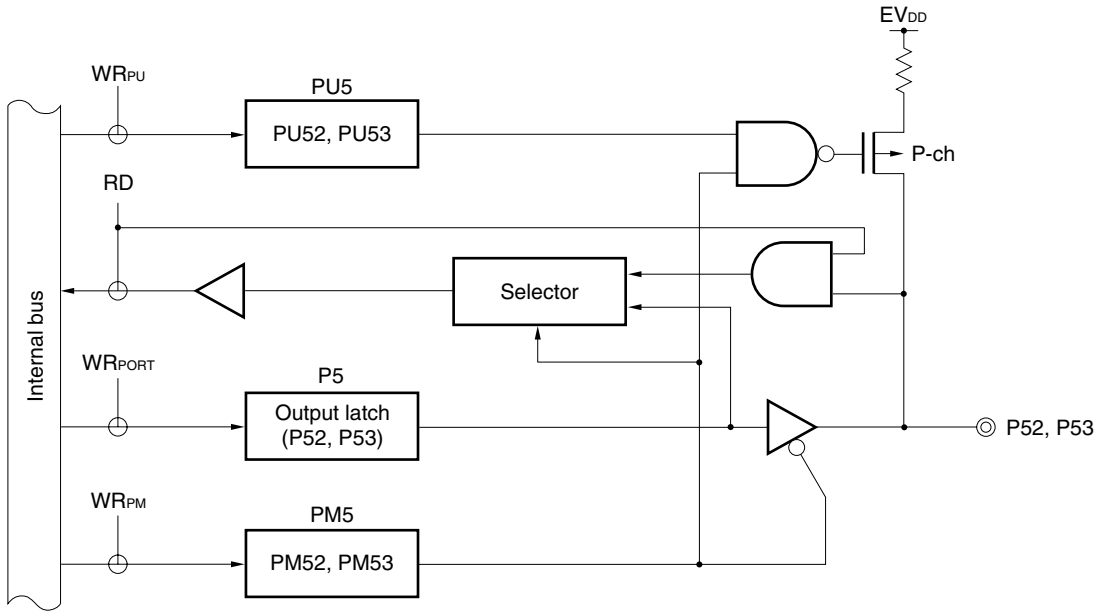
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

Figure 4-19. Block Diagram of P51



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

Figure 4-20. Block Diagram of P52 and P53



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

4.2.7 Port 6

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P60/SCLA0	√	
P61/SDAA0	√	
P62/SCLA1	√	
P63/SDAA1	√	

Remark √: Mounted

Port 6 is a 4-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

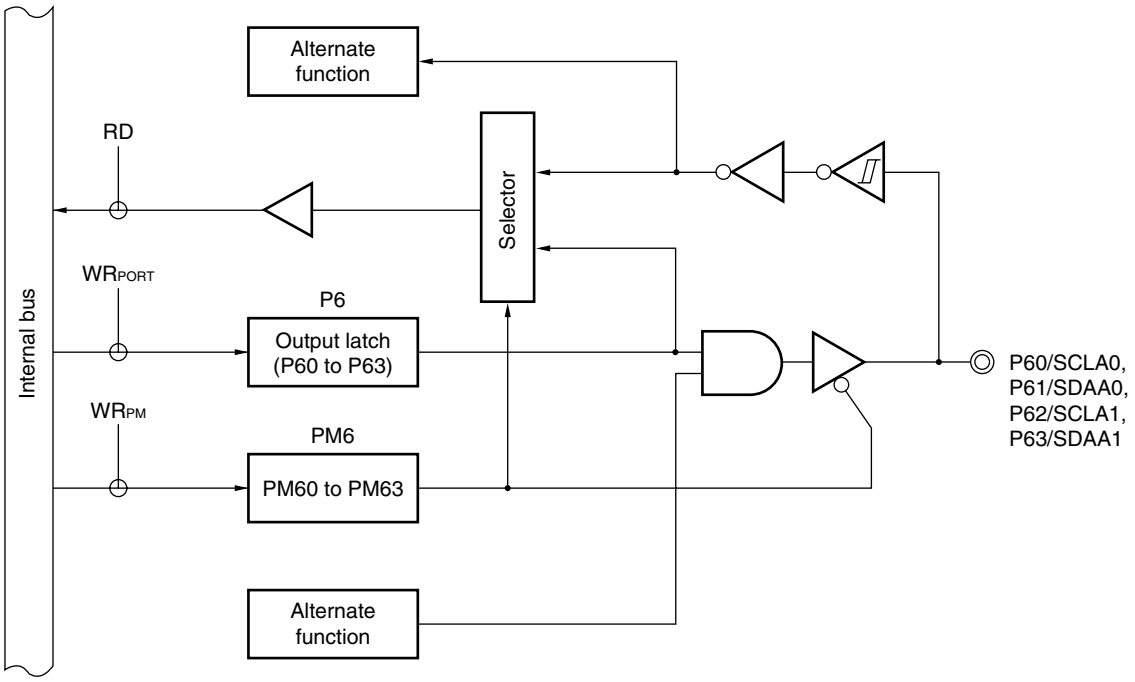
The output of the P60 to P63 pins is N-ch open-drain output (6 V tolerance).

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.

Figure 4-21 shows block diagrams of port 6.

Figure 4-21. Block Diagram of P60 to P63



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

Caution A through current flows through P60 to P63 if an intermediate potential is input to these pins, because the input buffer is also turned on when P60 to P63 are in output mode. Consequently, do not input an intermediate potential when P60 to P63 are in output mode.

4.2.8 Port 7

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P70/SCLA2/KR0	√	√
P71/SDAA2/KR1	√	√
P72/CECIO/KR2	√	√
P73/SI11/KR3	√	√
P74/SO11/KR4	P74/SO11[Note]	√
P75/$\overline{\text{SCK11}}$/KR5	P75/$\overline{\text{SCK11}}$[Note]	√
P76//KR6/$\overline{\text{SSI11}}$	–	√
P77/KR7	–	√

Note The 78K0/KC2-C is not provided with the KR4 and KR5 pins.

Remark √: Mounted, –: Not mounted

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P72 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

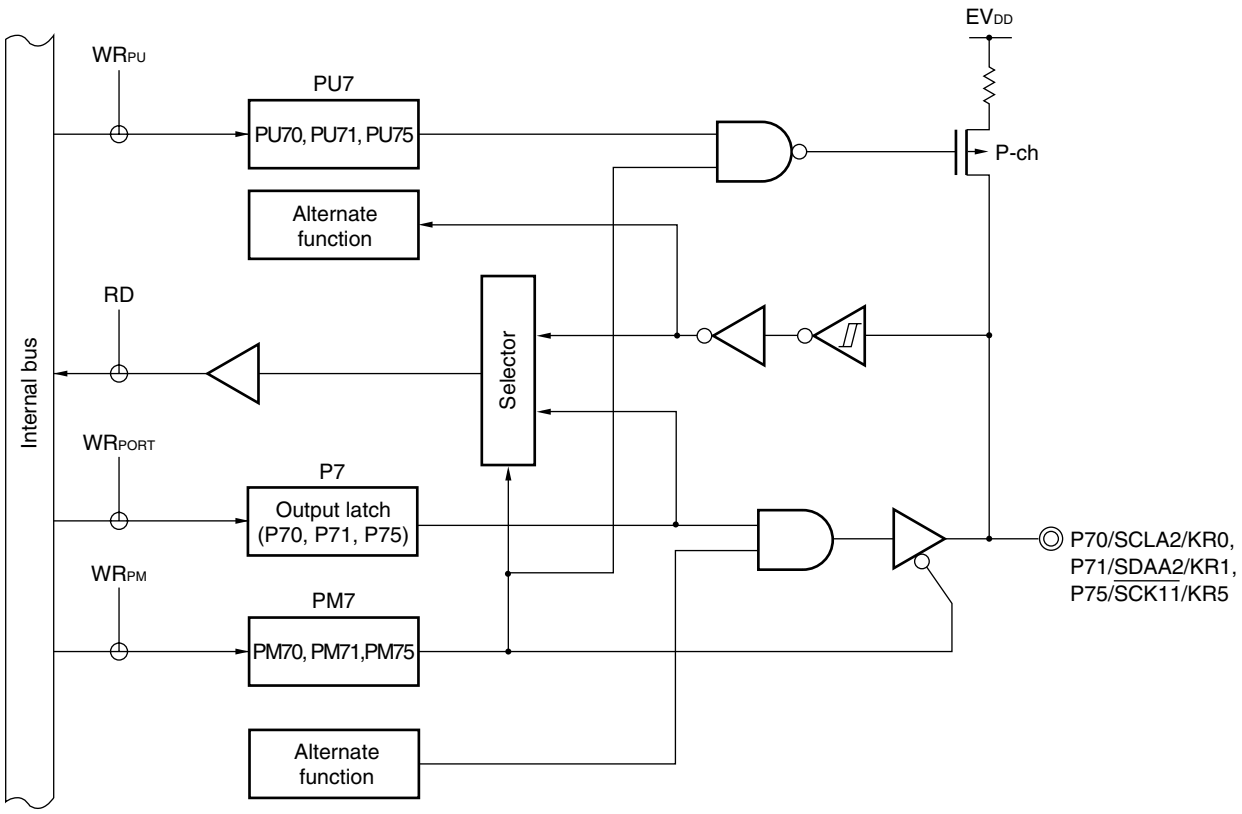
The output of the P70 to P77 pins is N-ch open-drain output (6 V tolerance).

This port can also be used for serial interface data I/O, clock I/O, chip select input, serial data I/O for CEC, and key return input.

Reset signal generation sets port 7 to input mode.

Figures 4-22 to 4-25 show a block diagram of port 7.

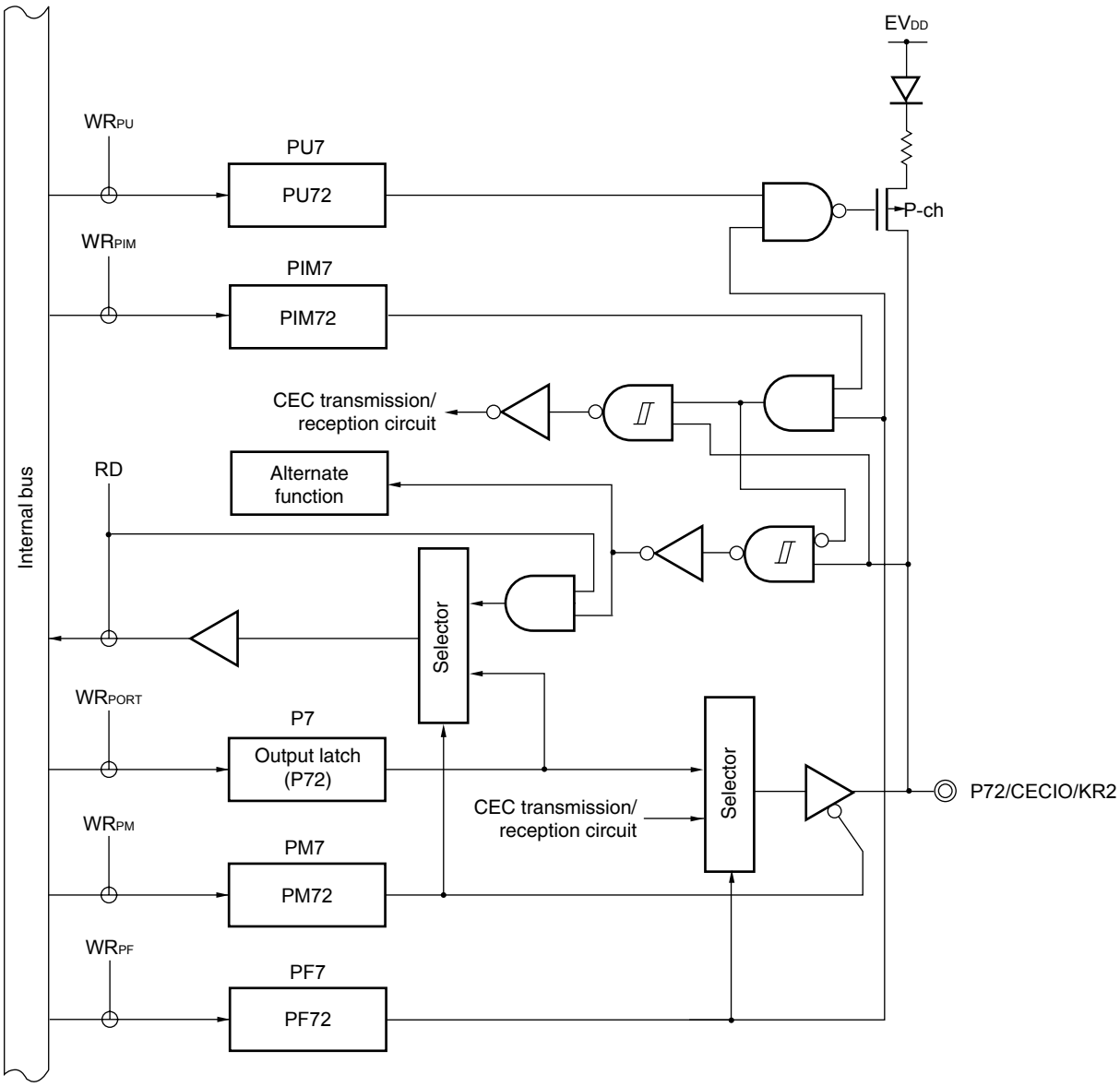
Figure 4-22. Block Diagram of P70, P71, P75



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

Remark With products not provided with an EVDD or EVSS pin, replace EVDD with VDD, or replace EVSS with VSS.

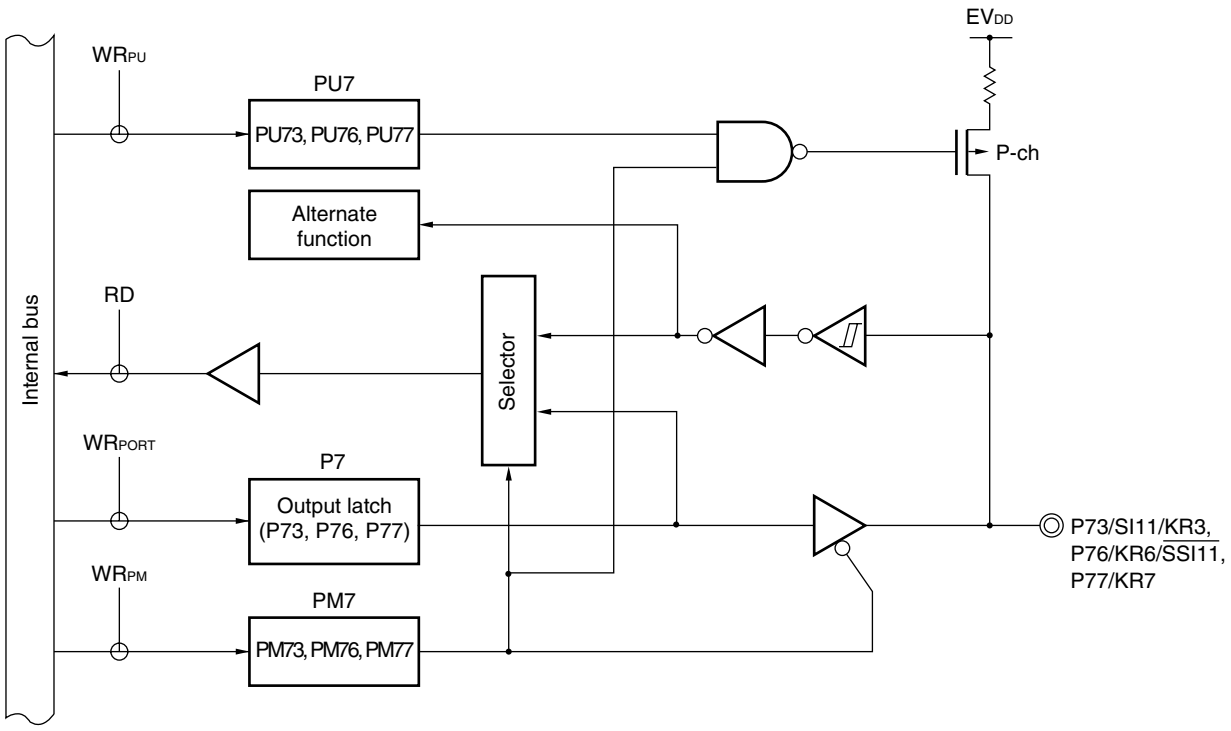
Figure 4-23. Block Diagram of P72



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

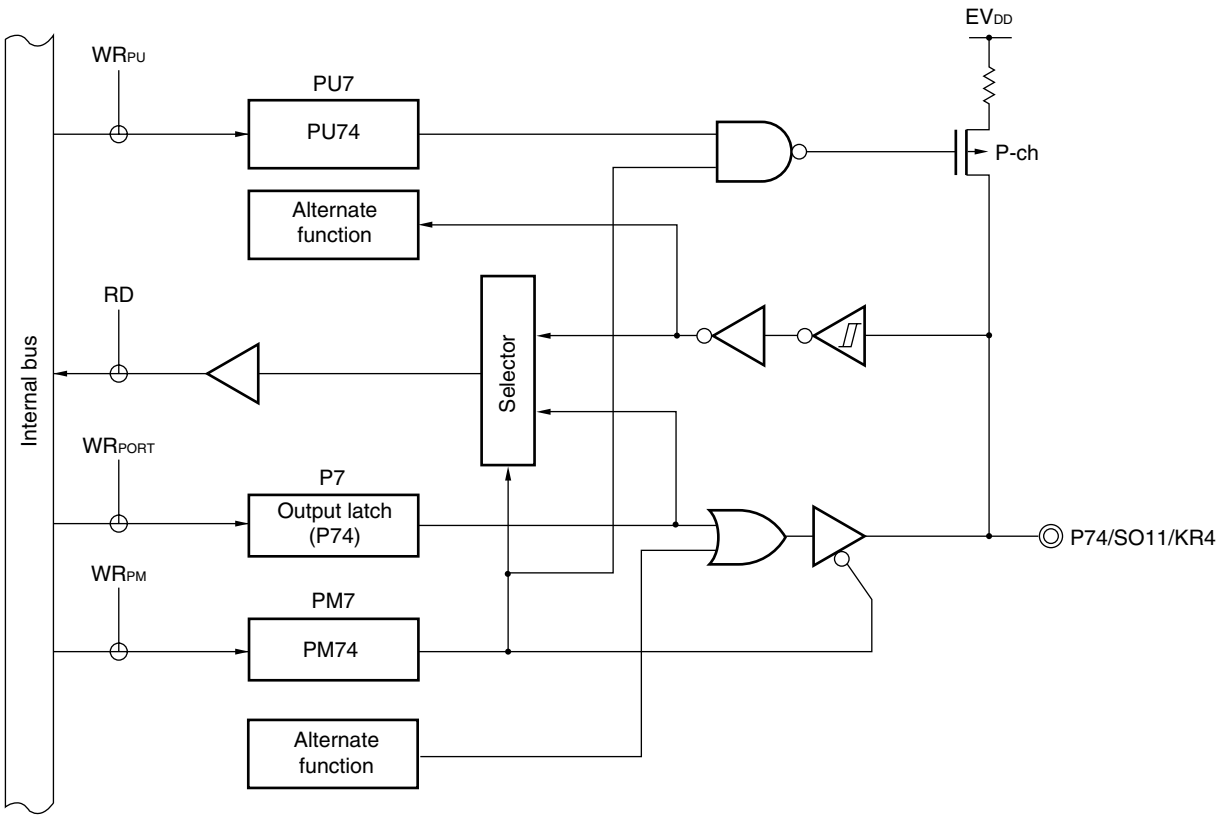
Figure 4-24. Block Diagram of P73, P76, P77



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Figure 4-25. Block Diagram of P74



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

4.2.9 Port 12

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P120/INTP0/EXLVI	√	
P121/X1/OCD0A	√	
P122/X2/EXCLK/OCD0B	√	
P123/XT1	√	
P124/XT2	√	

Remark √: Mounted

P120 is a 1-bit I/O port with an output latch. P120 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 only for P120, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 are 4-bit input ports.

This port can also be used as pins 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 4-26 and 4-27 show block diagrams of port 12.

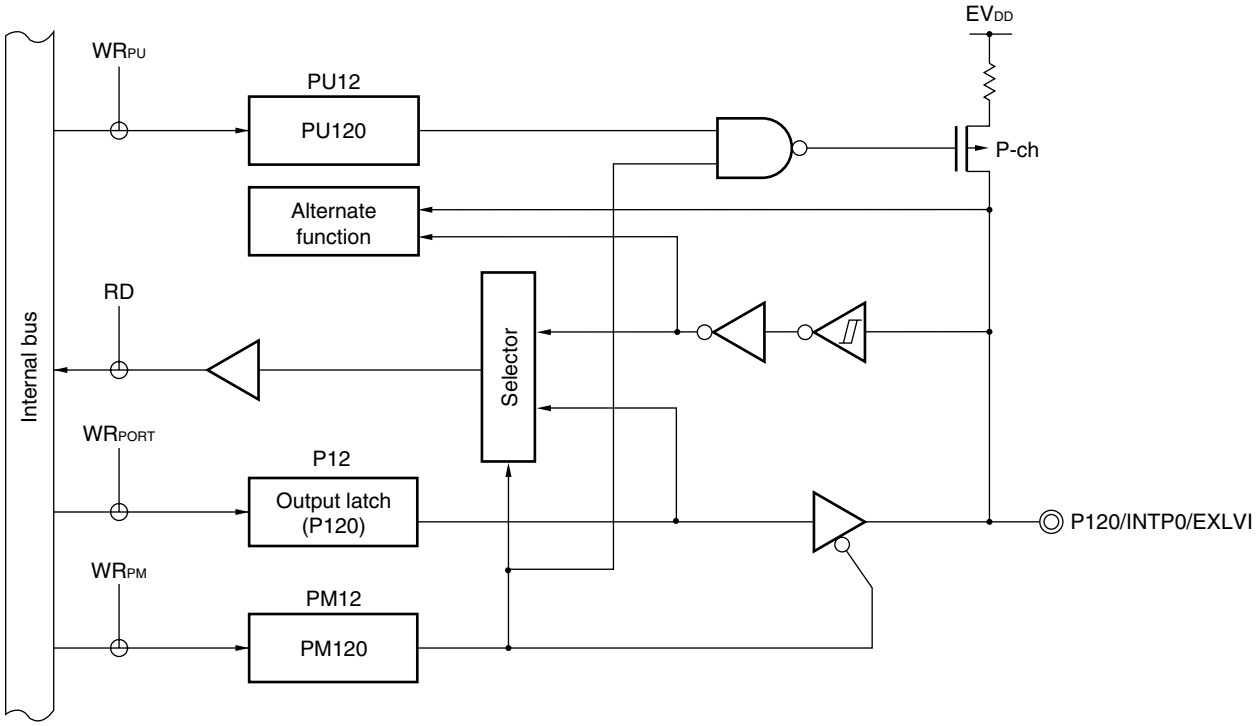
Cautions 1. When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2) or subsystem clock (XT1, XT2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, see 5.3 (1) Clock operation mode select register (OSCCTL) and (3) Setting of operation mode for subsystem clock pin). The reset value of OSCCTL is 00H (all of the P121 to P124 pins are input port pins). At this time, setting of the P121 to P124 pins is not necessary.

2. Process the P121/X1/OCD0A pin as follows, when it is not used when it is connected to a flash memory programmer or an on-chip debug emulator.

		P121/X1/OCD0A
Flash memory programmer connection		Connect to V_{SS} via a resistor.
On-chip debug emulator connection (when it is not used as an on-chip debug mode setting pin)	During reset	
	During reset released	Connect to V_{DD} or V_{SS} via a resistor.

Remark X1 and X2 can be used as on-chip debug mode setting pins (OCD0A, OCD0B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 28 ON-CHIP DEBUG FUNCTION.**

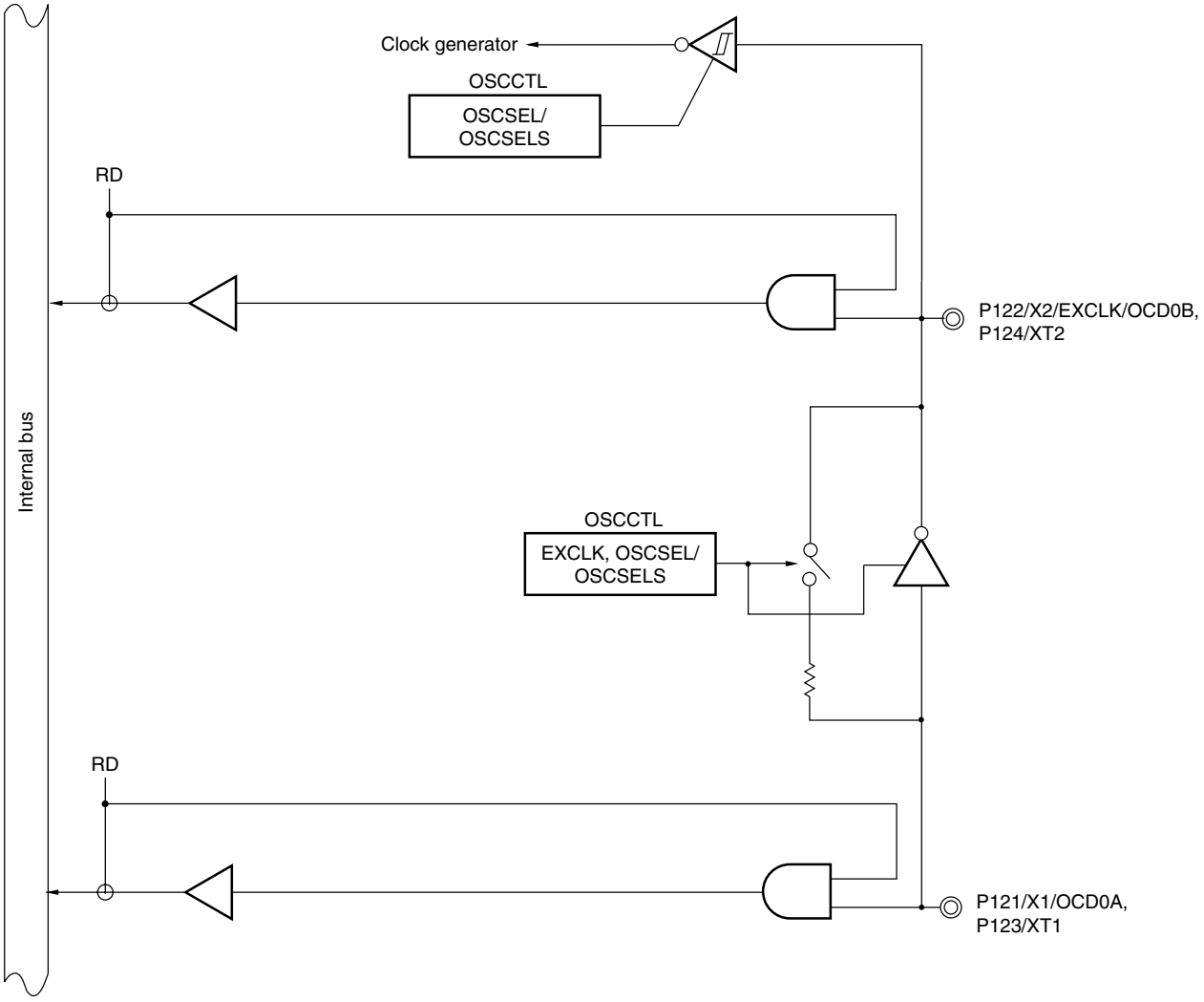
Figure 4-26. Block Diagram of P120



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Figure 4-27. Block Diagram of P121 to P124



OSCCTL: Clock operation mode select register
RD: Read signal

4.2.10 Port 13

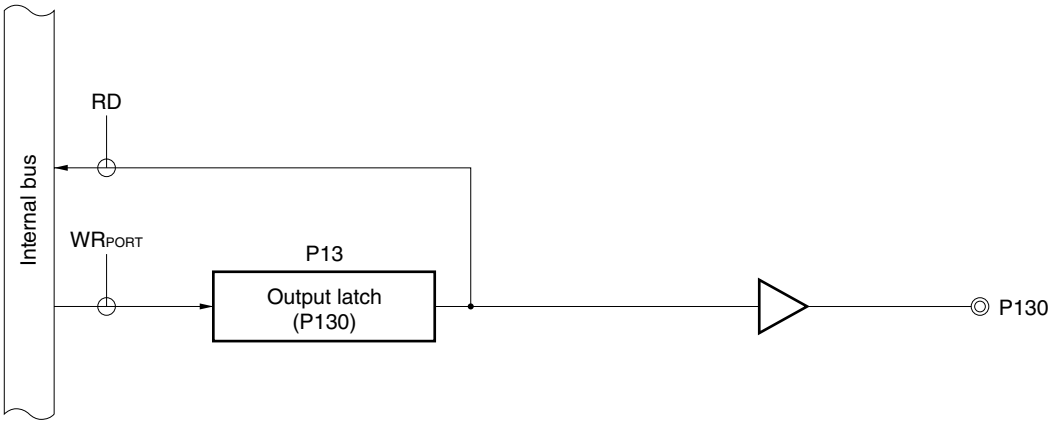
	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P130	√	

Remark √: Mounted

Port 13 is an output-only port.

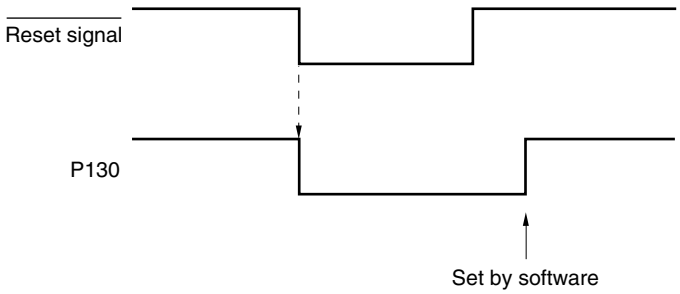
Figure 4-28 shows a block diagram of port 13.

Figure 4-28. Block Diagram of P130



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

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.



4.2.11 Port 14

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
P140/PCL/INTP6/RIN	√	√
P141/BUZ	–	√

Remark √: Mounted, –: Not mounted

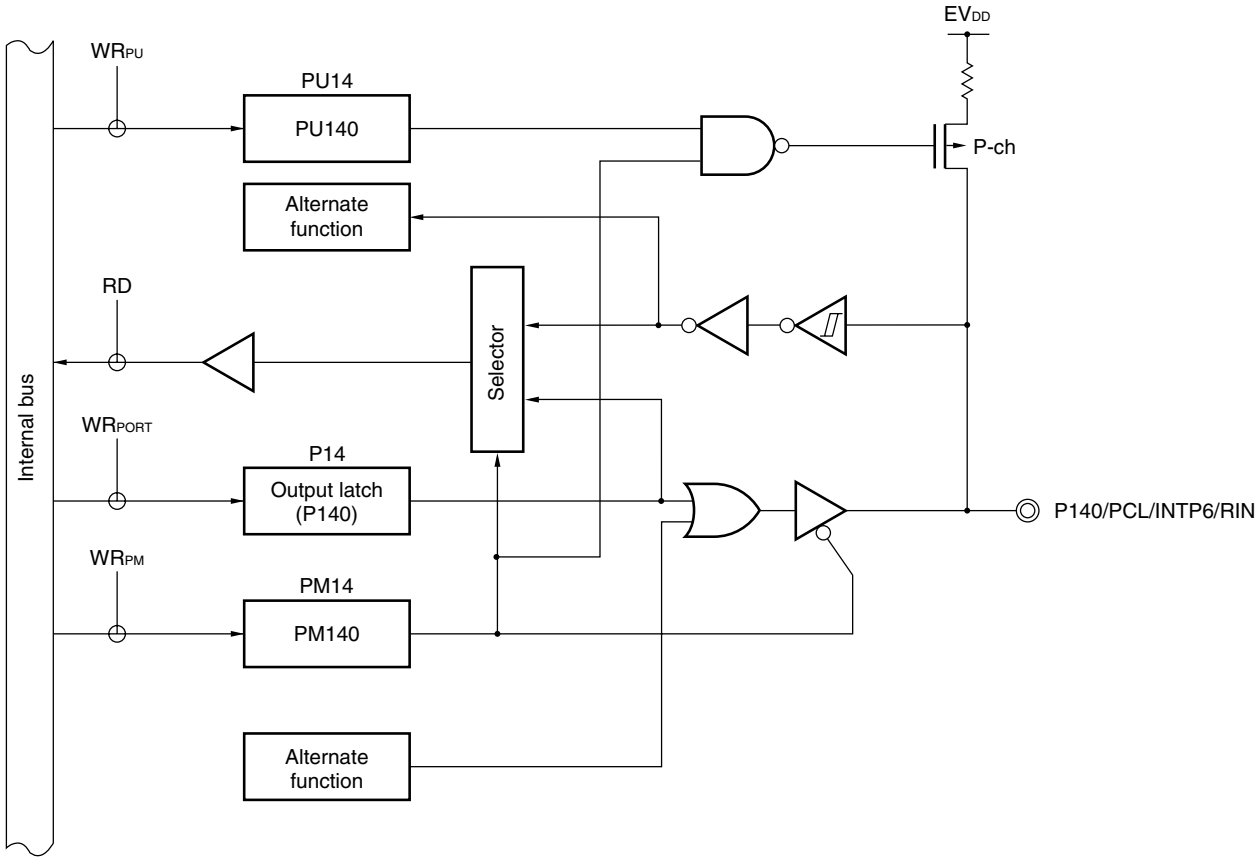
Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 and P141 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register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

This port can also be used for external interrupt request input, buzzer output, clock output, and remote control receive data input.

Reset signal generation sets port 14 to input mode.

Figures 4-29 and 4-30 shows a block diagram of port 14.

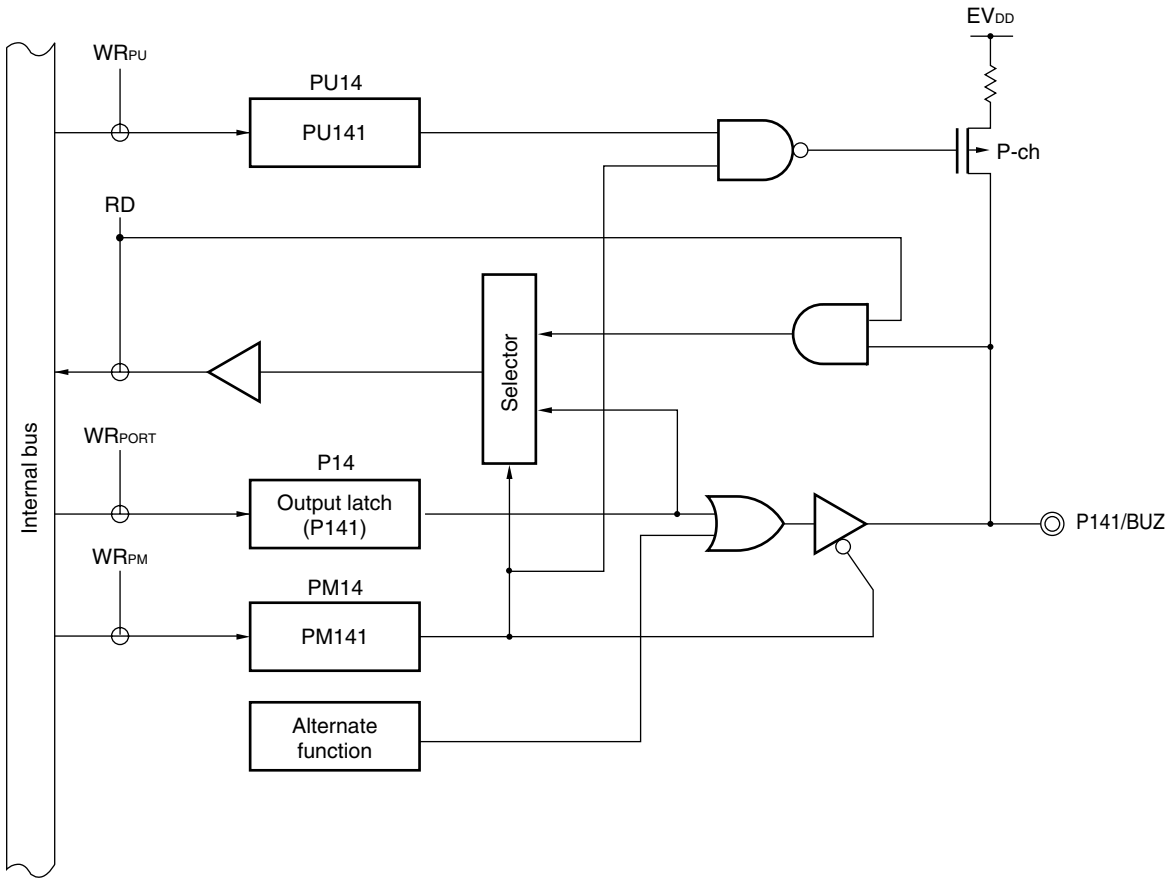
Figure 4-29. Block Diagram of P140



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

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Figure 4-30. Block Diagram of P141



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

Remark With products not provided with an EVDD or EVSS pin, replace EVDD with VDD, or replace EVSS with VSS.

4.3 Registers Controlling Port Function

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

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- 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 4-31. Format of Port Mode Register (78K0/KC2-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FF24H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FF2CH	FFH	R/W
PM14	1	1	1	1	1	1	1	PM140	FF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 4, 6, 7, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 2 to 7 of PM0, bits 4 to 7 of PM3, bits 2 to 7 of PM4, bits 4 to 7 of PM6, bits 6 and 7 of PM7, bits 1 to 7 of PM12, and bits 1 to 7 of PM14 to "1".

Figure 4-32. Format of Port Mode Register (78K0/KE2-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM5	1	1	1	1	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FF2CH	FFH	R/W
PM14	1	1	1	1	1	1	PM141	PM140	FF2EH	FFH	R/W

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

Caution Be sure to set bit 7 of PM0, bits 4 to 7 of PM3, bits 4 to 7 of PM4, bits 4 to 7 of PM5, bits 4 to 7 of PM6, bits 1 to 7 of PM12, and bits 2 to 7 of PM14 to "1".

(2) Port registers (Pxx)

These registers write the data that is output from the chip when data is output from a port.

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

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

Reset signal generation clears these registers to 00H.

Figure 4-33. Format of Port Register (78K0/KC2-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	P01	P00	FF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	00H (output latch)	R/W
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FF04H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FF06H	00H (output latch)	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FF07H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FF0CH	Undefined	R/W Note
P13	0	0	0	0	0	0	0	P130	FF0DH	00H (output latch)	R/W
P14	0	0	0	0	0	0	0	P140	FF0EH	00H (output latch)	R/W

Pmn	m = 0 to 4, 6, 7, 12 to 14; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note P121 to P124 are read-only.

Figure 4-34. Format of Port Register (78K0/KE2-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	P06	P05	P04	P03	P02	P01	P00	FF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	00H (output latch)	R/W
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
P4	0	0	0	0	P43	P42	P41	P40	FF04H	00H (output latch)	R/W
P5	0	0	0	0	P53	P52	P51	P50	FF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FF07H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FF0CH	Undefined	R/W Note
P13	0	0	0	0	0	0	0	P130	FF0DH	00H (output latch)	R/W
P14	0	0	0	0	0	0	P141	P140	FF0EH	00H (output latch)	R/W

Pmn	m = 0 to 7, 12 to 14; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note P121 to P124 are read-only.

(3) Pull-up resistor option registers (PUxx)

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

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

Reset signal generation clears these registers to 00H.

Figure 4-35. Format of Pull-up Resistor Option Register (78K0/KC2-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	FF34H	00H	R/W
PU7	0	0	0	0	0	PU72	0	0	FF37H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
PU14	0	0	0	0	0	0	0	PU140	FF3EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3, 4, 7, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

<R>

Figure 4-36. Relationship Between PU7 Register and PF7 Register

PU72	PF72	Whether to use a diode connection
0	0	Does not connect pull-up resistor to diode
1	0	
0	1	
1	1	Connects pull-up resistor to diode

Caution CECIO and CECIN/CECOUT pins must not be used at the same time.
Do not set PF40 and PF41 to 1 while PF72 = 1.
Do not set PF72 to 1 while PF40 and PF41 = 1.

Figure 4-37. Format of Pull-up Resistor Option Register (78K0/KE2-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU40	FF34H	00H	R/W
PU5	0	0	0	0	PU53	PU52	PU51	PU50	FF35H	00H	R/W
PU7	0	0	0	0	0	PU72	0	0	FF37H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
PU14	0	0	0	0	0	0	PU141	PU140	FF3EH	00H	R/W

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

<R> **Figure 4-38. Relationship Between PU7 Register and PF7 Register**

PU72	PF72	Whether to use a diode connection
0	0	Does not connect pull-up resistor to diode
1	0	
0	1	
1	1	Connects pull-up resistor to diode

Caution CECIO and CECIN/CECOUT pins must not be used at the same time.
Do not set PF40 and PF41 to 1 while PF72 = 1.
Do not set PF72 to 1 while PF40 and PF41 = 1.

(4) Port function register 4 (PF4)

This register sets whether to use pin P40/CECOUT/RTC1HZ as I/O port mode or real-time counter correction clock output/CECOUT mode.

This register sets whether to use pin P41/CECIN/RTCDIV/RTCCL as I/O port mode or real-time counter clock output/CECIN mode.

Figure 4-39. Format of Port Function Register 4 (PF4)

Address: FF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF4	0	0	0	0	0	0	PF41	PF40

PF40	P40 operation mode specification
0	Used as I/O port mode or real-time counter correction clock output
1	Used as CECOUT mode

PF41	P41 operation mode specification
0	Used as I/O port mode or real-time counter clock output
1	Used as CECIN mode

Caution CECIO and CECIN/CECOUT pins must not be used at the same time.
Do not set PF40 and PF41 to 1 while PF72 = 1.
Do not set PF72 to 1 while PF40 and PF41 = 1.

(5) Port function register 7 (PF7)

This register sets whether to use pin P72 as I/O port mode or key interrupt input, or CECIO mode.

Input to the P72 pin can be specified through a normal input buffer or a CEC input buffer, using port function register 7 (PF7) and port input mode register 7 (PIM7).

Whether to connect a diode can be set by setting the PF7 register and pull-up resistor option register 7 (PU7).

Figure 4-40. Format of Port Function Register 7 (PF7)

Address: FF36H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF7	0	0	0	0	0	PF72	0	0

PF72	P72 operation mode specification
0	Used as I/O port mode or key interrupt input
1	Used as CECIO mode

Figure 4-41. Relationship Between PF7 Register, PIM7 Register, and PU7 Register

PF72	PIM72	Input buffer specification
0	0	Normal input buffer
0	1	
1	0	
1	1	CEC input buffer

PF72	PU72	Whether to use a diode connection
0	0	Does not connect pull-up resistor to diode
0	1	
1	0	
1	1	Connects pull-up resistor to diode

Cautions 1. CECIO and CECIN/CECOUT pins must not be used at the same time.
Do not set PF40 and PF41 to 1 while PF72 = 1.
Do not set PF72 to 1 while PF40 and PF41 = 1.
2. When PF72 = 1 is set, set PM72 = 0 and P72 = 1.

(6) Port input mode register (PIM7)

Input to the P72 pin can be specified through a normal input buffer or a CEC input buffer, using port input mode register 7 (PIM7) and port function register 7 (PF7).

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

Reset signal generation clears these registers to 00H.

Figure 4-42. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM7	0	0	0	0	0	PIM72	0	0	FF2DH	00H	R/W

Figure 4-43. Relationship Between PIM7 Register and PF7 Register

PIM72	PF72	Input buffer specification
0	0	Normal input buffer
0	1	
1	0	
1	1	CEC input buffer

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

This register switches the P20/ANI0 to 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 clears this register to 00H.

Figure 4-44. Format of A/D Port Configuration Register (ADPC)

Address: FF2FH After reset: 00H R/W

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

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching							
				ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than the above				Setting prohibited							

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

2. If data is written to ADPC, a wait cycle is generated. Do not write data to ADPC when the peripheral hardware clock is stopped.

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.

4.5 Settings of PF4, PF7, Port Mode Register, and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the PF4, PF7, port mode register, and output latch as shown in Table 4-6.

Remark The port pins mounted depend on the product. See **Table 4-3. Port Functions**.

Table 4-6. Settings of Port PF4, PF7, Mode Register, and Output Latch When Using Alternate Function (1/3)

Pin Name	Alternate Function		PF4	PF7	PM××	P××
	Function Name	I/O				
P00	TI000	Input			1	×
P01	TI010	Input			1	×
	TO00	Output			0	0
P05	TI001	Input			1	×
P06	TI011	Input			1	×
	TO01	Output			0	0
P10	$\overline{SCK10}$	Input			1	×
		Output			0	1
	TxD0	Output			0	1
P11	SI10	Input			1	×
	RxD0	Input			1	×
P12	SO10	Output			0	0
P13	TxD60	Output			0	1
P14	RxD60	Input			1	×
P15	TOH0	Output			0	0
P16	TOH1	Output			0	0
	INTP5	Input			1	×
P17	TI50	Input			1	×
	TO50	Output			0	0
P20 to P27 Note	ANI0 to ANI7 Note	Input			1	×

Note The function of the ANI0/P20 to ANI7/P27 pins can be selected by using the A/D port configuration register (ADPC), the analog input channel specification register (ADS), and PM2.

ADPC	PM2	ADS	ANI0/P20 to ANI7/P27 Pins
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.	
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output

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

Table 4-6. Settings of Port PF4, PF7, Mode Register, and Output Latch When Using Alternate Function (2/3)

Pin Name	Alternate Function		PF4	PF7	PM××	P××
	Function Name	I/O				
P30	INTP1	Input			1	×
P31	INTP2	Input			1	×
P32	ROUT	Output			0	0
	INTP3	Input			1	×
P33	TI51	Input			1	×
	TO51	Output			0	0
	INTP4	Input			1	×
P40	CECOUT	Output	PF40 = 1	PF72 = 0	0	×
	RTC1HZ	Output	PF40 = 0		0	0
P41	CECIN	Input	PF41 = 1	PF72 = 0	1	×
	RTCDIV	Output	PF41 = 0		0	0
	RTCCL	Output	PF41 = 0		0	0
P42	TxD61	Output			0	1
P43	RxD61	Input			1	×
P50	TI002	Input			1	×
P51	TI012	Input			1	×
	TO02	Output			0	0
P60	SCLA0	I/O			0	1
P61	SDAA0	I/O			0	1
P62	SCLA1	I/O			0	1
P63	SDAA1	I/O			0	1

Remarks 1. ×: Don't care
PM××: Port mode register
P××: Port output latch

2. X1, X2, P31, and P32 can be used as on-chip debug mode setting pins (OCD0A, OCD0B, OCD1A, and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 28 ON-CHIP DEBUG FUNCTION.**

Table 4-6. Settings of Port PF4, PF7, Mode Register, and Output Latch When Using Alternate Function (3/3)

Pin Name	Alternate Function		PF4	PF7	PM××	P××
	Function Name	I/O				
P70	SCLA2	I/O			0	1
	KR0	Input			1	×
P71	SDAA2	I/O			0	1
	KR1	Input			1	×
P72	CECIO	I/O	PF40 = 0, PF41 = 0	PF72 = 1	0	×
	KR2	Input		PF72 = 0	1	×
P73	SI11	Input			1	×
	KR3	Input			1	×
P74	SO11	Output			0	0
	KR4 Note 1	Input			1	×
P75	$\overline{\text{SCK11}}$	Input			1	×
		Output			0	1
	KR5 Note 1	Input			1	×
P76	KR6	Input			1	×
	$\overline{\text{SSI11}}$	Input			1	×
P77	KR7	Input			1	×
P120	INTP0	Input			1	×
	EXLVI	Input			1	×
P121	X1 Note 2	–			×	×
P122	X2 Note 2	–			×	×
	EXCLK Note 2	Input			×	×
P123	XT1 Note 2	–			×	×
P124	XT2 Note 2	–			×	×
P140	PCL	Output			0	0
	INTP6	Input			1	×
	RIN	Input			1	×
P141	BUZ	Output			0	0

Notes 1. The 78K0/KC2-C is not provided with the KR4 and KR5 pins.

2. When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2) or subsystem clock (XT1, XT2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, see **5.3 (1) Clock operation mode select register (OSCCTL)** and **(3) Setting of operation mode for subsystem clock pin**). The reset value of OSCCTL is 00H (all of the P121 to P124 are input port pins). At this time, setting of P121 to P124 is not necessary.

Remarks 1. ×: Don't care
PM××: Port mode register
P××: Port output latch

2. X1, X2, P31, and P32 can be used as on-chip debug mode setting pins (OCD0A, OCD0B, OCD1A, and OCD1B) when the on-chip debug function is used. For how to connect an on-chip debug emulator (QB-MINI2), see **CHAPTER 28 ON-CHIP DEBUG FUNCTION.**

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

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

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

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

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

A 1-bit manipulation instruction is executed in the following order in the 78K0/Kx2-C microcontrollers.

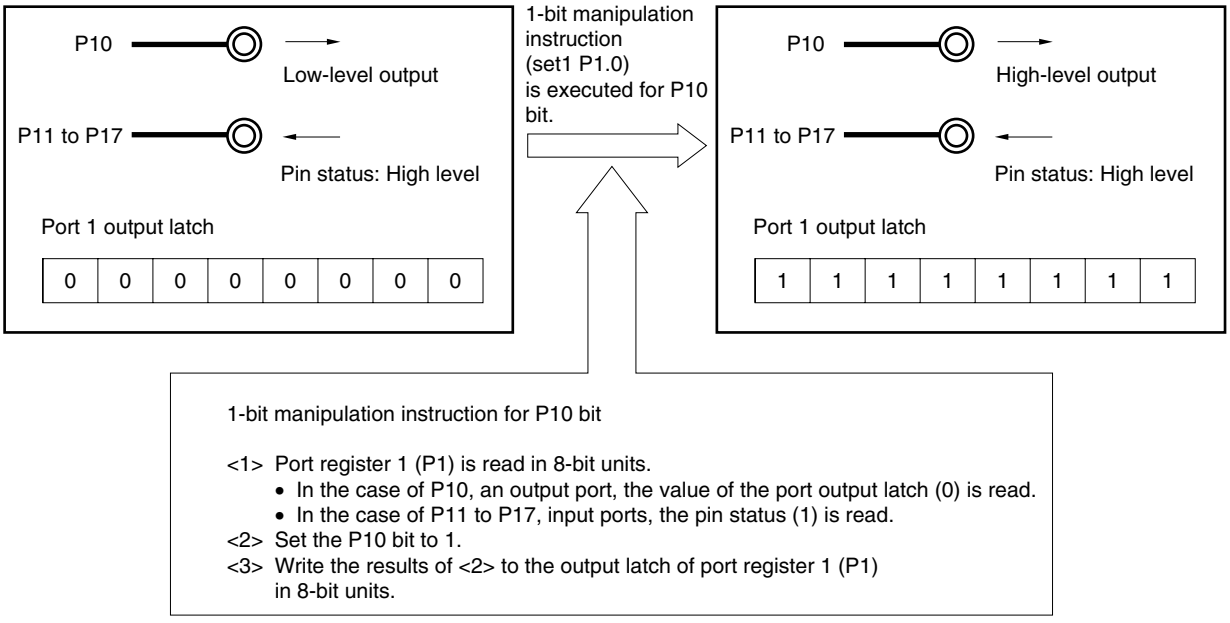
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.

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

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

Figure 4-45. 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 using the main OSC control register (MOC).

<2> Internal high-speed oscillator

This circuit oscillates a clock of f_{RH} = 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 using the internal oscillation mode register (RCM).

An external main system clock (f_{EXCLK} = 2 to 20 MHz) can also be supplied from the OCD0B/EXCLK/X2/P122 pin.
An external main system clock input can be disabled by executing the STOP instruction or using RCM.
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 using the main clock mode register (MCM).

(2) Subsystem clock

- **Subsystem clock oscillator**

  This circuit oscillates at a frequency of f_{XT} = 32.768 kHz by connecting a 32.768 kHz resonator across XT1 and XT2. Oscillation can be stopped by using the processor clock control register (PCC) and clock operation mode select register (OSCCTL).

Remarks 1. f_X: X1 clock oscillation frequency
2. f_{RH}: Internal high-speed oscillation clock frequency
3. f_{EXCLK}: External main system clock frequency
4. f_{XT}: XT1 clock oscillation frequency

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

- **Internal low-speed oscillator**

  This circuit oscillates a clock of f_{RL} = 240 kHz (TYP.). After a reset release, the internal low-speed oscillation clock always starts operating.

  Oscillation can be stopped by using the internal oscillation mode register (RCM) when "internal low-speed oscillator can be stopped by software" is set by option byte.

  The internal low-speed oscillation clock cannot be used as the CPU clock. The following hardware operates with the internal low-speed oscillation clock.

  - Watchdog timer
  - 8-bit timer H1 (if f_{RL}, $f_{RL}/2^7$ or $f_{RL}/2^9$ is selected as the count clock)

Remark f_{RL}: Internal low-speed oscillation clock frequency

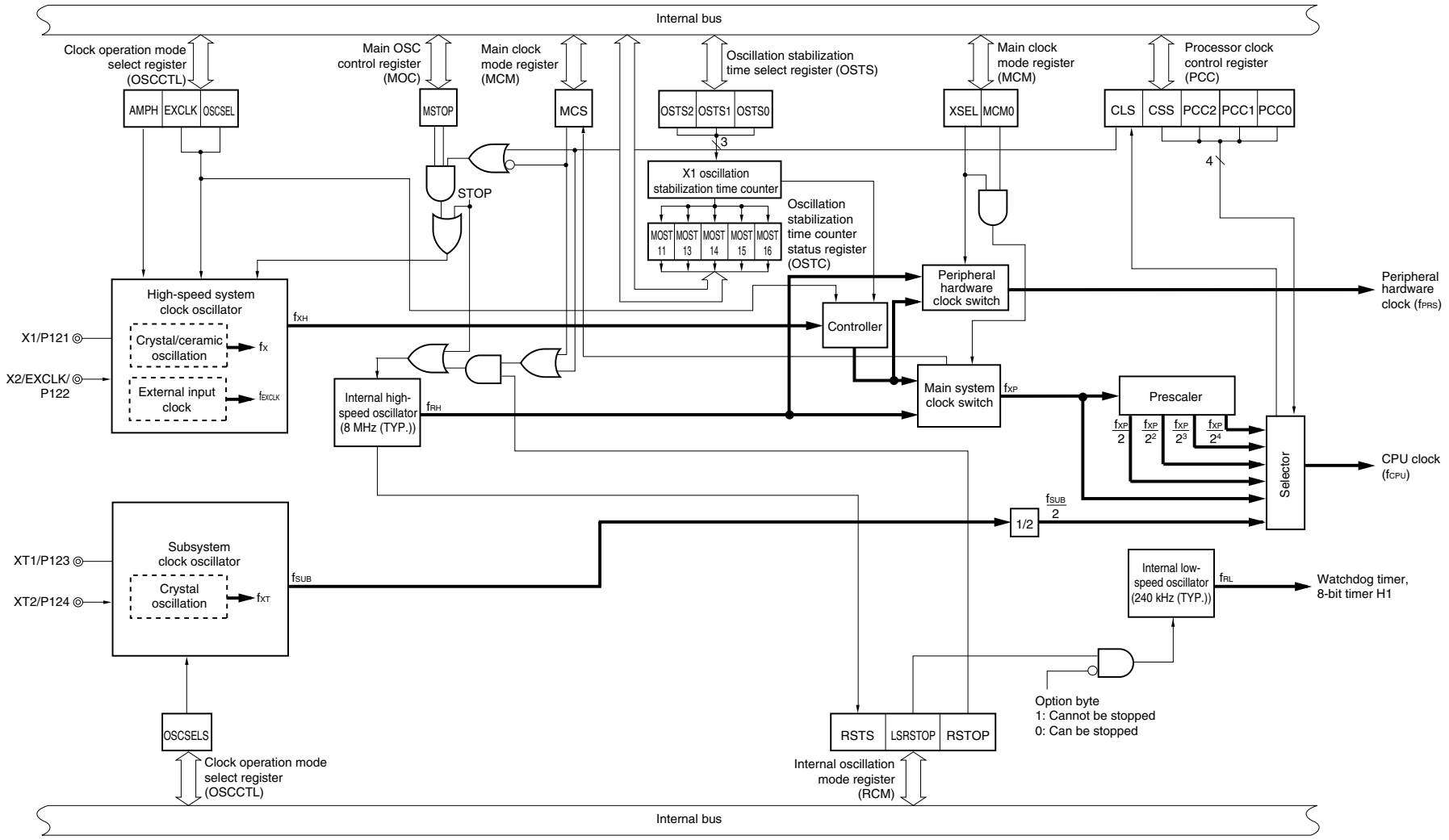
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode select register (OSCCTL) Processor clock control register (PCC) Internal oscillation mode register (RCM) Main OSC control register (MOC) Main clock mode register (MCM) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



Remarks 1. f_X: X1 clock oscillation frequency
2. f_{RH}: Internal high-speed oscillation clock frequency
3. f_{EXCLK}: External main system clock frequency
4. f_{XH}: High-speed system clock frequency
5. f_{XP}: Main system clock frequency
6. f_{PRS}: Peripheral hardware clock frequency
7. f_{CPU}: CPU clock frequency
8. f_{XT}: XT1 clock oscillation frequency
9. f_{SUB}: Subsystem clock frequency
10. f_{RL}: Internal low-speed oscillation clock frequency

5.3 Registers Controlling Clock Generator

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

- Clock operation mode select register (OSCCTL)
- Processor clock control register (PCC)
- Internal oscillation mode register (RCM)
- Main OSC control register (MOC)
- Main clock mode register (MCM)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

(1) Clock operation mode select register (OSCCTL)

This register selects the operation modes of the high-speed system and subsystem clocks, and the gain of the on-chip oscillator.

OSCCTL can be set 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 Select Register (OSCCTL)

Address: FF9FH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
OSCCTL	EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH

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

AMPH	Operating frequency control
0	1 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

Cautions 1. Be sure to set AMPH to 1 if the high-speed system clock oscillation frequency exceeds 10 MHz.

2. Set AMPH before setting the main system mode register (MCM).

3. Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. When the high-speed system clock (X1 oscillation) is selected as the CPU clock, supply of the CPU clock is stopped for 4.06 to 16.12 μs after AMPH is set to 1. When the high-speed system clock (external clock input) is selected as the CPU clock, supply of the CPU clock is stopped for the duration of 160 external clocks after AMPH is set to 1.

4. If the STOP instruction is executed when AMPH = 1, supply of the CPU clock is stopped for 4.06 to 16.12 μs after the STOP mode is released when the internal high-speed oscillation clock is selected as the CPU clock, or for the duration of 160 external clocks when the high-speed system clock (external clock input) is selected as the CPU clock. When the high-speed system clock (X1 oscillation) is selected as the CPU clock, the oscillation stabilization time is counted after the STOP mode is released.

5. To change the value of EXCLK and OSCSEL, be sure to confirm that bit 7 (MSTOP) of the main OSC control register (MOC) is 1 (the X1 oscillator stops or the external clock from the EXCLK pin is disabled).

6. Be sure to clear bits 1 to 3, and 5 to "0".

Remark f_{XH}: High-speed system clock oscillation frequency

(2) Processor clock control register (PCC)

This register is used to select the CPU clock, the division ratio, and operation mode for subsystem clock.

PCC is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PCC to 01H.

Figure 5-3. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 01H R/W[Note]

Symbol	7	6	<5>	<4>	3	2	1	0
PCC	0	0	CLS	CSS	0	PCC2	PCC1	PCC0

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

CSS	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
0	0	0	0	f_{XP}
	0	0	1	f_{XP}/2 (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
1	0	0	0	f_{SUB}/2
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

Note Bit 5 is read-only.

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

Remarks 1. f_{XP}: Main system clock oscillation frequency

2. f_{SUB}: Subsystem clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/Kx2-C. Therefore, the relationship between the CPU clock (f_{CPU}) 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 (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$			
	Main System Clock			Subsystem Clock
	High-Speed System Clock[Note]		Internal High-Speed Oscillation Clock[Note]	
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{XP}	0.2 μs	0.1 μs	0.25 μs (TYP.)	–
$f_{XP}/2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	–
$f_{XP}/2^2$	0.8 μs	0.4 μs	1.0 μs (TYP.)	–
$f_{XP}/2^3$	1.6 μs	0.8 μs	2.0 μs (TYP.)	–
$f_{XP}/2^4$	3.2 μs	1.6 μs	4.0 μs (TYP.)	–
$f_{SUB}/2$	–		–	122.1 μs

Note The main clock mode register (MCM) is used to set the main system clock supplied to CPU clock (high-speed system clock/internal high-speed oscillation clock) (see **Figure 5-6**).

(3) Setting of operation mode for subsystem clock pin

The operation mode for the subsystem clock pin can be set by using bit 4 (OSCSELS) of the clock operation mode select register (OSCCTL) in combination.

Table 5-3. Setting of Operation Mode for Subsystem Clock Pin

Bit 4 of OSCCTL OSCSELS	Subsystem Clock Pin Operation Mode	P123/XT1 Pin	P124/XT2 Pin
0	Input port mode	Input port	
1	XT1 oscillation mode	Crystal resonator connection	

Caution Confirm that bit 5 (CLS) of the processor clock control register (PCC) is 0 (CPU is operating with main system clock) when changing the current values of OSCSELS.

(4) Internal oscillation mode register (RCM)

This register sets the operation mode of internal oscillator.

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

Reset signal generation sets this register to 80H[Note 1].

Figure 5-4. Format of Internal Oscillation Mode Register (RCM)

Address: FFA0H After reset: 80H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	Status of internal high-speed oscillator
0	Waiting for accuracy stabilization of internal high-speed oscillator
1	Stability operating of internal high-speed oscillator

LSRSTOP	Internal low-speed oscillator oscillating/stopped
0	Internal low-speed oscillator oscillating
1	Internal low-speed oscillator stopped

RSTOP	Internal high-speed oscillator oscillating/stopped
0	Internal high-speed oscillator oscillating
1	Internal high-speed oscillator stopped

Notes 1. The value of this register is 00H immediately after a reset release but automatically changes to 80H after internal high-speed oscillator has been stabilized.

2. Bit 7 is read-only.

Caution When setting RSTOP to 1, be sure to confirm that the CPU operates with a clock other than the internal high-speed oscillation clock. Specifically, set under either of the following conditions.

- **When MCS = 1 (when CPU operates with the high-speed system clock)**
- **When CLS = 1 (when CPU operates with the subsystem clock)**

In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock before setting RSTOP to 1.

(5) Main OSC control register (MOC)

This register selects the operation mode of the high-speed system clock.

This register is used to stop the X1 oscillator or to disable an external clock input from the EXCLK pin when the CPU operates with a clock other than the high-speed system clock.

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

Reset signal generation sets this register to 80H.

Figure 5-5. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 80H R/W

Symbol	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	Control of high-speed system clock operation	
	X1 oscillation mode	External clock input mode
0	X1 oscillator operating	External clock from EXCLK pin is enabled
1	X1 oscillator stopped	External clock from EXCLK pin is disabled

Cautions 1. When setting MSTOP to 1, be sure to confirm that the CPU operates with a clock other than the high-speed system clock. Specifically, set under either of the following conditions.

- **When MCS = 0 (when CPU operates with the internal high-speed oscillation clock)**
- **When CLS = 1 (when CPU operates with the subsystem clock)**

In addition, stop peripheral hardware that is operating on the high-speed system clock before setting MSTOP to 1.

2. Do not clear MSTOP to 0 while bit 6 (OSCSEL) of the clock operation mode select register (OSCCTL) is 0 (I/O port mode).

3. The peripheral hardware cannot operate when the peripheral hardware clock is stopped. To resume the operation of the peripheral hardware after the peripheral hardware clock has been stopped, initialize the peripheral hardware.

(6) Main clock mode register (MCM)

This register selects the main system clock supplied to CPU clock and clock supplied to peripheral hardware clock.

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

Reset signal generation clears this register to 00H.

Figure 5-6. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W[Note]

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

XSEL	MCM0	Selection of clock supplied to main system clock and peripheral hardware	
		Main system clock (f_{XP})	Peripheral hardware clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})
1	1	High-speed system clock (f_{XH})	

MCS	Main system clock status
0	Operates with internal high-speed oscillation clock
1	Operates with high-speed system clock

Note Bit 1 is read-only.

Cautions 1. XSEL can be changed only once after a reset release.

2. A clock other than f_{PRS} is supplied to the following peripheral functions regardless of the setting of XSEL and MCM0.

- **Watchdog timer (operates with internal low-speed oscillation clock)**
- **When "f_{RL}", "$f_{RL}/2^7$", or "$f_{RL}/2^9$" is selected as the count clock for 8-bit timer H1 (operates with internal low-speed oscillation clock)**
- **Peripheral hardware selects the external clock as the clock source (Except when the external count clock of TM00 is selected (TI000 pin valid edge))**

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

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter. When X1 clock oscillation starts with the internal high-speed oscillation clock or subsystem clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

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, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 5-7. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status		
						f_X = 10 MHz	f_X = 20 MHz
1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	0	0	$2^{14}/f_X$ min.	1.64 ms min.	819.2 μs min.
1	1	1	1	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_X$ min.	6.55 ms min.	3.27 ms min.

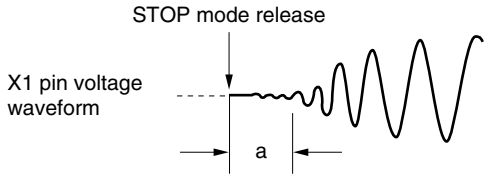
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

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



Remark f_X: X1 clock oscillation frequency

(8) 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 OSTS to 05H.

Figure 5-8. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fx = 10 MHz	fx = 20 MHz
0	0	1	2^{11}/fx	204.8 μs	102.4 μs
0	1	0	2^{13}/fx	819.2 μs	409.6 μs
0	1	1	2^{14}/fx	1.64 ms	819.2 μs
1	0	0	2^{15}/fx	3.27 ms	1.64 ms
1	0	1	2^{16}/fx	6.55 ms	3.27 ms
Other than above			Setting prohibited		

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

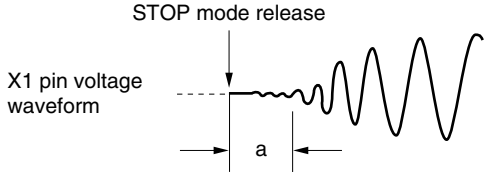
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. 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.

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



Remark fx: X1 clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 X1 oscillator

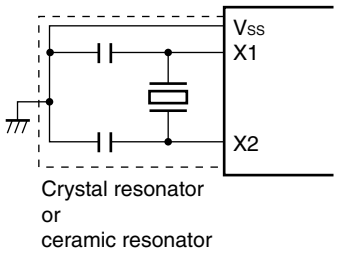
The X1 oscillator oscillates with a crystal resonator or ceramic resonator (2 to 20 MHz) connected to the X1 and X2 pins.

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

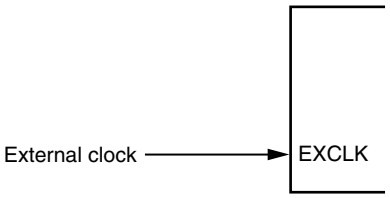
Figure 5-9 shows an example of the external circuit of the X1 oscillator.

Figure 5-9. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



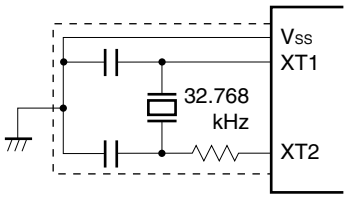
5.4.2 XT1 oscillator

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

Figure 5-10 shows an example of the external circuit of the XT1 oscillator.

Figure 5-10. Example of External Circuit of XT1 Oscillator

(a) Crystal oscillation



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

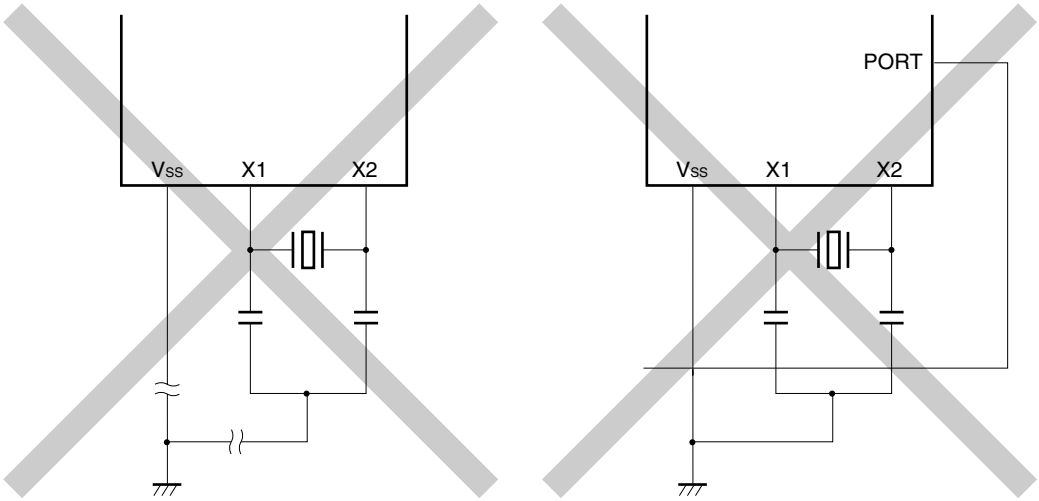
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as VSS. Do not ground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

Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.

Figure 5-11 shows examples of incorrect resonator connection.

Figure 5-11. Examples of Incorrect Resonator Connection (1/2)

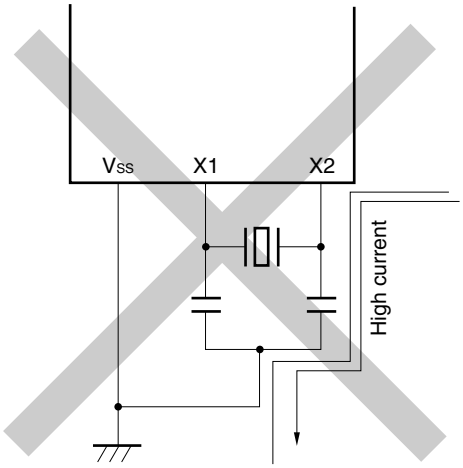
(a) Too long wiring **(b) Crossed signal line**



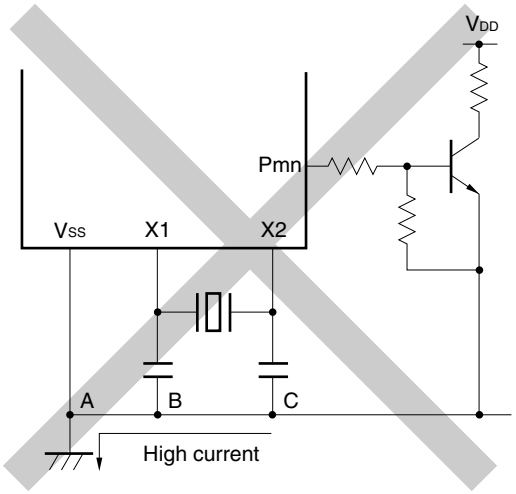
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-11. Examples of Incorrect Resonator Connection (2/2)

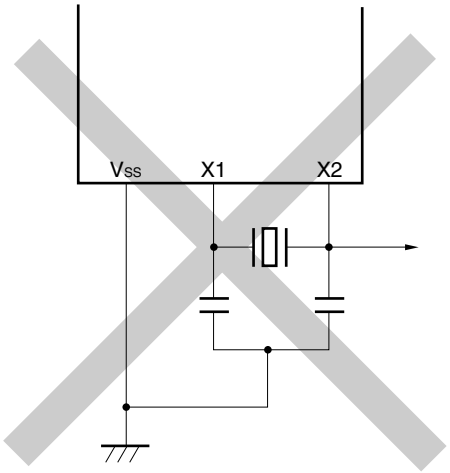
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

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

5.4.3 When subsystem clock is not used

If it is not necessary to use the subsystem clock for low power consumption operations, or if not using the subsystem clock as an I/O port, set the XT1 and XT2 pins to Input port mode (OSCSELS = 0) and independently connect to V_{DD} or V_{SS} via a resistor.

Remark OSCSELS: Bit 4 of clock operation mode select register (OSCCTL)

5.4.4 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0/Kx2-C. Oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal high-speed oscillator automatically starts oscillation (8 MHz (TYP.)).

5.4.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0/Kx2-C.

The internal low-speed oscillation clock is only used as the clock of the watchdog timer and 8-bit timer H1. The internal low-speed oscillation clock cannot be used as the CPU clock.

“Can be stopped by software” or “Cannot be stopped” can be selected by the option byte. When “Can be stopped by software” is set, oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled using the option byte.

5.4.6 Prescaler

The prescaler generates various clocks by dividing the main system clock when the main system clock is selected as the clock to be supplied to the CPU.

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_{XP}
  - High-speed system clock f_{XH}
    X1 clock f_X
    External main system clock f_{EXCLK}
  - Internal high-speed oscillation clock f_{RH}
- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
- Internal low-speed oscillation clock f_{RL}
- CPU clock f_{CPU}
- Peripheral hardware clock f_{PRS}

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the 78K0/Kx2-C, thus enabling the following.

(1) Enhancement of security function

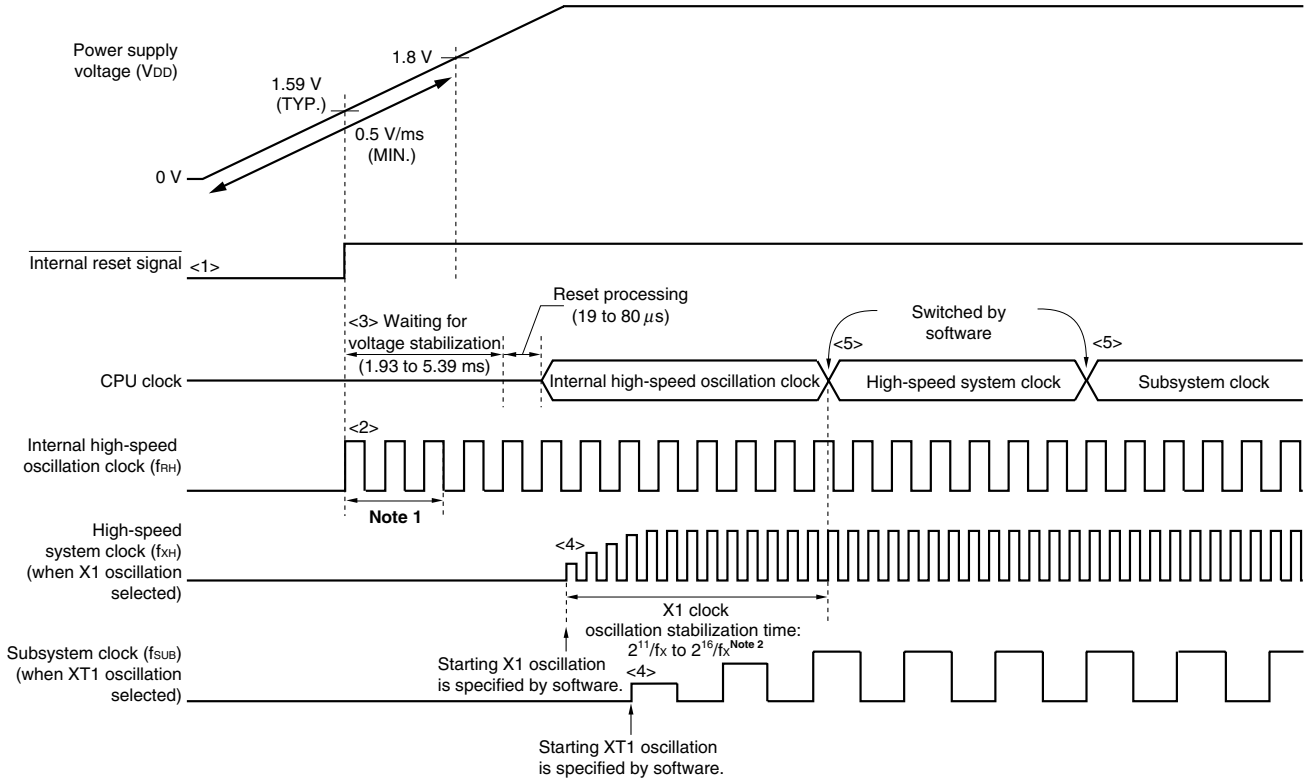
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. Consequently, the system can be safely shut down by performing a minimum operation, such as acknowledging a reset source by software or performing safety processing when there is a malfunction.

(2) Improvement of performance

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

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-12.

Figure 5-12. Clock Generator Operation When Power Supply Voltage Is Turned On (When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0))



<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 1.59 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

<3> When the power supply voltage rises with a slope of 0.5 V/ms (MIN.), the CPU starts operation on the internal high-speed oscillation clock after the reset is released and after the stabilization times for the voltage of the power supply and regulator have elapsed, and then reset processing is performed.

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

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

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

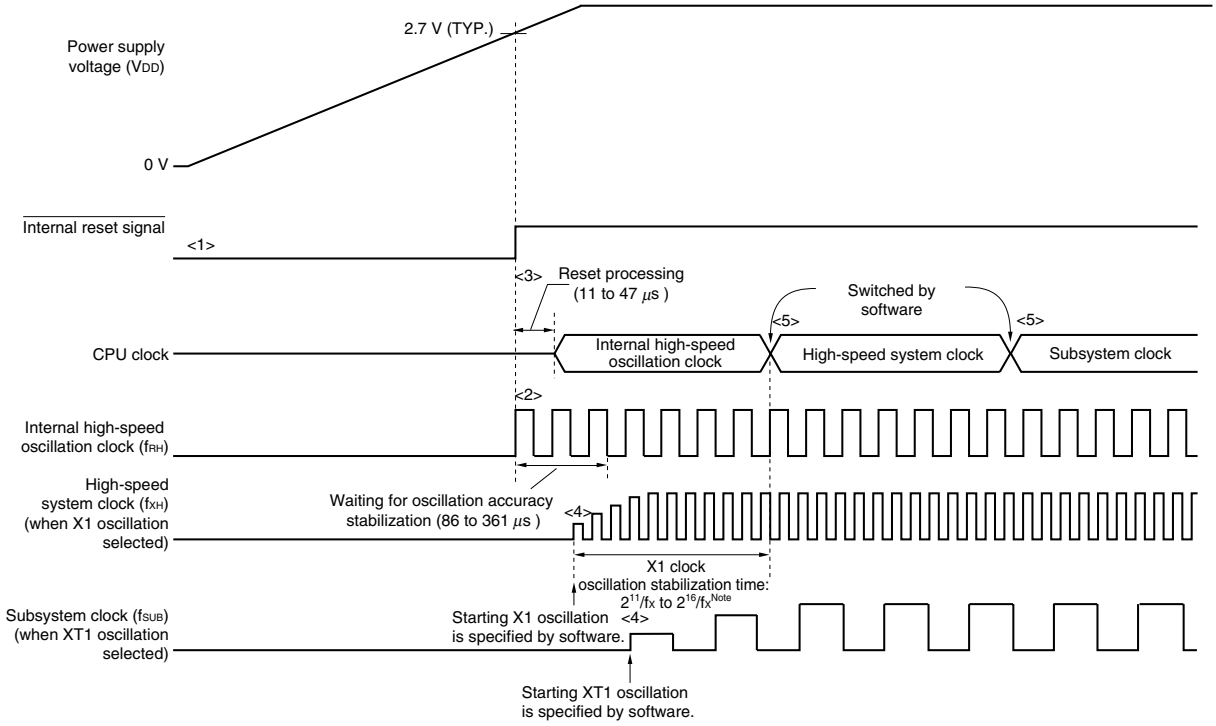
2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

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 2.7 V/1.59 V POC mode by using the option byte (POCMODE = 1) (see Figure 5-13). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-12 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.

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(4)** in **5.6.3 Example of controlling subsystem clock**).

Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When 2.7 V/1.59 V POC Mode Is Set (Option Byte: POCMODE = 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 2.7 V (TYP.), the reset is released and the internal high-speed oscillator 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.

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

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

Note When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

Cautions 1. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the time the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) is within 1.93 to 5.39 ms, a power supply stabilization wait time of 0 to 5.39 ms occurs automatically before reset processing, and the reset processing time becomes 19 to 80 μs.

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

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(4)** in **5.6.3 Example of controlling subsystem clock**).

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 across 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 OCD0A/X1/P121 and OCD0B/X2/EXCLK/P122 pins can be used as I/O port pins.

Caution The OCD0A/X1/P121 and OCD0B/X2/EXCLK/P122 pins are in the I/O 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 clock and peripheral hardware clock
(4) When stopping high-speed system clock

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

<1> Setting frequency (OSCCTL register)
Using AMPH, set the gain of the on-chip oscillator according to the frequency to be used.

AMPH[Note]	Operating Frequency Control
0	1 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

Note Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. When AMPH is set to 1, the clock supply to the CPU is stopped for 4.06 to 16.12 μs.

Remark f_{XH}: High-speed system clock oscillation frequency

<2> Setting P121/X1 and P122/X2/EXCLK pins and selecting X1 clock or external clock (OSCCTL register)
When EXCLK is cleared to 0 and OSCSEL is set to 1, the mode is switched from port mode to X1 oscillation mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	

<3> Controlling oscillation of X1 clock (MOC register)
If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<4> 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. Do not change the value of EXCLK and OSCSEL while the X1 clock is operating.
2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 30 ELECTRICAL SPECIFICATIONS).

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

<1> Setting frequency (OSCCTL register)
Using AMPH, set the frequency to be used.

AMPH[Note]	Operating Frequency Control
0	1 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

Note Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. The clock supply to the CPU is stopped for the duration of 160 external clocks after AMPH is set to 1.

Remark f_{XH}: High-speed system clock oscillation frequency

<2> Setting P121/X1 and P122/X2/EXCLK pins and selecting operation mode (OSCCTL register)
When EXCLK and OSCSEL are set to 1, the mode is switched from port mode to external clock input mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
1	1	External clock input mode	I/O port	External clock input

<3> Controlling external main system clock input (MOC register)
When MSTOP is cleared to 0, the input of the external main system clock is enabled.

Cautions 1. Do not change the value of EXCLK and OSCSEL while the external main system clock is operating.
2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 30 ELECTRICAL SPECIFICATIONS).

(3) Example of setting procedure when using high-speed system clock as CPU clock and 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 main system clock (MCM register)

When XSEL and MCM0 are set to 1, the high-speed system clock is supplied as the main system clock and peripheral hardware clock.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
1	1	High-speed system clock (f_{XH})	High-speed system clock (f_{XH})

Caution If the high-speed system clock is selected as the main system clock, a clock other than the high-speed system clock cannot be set as the peripheral hardware clock.

<3> Setting the main system clock as the CPU clock and selecting the division ratio (PCC register)

When CSS is cleared to 0, the main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_{XP}
	0	0	1	f_{XP}/2 (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
	Other than above			Setting prohibited

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

The high-speed system clock can be stopped in the following two ways.

- Executing the STOP instruction to set the STOP mode
- Setting MSTOP to 1 and stopping the X1 oscillation (disabling clock input if the external clock is used)

(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 22 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<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 (PCC and MCM registers)

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
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the high-speed system clock (MOC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

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 clock, and internal high-speed oscillation clock or high-speed system clock as 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]

<1> Setting restart of oscillation of the internal high-speed oscillation clock (RCM register)

When RSTOP is cleared to 0, the internal high-speed oscillation clock starts operating.

<2> Waiting for the oscillation accuracy stabilization time of internal high-speed oscillation clock (RCM register)

Wait until RSTS is set to 1[Note 2].

Notes 1. After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU clock.

2. This wait time is not necessary if high accuracy is not necessary for the CPU clock and peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as 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**).

• Oscillating the high-speed system clock[Note]
(This setting is required when using the high-speed system clock as the peripheral hardware clock. 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 the internal high-speed oscillation clock or high-speed system clock is already operating.

<2> Selecting the clock supplied as the main system clock and peripheral hardware clock (MCM register)
Set the main system clock and peripheral hardware clock using XSEL and MCM0.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})

<3> Selecting the CPU clock division ratio (PCC register)
When CSS is cleared to 0, the main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_{XP}
	0	0	1	f_{XP}/2 (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
	Other than above			Setting prohibited

(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 to set the STOP mode
- Setting RSTOP to 1 and stopping the internal high-speed oscillation clock

(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 22 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<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 RSTOP to 1

<1> Confirming the CPU clock status (PCC and MCM registers)

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
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the internal high-speed oscillation clock (RCM register)

When RSTOP is set to 1, internal high-speed oscillation clock is stopped.

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting RSTOP 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 following two types of subsystem clocks are available.

- XT1 clock: Crystal/ceramic resonator is connected across 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 XT1 clock
(2) When using subsystem clock as CPU clock
(3) When stopping subsystem clock

(1) Example of setting procedure when oscillating the XT1 clock

<1> Setting XT1 and XT2 pins and selecting operation mode (PCC and OSCCTL registers)
When OSCSELS is set as any of the following, the mode is switched from port mode to XT1 oscillation mode.

OSCSELS	Operation Mode of Subsystem Clock Pin	P123/XT1 Pin	P124/XT2 Pin
1	XT1 oscillation mode	Crystal/ceramic resonator connection	

<2> 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 Do not change the value of OSCSELS while the subsystem clock is operating.

(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 XT1 clock**)

Note The setting of <1> is not necessary when while the subsystem clock is operating.

<2> Switching the CPU clock (PCC register)
When CSS is set to 1, the subsystem clock is supplied to the CPU.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
1	0	0	0	$f_{SUB}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
	Other than above			Setting prohibited

(3) Example of setting procedure when stopping the subsystem clock

<1> Confirming the CPU clock status (PCC and MCM registers)

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
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the subsystem clock (OSCCTL register)

When OSCSELS is cleared to 0, XT1 oscillation is stopped.

Cautions 1. Be sure to confirm that CLS = 0 when clearing OSCSELS to 0. 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.

Only the following peripheral hardware can operate with this clock.

- Watchdog timer
- 8-bit timer H1 (if f_{RL}, $f_{RL}/2^7$ or $f_{RL}/2^9$ is selected as the count clock)

In addition, the following operation modes can be selected by the option byte.

- Internal low-speed oscillator cannot be stopped
- Internal low-speed oscillator can be stopped by software

The internal low-speed oscillator automatically starts oscillation after a reset release, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation has been enabled by the option byte.

(1) Example of setting procedure when stopping the internal low-speed oscillation clock

<1> Setting LSRSTOP to 1 (RCM register)

When LSRSTOP is set to 1, the internal low-speed oscillation clock is stopped.

(2) Example of setting procedure when restarting oscillation of the internal low-speed oscillation clock

<1> Clearing LSRSTOP to 0 (RCM register)

When LSRSTOP is cleared to 0, the internal low-speed oscillation clock is restarted.

Caution If "Internal low-speed oscillator cannot be stopped" is selected by the option byte, oscillation of the internal low-speed oscillation clock cannot be controlled.

5.6.5 Clocks supplied to CPU and peripheral hardware

The following table shows the relation among the clocks supplied to the CPU and peripheral hardware, and setting of registers.

Table 5-4. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting

Supplied Clock		XSEL	CSS	MCM0	EXCLK
Clock Supplied to CPU	Clock Supplied to Peripheral Hardware				
Internal high-speed oscillation clock		0	0	×	×
Internal high-speed oscillation clock	X1 clock	1	0	0	0
	External main system clock	1	0	0	1
X1 clock		1	0	1	0
External main system clock		1	0	1	1
Subsystem clock	Internal high-speed oscillation clock	0	1	×	×
	X1 clock	1	1	0	0
		1	1	1	0
	External main system clock	1	1	0	1
		1	1	1	1

Remarks 1. XSEL: Bit 2 of the main clock mode register (MCM)

2. CSS: Bit 4 of the processor clock control register (PCC)

3. MCM0: Bit 0 of MCM

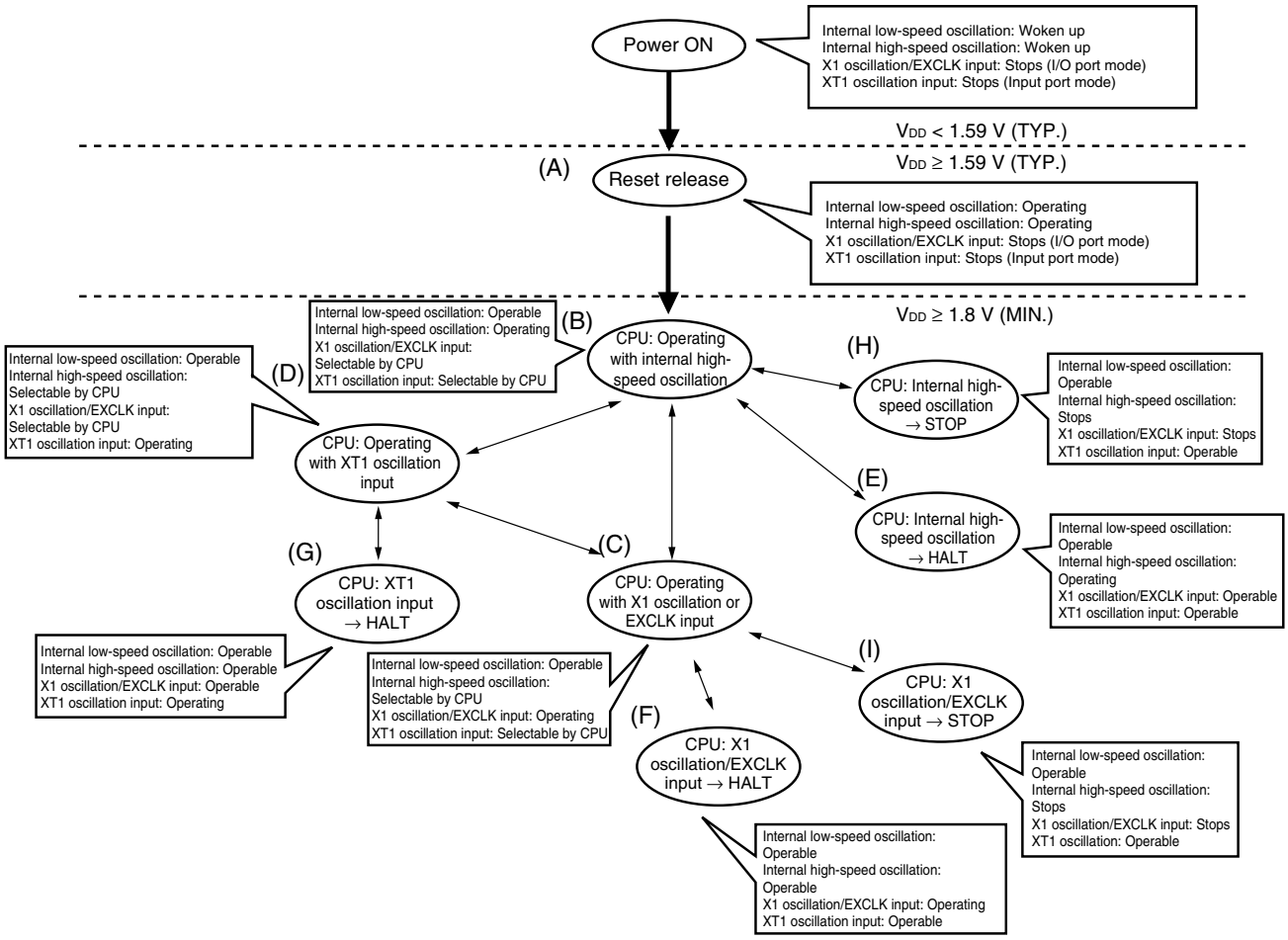
4. EXCLK: Bit 7 of the clock operation mode select register (OSCCTL)

5. ×: don't care

5.6.6 CPU clock status transition diagram

Figure 5-14 shows the CPU clock status transition diagram of this product.

Figure 5-14. CPU Clock Status Transition Diagram
(When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0))



Remark In the 2.7 V/1.59 V POC mode (option byte: POCMODE = 1), the CPU clock status changes to (A) in the above figure when the supply voltage exceeds 2.7 V (TYP.), and to (B) after reset processing (11 to 47 μs (TYP.)).

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

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

(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	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	1	1	0	Must not be checked	1	1
(A) → (B) → (C) (X1 clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	1	1	0	Must not be checked	1	1

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 30 ELECTRICAL SPECIFICATIONS).

(3) CPU operating with subsystem clock (D) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSCSELS	Waiting for Oscillation Stabilization	CSS
(A) → (B) → (D)	1	Necessary	1

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-14.

2. EXCLK, OSCSEL, OSCSELS, AMPH: Bits 7, 6, and 4 of the clock operation mode select register (OSCCTL)

MSTOP: Bit 7 of the main OSC control register (MOC)

XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)

CSS: Bit 4 of the processor clock control register (PCC)

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (2/4)

(4) 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	AMPH[Note]	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0
(B) → (C) (X1 clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	1	1	0	Must not be checked	1	1
(B) → (C) (X1 clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	1	1	0	Must not be checked	1	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 30 ELECTRICAL SPECIFICATIONS).

(5) 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	OSCSELS	Waiting for Oscillation Stabilization	CSS
(B) → (D)	1	Necessary	1

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-14.

2. EXCLK, OSCSEL, OSCSELS, AMPH: Bits 7, 6, and 4 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (3/4)

(6) 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	RSTOP	RSTS	MCM0
(C) → (B)	0	Confirm this flag is 1.	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

(7) 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	OSCSELS	Waiting for Oscillation Stabilization	CSS
(C) → (D)	1	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

(8) 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	RSTOP	RSTS	MCM0	CSS
(D) → (B)	0	Confirm this flag is 1.	0	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

↑

Unnecessary if XSEL is 0

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-14.

2. MCM0: Bit 0 of the main clock mode register (MCM)
OSCSELS: Bit 4 of the clock operation mode select register (OSCCTL)
RSTS, RSTOP: Bits 7 and 0 of the internal oscillation mode register (RCM)
CSS: Bit 4 of the processor clock control register (PCC)

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

(9) 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	AMPH[Note]	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0	CSS
(D) → (C) (X1 clock: 1 MHz ≤ f_{XH} ≤ 10 MHz)	0	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: 1 MHz ≤ f_{XH} ≤ 10 MHz	0	1	1	0	Must not be checked	1	1	0
(D) → (C) (X1 clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: 10 MHz < f_{XH} ≤ 20 MHz)	1	1	1	0	Must not be checked	1	1	0

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Unnecessary if this register is already set

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 30 ELECTRICAL SPECIFICATIONS).

(10) • 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)

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

(11) • 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) (C) → (I)	Stopping peripheral functions that cannot operate in STOP mode	Executing STOP instruction

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-14.

2. EXCLK, OSCSEL, AMPH: Bits 7 and 6 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

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 5-6. Changing CPU Clock

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin • MSTOP = 0, OSCSEL = 1, EXCLK = 1	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 1.
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • RSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
Internal high-speed oscillation clock	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (RSTOP = 1).
X1 clock			X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
XT1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • RSTOP = 0, MCS = 0	XT1 oscillation can be stopped (OSCSELS = 0).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time • MCS = 1	• XT1 oscillation can be stopped (OSCSELS = 0). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 1 • MCS = 1	• XT1 oscillation can be stopped (OSCSELS = 0). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 1.

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

By setting bits 0 to 2 (PCC0 to PCC2) and bit 4 (CSS) of the processor clock control register (PCC), the CPU clock can be switched (between the main system clock and the subsystem clock) and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to PCC; operation continues on the pre-switchover clock for several clocks (see **Table 5-7**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 5 (CLS) of the PCC register.

Table 5-7. Time Required for Switchover of CPU Clock and Main System Clock Cycle Division Factor

Set Value Before Switchover				Set Value After Switchover																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
0	0	0	0					16 clocks				16 clocks				16 clocks				16 clocks				$2f_{XP}/f_{SUB}$ clocks			
	0	0	1	8 clocks								8 clocks				8 clocks				8 clocks				f_{XP}/f_{SUB} clocks			
	0	1	0	4 clocks				4 clocks								4 clocks				4 clocks				$f_{XP}/2f_{SUB}$ clocks			
	0	1	1	2 clocks				2 clocks				2 clocks								2 clocks				$f_{XP}/4f_{SUB}$ clocks			
	1	0	0	1 clock				1 clock				1 clock				1 clock								$f_{XP}/8f_{SUB}$ clocks			
1	×	×	×	2 clocks				2 clocks				2 clocks				2 clocks				2 clocks							

Caution Selection of the main system clock cycle division factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously.

Simultaneous setting is possible, however, for selection of the main system clock cycle division factor (PCC0 to PCC2) and switchover from the subsystem clock to the main system clock (changing CSS from 1 to 0).

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

2. When switching the CPU clock from the main system clock to the subsystem clock, calculate the number of clocks by rounding up to the next clock and discarding the decimal portion, as shown below.

Example When switching CPU clock from $f_{XP}/2$ to $f_{SUB}/2$ (@ oscillation with f_{XP} = 10 MHz, f_{SUB} = 32.768 kHz)

f_{XP}/f_{SUB} = 10000/32.768 ≅ 305.1 → 306 clocks

By setting bit 0 (MCM0) of the main clock mode register (MCM), the main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to MCM0; operation continues on the pre-switchover clock for several clocks (see **Table 5-8**).

Whether the CPU is operating on the internal high-speed oscillation clock or the high-speed system clock can be ascertained using bit 1 (MCS) of MCM.

Table 5-8. Maximum Time Required for Main System Clock Switchover

Set Value Before Switchover	Set Value After Switchover	
MCM0	MCM0	
	0	1
0		1 + $2f_{RH}/f_{XH}$ clock
1	1 + $2f_{XH}/f_{RH}$ clock	

Caution When switching the internal high-speed oscillation clock to the high-speed system clock, bit 2 (XSEL) of MCM must be set to 1 in advance. The value of XSEL can be changed only once after a reset release.

Remarks 1. The number of clocks listed in Table 5-8 is the number of main system clocks before switchover.

2. Calculate the number of clocks in Table 5-8 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_{RH} = 8 MHz, f_{XH} = 10 MHz)

$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2$ 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 5-9. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
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)	RSTOP = 1
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
XT1 clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock)	OSCSELS = 0

5.6.10 Peripheral hardware and source clocks

The following lists peripheral hardware and source clocks incorporated in the 78K0/Kx2-C.

Table 5-10. Peripheral Hardware and Source Clocks

Peripheral Hardware \ Source Clock		Peripheral Hardware Clock (f_{PRS})	Subsystem Clock (f_{SUB})	Internal Low-Speed Oscillation Clock (f_{RL})	TM50 Output	External Clock from Peripheral Hardware Pins
16-bit timer/event counter	00	Y	N	N	N	Y (TI000 pin)[Note 1]
	01	Y	N	N	N	Y (TI001 pin)[Notes 1, 2]
	02	Y	N	N	N	Y (TI002 pin)[Notes 1, 2]
8-bit timer/event counter	50	Y	N	N	N	Y (TI50 pin)[Note 1]
	51	Y	N	N	N	Y (TI51 pin)[Note 1]
8-bit timer	H0	Y	N	N	Y	N
	H1	Y	N	Y	N	N
Real-time counter		Y	Y	N	N	N
Watchdog timer		N	N	Y	N	N
Buzzer output[Note 3]		Y	N	N	N	N
Clock output		Y	Y	N	N	N
A/D converter		Y	N	N	N	N
Serial interface	UART0	Y	N	N	Y	N
	UART60	Y	N	N	Y	N
	UART61[Note 3]	Y	N	N	N	N
	CSI10	Y	N	N	N	Y ($\overline{SCK10}$ pin)[Note 1]
	CSI11	Y	N	N	N	Y ($\overline{SCK11}$ pin)[Note 1]
	IICA00	Y	N	N	N	Y (SCLA0 pin)[Note 1]
	IICA01	Y	N	N	N	Y (SCLA1 pin)[Note 1]
	IICA02	Y	N	N	N	Y (SCLA2 pin)[Note 1]
CEC transmission/reception circuit		Y	Y	N	N	N
Remote controller receiver		Y	Y	N	N	N

Notes 1. When the CPU is operating on the subsystem clock and the internal high-speed oscillation clock has been stopped, do not start operation of these functions on the external clock input from peripheral hardware pins.

2. The TM01 and TM02 of 78K0/KC2-C are not provided with the timer I/O pins.

3. 78K0/KE2-C only

Remark Y: Can be selected, N: Cannot be selected

CHAPTER 6 16-BIT TIMER/EVENT COUNTERS 00, 01, AND 02

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
16-bit timer/event counters 00	√	
16-bit timer/event counters 01	√ Note	
16-bit timer/event counters 02	√ Note	

Note The 78K0/KC2-C is not provided with the KR4 and KR5 pins.

Caution If INTTM001 occurs for TM01, data cannot be received using UART61 (only 64-pin products).
If INTTM011 occurs for TM01, data cannot be transmitted using UART61 (only 64-pin products).
If INTTM012 occurs for TM02, multiplier/divider cannot be used.

Remark √: Mounted

6.1 Functions of 16-Bit Timer/Event Counters 00, 01, and 02

16-bit timer/event counters 00, 01, and 02 have the following functions.

(1) Interval timer

16-bit timer/event counters 00 and 01 generate an interrupt request at the preset time interval.

(2) Square-wave output

16-bit timer/event counters 00 and 01 can output a square wave with any selected frequency.

(3) External event counter

16-bit timer/event counters 00 and 01 can measure the number of pulses of an externally input signal.

(4) One-shot pulse output

16-bit timer event counters 00 and 01 can output a one-shot pulse whose output pulse width can be set freely.

(5) PPG output

16-bit timer/event counters 00 and 01 can output a rectangular wave whose frequency and output pulse width can be set freely.

(6) Pulse width measurement

16-bit timer/event counters 00 and 01 can measure the pulse width of an externally input signal.

6.2 Configuration of 16-Bit Timer/Event Counters 00, 01, and 02

16-bit timer/event counters 00 and 01 include the following hardware.

Table 6-1. Configuration of 16-Bit Timer/Event Counters 00 and 01

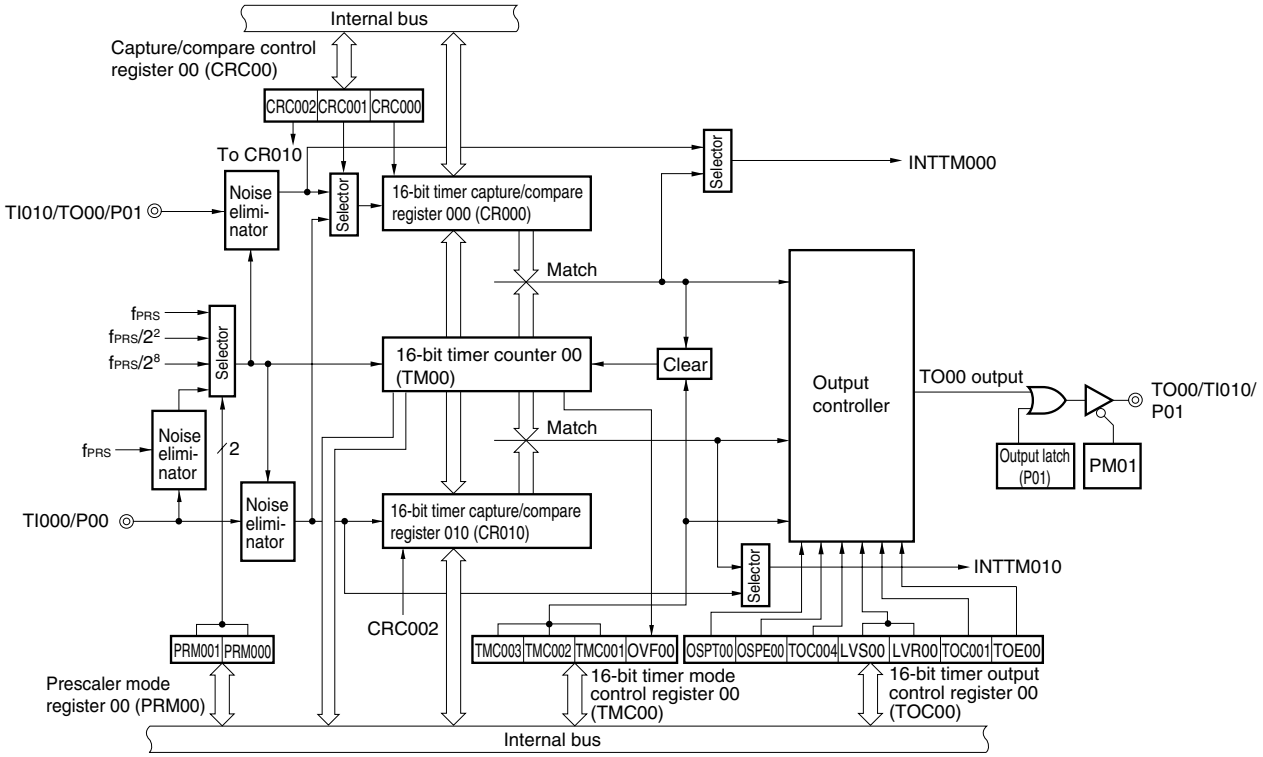
Item	Configuration
Time/counter	16-bit timer counter 0n (TM0n)
Register	16-bit timer capture/compare registers 00n, 01n (CR00n, CR01n)
Timer input	TI00n, TI01n pins
Timer output	TO0n pin, output controller
Control registers	16-bit timer mode control register 0n (TMC0n) 16-bit timer capture/compare control register 0n (CRC0n) 16-bit timer output control register 0n (TOC0n) Prescaler mode register 0n (PRM0n) Port mode registers 0, 5Note (PM0, PM5Note) Port registers 0, 5Note (P0, P5Note)

Note 78K0/KE2-C only

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figures 6-1 to 6-3 show the block diagrams.

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 00



(Cautions 1 to 3 are listed on the next page after next.)

Figure 6-2. Block Diagram of 16-Bit Timer/Event Counters 01 and 02 (78K0/KC2-C)

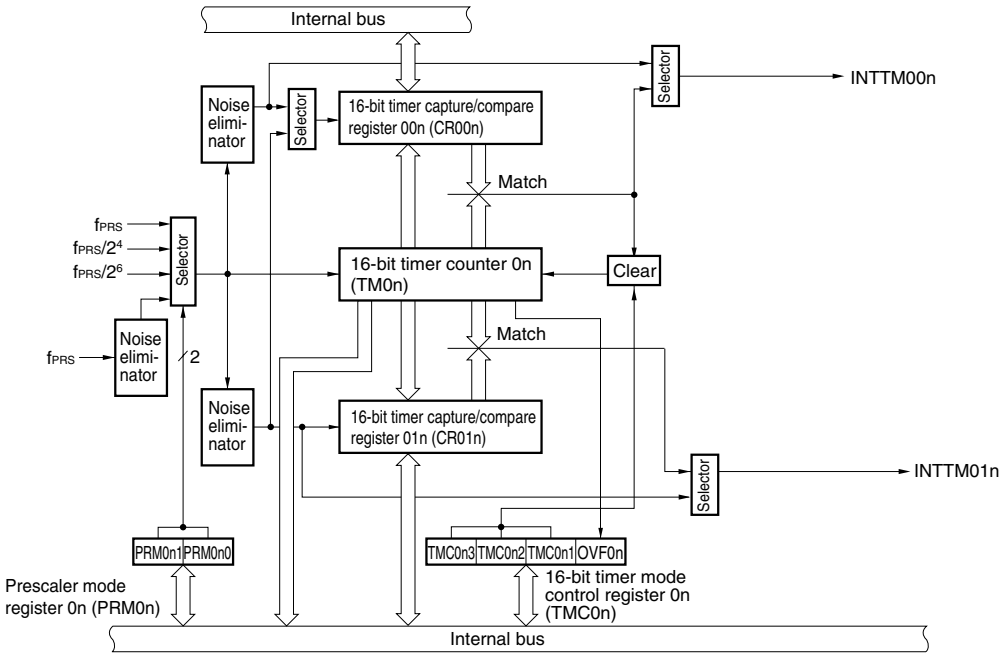
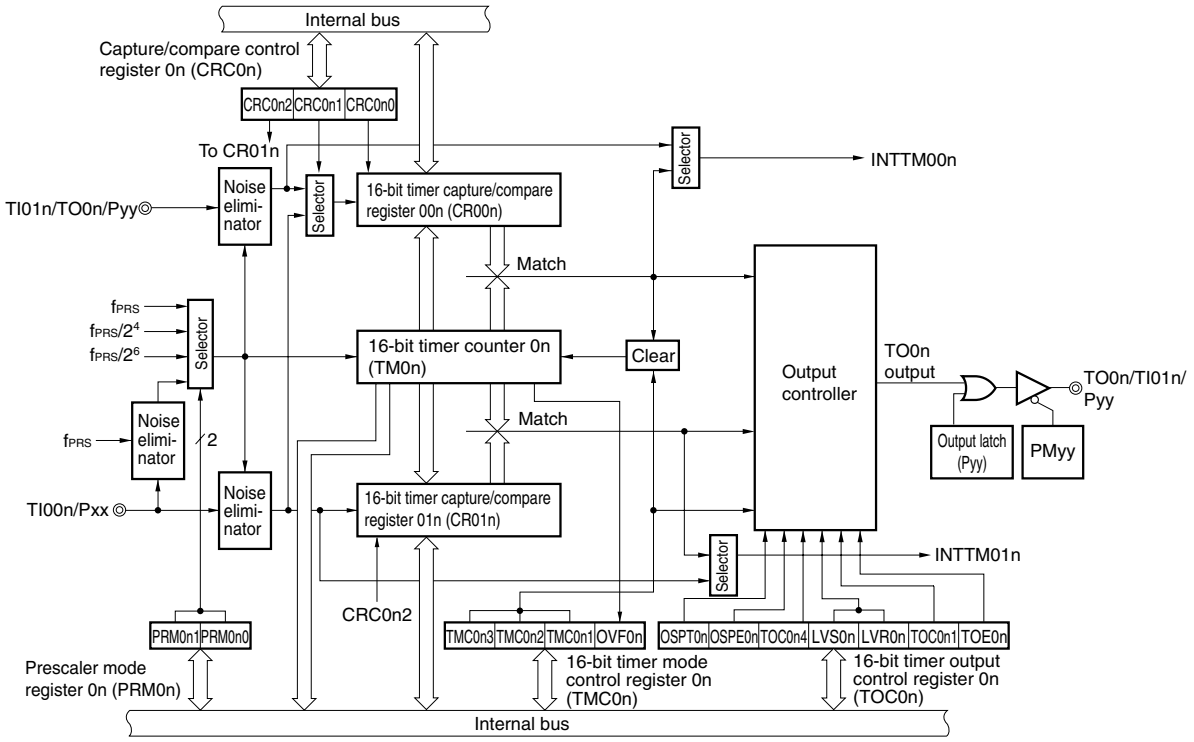


Figure 6-3. Block Diagram of 16-Bit Timer/Event Counters 01 and 02 (78K0/KE2-C)



Remarks 1. n = 1, 2

2. In the case of n = 1: xx =05, yy = 06
In the case of n = 2: xx =50, yy = 51

(Cautions 1 to 3 are listed on the next page.)

Cautions 1. The valid edge of TI010 and timer output (TO00) cannot be used for the P01 pin at the same time, the valid edge of TI011 and timer output (TO01) cannot be used for the P06 pin at the same time, and the valid edge of TI012 and timer output (TO02) cannot be used for the P51 pin at the same time. Select either of the functions.

2. If clearing of bits 3 and 2 (TMC0n3 and TMC0n2) of 16-bit timer mode control register 0n (TMC0n) to 00 and input of the capture trigger conflict, then the captured data is undefined.

3. To change the mode from the capture mode to the comparison mode, first clear the TMC0n3 and TMC0n2 bits to 00, and then change the setting.
A value that has been once captured remains stored in CR00n unless the device is reset. If the mode has been changed to the comparison mode, be sure to set a comparison value.

(1) 16-bit timer counter 0n (TM0n)

TM0n is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 6-4. Format of 16-Bit Timer Counter 0n (TM0n)

Address: FF10H, FF11H (TM00), FF44H, FF45H (TM01), FF82H, FF83H (TM02) After reset: 0000H R

The count value of TM0n can be read by reading TM0n when the value of bits 3 and 2 (TMC0n3 and TMC0n2) of 16-bit timer mode control register 0n (TMC0n) is other than 00. The value of TM0n is 0000H if it is read when TMC0n3 and TMC0n2 = 00.

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

- At reset signal generation
- If TMC0n3 and TMC0n2 are cleared to 00
- If the valid edge of the TI00n pin is input in the mode in which the clear & start occurs when inputting the valid edge to the TI00n pin
- If TM0n and CR00n match in the mode in which the clear & start occurs when TM0n and CR00n match
- OSPT0n is set to 1 in one-shot pulse output mode or the valid edge is input to the TI00n pin

Caution Even if TM0n is read, the value is not captured by CR01n.

Remark n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n) in the 78K0/KC2-C: n = 0).

(2) 16-bit timer capture/compare register 00n (CR00n), 16-bit timer capture/compare register 01n (CR01n)

CR00n and CR01n are 16-bit registers that are used with a capture function or comparison function selected by using CRC0n.

Change the value of CR00n while the timer is stopped (TMC0n3 and TMC0n2 = 00).

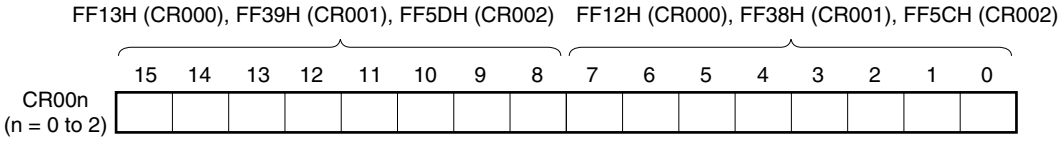
The value of CR01n can be changed during operation if the value has been set in a specific way. For details, see **6.5.1 Rewriting CR01n during TM0n operation**.

These registers can be read or written in 16-bit units.

Reset signal generation clears these registers to 0000H.

Figure 6-5. Format of 16-Bit Timer Capture/Compare Register 00n (CR00n)

Address: FF12H, FF13H (CR000), FF38H, FF39H (CR001), FF5CH, FF5DH (CR002) After reset: 0000H R/W



(i) When CR00n is used as a compare register

The value set in CR00n is constantly compared with the TM0n count value, and an interrupt request signal (INTTM00n) is generated if they match. The value is held until CR00n is rewritten.

Caution CR00n does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

(ii) When CR00n is used as a capture register

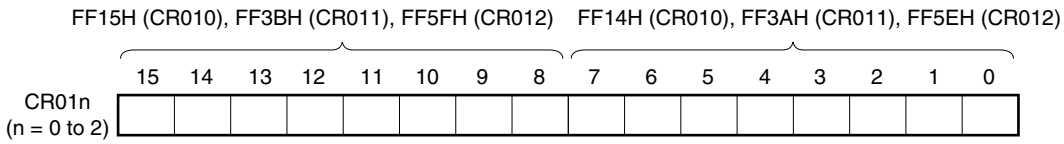
The count value of TM0n is captured to CR00n when a capture trigger is input.

As the capture trigger, an edge of a phase reverse to that of the TI00n pin or the valid edge of the TI01n pin can be selected by using CRC0n or PRM0n.

Remark n = 0 to 2 (This is, however, in the case of the timer Input pins (TI00n, TI01n), and CRC0n register in the 78K0/KC2-C: n = 0).

Figure 6-6. Format of 16-Bit Timer Capture/Compare Register 01n (CR01n)

Address: FF14H, FF15H (CR010), FF3AH, FF3BH (CR011), FF5EH, FF5FH (CR012) After reset: 0000H R/W



(i) When CR01n is used as a compare register

The value set in CR01n is constantly compared with the TM0n count value, and an interrupt request signal (INTTM01n) is generated if they match.

Caution CR01n does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

(ii) When CR01n is used as a capture register

The count value of TM0n is captured to CR01n when a capture trigger is input.

It is possible to select the valid edge of the TI00n pin as the capture trigger. The TI00n pin valid edge is set by PRM0n.

Remark n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n) in the 78K0/KC2-C: n = 0).

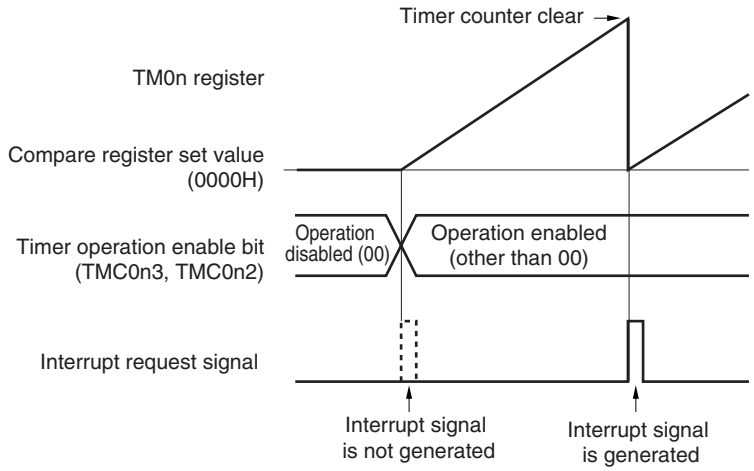
(iii) Setting range when CR00n or CR01n is used as a compare register

When CR00n or CR01n is used as a compare register, set it as shown below.

Operation	CR00n Register Setting Range	CR01n Register Setting Range
Operation as interval timer	0000H < N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH Normally, this setting is not used. Mask the match interrupt signal (INTTM01n).
Operation as square-wave output		
Operation as external event counter		
Operation in the clear & start mode entered by TI00n pin valid edge input	0000H[Note] ≤ N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH
Operation as free-running timer		
Operation as PPG output	M < N ≤ FFFFH	0000H[Note] ≤ M < N
Operation as one-shot pulse output	0000H[Note] ≤ N ≤ FFFFH (N ≠ M)	0000H[Note] ≤ M ≤ FFFFH (M ≠ N)

Note When 0000H is set, a match interrupt immediately after the timer operation does not occur and timer output is not changed, and the first match timing is as follows. A match interrupt occurs at the timing when the timer counter (TM0n register) is changed from 0000H to 0001H.

- When the timer counter is cleared due to overflow
- When the timer counter is cleared due to TI00n pin valid edge (when clear & start mode is entered by TI00n pin valid edge input)
- When the timer counter is cleared due to compare match (when clear & start mode is entered by match between TM0n and CR00n (CR00n = other than 0000H, CR01n = 0000H))

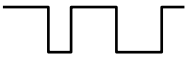
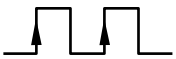
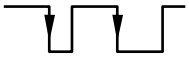
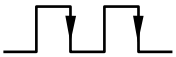
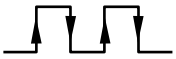
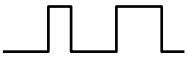
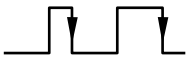
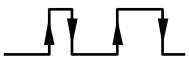


Remarks 1. N: CR00n register set value, M: CR01n register set value

2. For details of TMC0n3 and TMC0n2, see **6.3 (1) 16-bit timer mode control register 0n (TMC0n)**.

3. n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n) in the 78K0/KC2-C: n = 0).

Table 6-2. Capture Operation of CR00n and CR01n

External Input Signal / Capture Operation	TI00n Pin Input		TI01n Pin Input	
Capture operation of CR00n	CRC0n1 = 1 TI00n pin input (reverse phase) 	Set values of ES0n1 and ES0n0 Position of edge to be captured	CRC0n1 bit = 0 TI01n pin input 	Set values of ES1n1 and ES1n0 Position of edge to be captured
		01: Rising 		01: Rising 
		00: Falling 		00: Falling 
		11: Both edges (cannot be captured)		11: Both edges 
	Interrupt signal	INTTM00n signal is not generated even if value is captured.	Interrupt signal	INTTM00n signal is generated each time value is captured.
Capture operation of CR01n	TI00n pin input[Note] 	Set values of ES0n1 and ES0n0 Position of edge to be captured		
		01: Rising 		
		00: Falling 		
		11: Both edges 		
	Interrupt signal	INTTM01n signal is generated each time value is captured.		

Note The capture operation of CR01n is not affected by the setting of the CRC0n1 bit.

Caution To capture the count value of the TM0n register to the CR00n register by using the phase reverse to that input to the TI00n pin, the interrupt request signal (INTTM00n) is not generated after the value has been captured. If the valid edge is detected on the TI01n pin during this operation, the capture operation is not performed but the INTTM00n signal is generated as an external interrupt signal. To not use the external interrupt, mask the INTTM00n signal.

Remarks 1. CRC0n1: See **6.3 (2) Capture/compare control register 0n (CRC0n)**.
ES1n1, ES1n0, ES0n1, ES0n0: See **6.3 (4) Prescaler mode register 0n (PRM0n)**.

2. n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n) in the 78K0/KC2-C: n = 0).

6.3 Registers Controlling 16-Bit Timer/Event Counters 00, 01, and 02

Registers used to control 16-bit timer/event counters 00 and 01 are shown below.

- 16-bit timer mode control register 0n (TMC0n)
- Capture/compare control register 0n (CRC0n)
- 16-bit timer output control register 0n (TOC0n)
- Prescaler mode register 0n (PRM0n)
- Port mode registers 0, 5[Note] (PM0, PM5[Note])
- Port registers 0, 5[Note] (P0, P5[Note])

Note 78K0/KE2-C only

(1) 16-bit timer mode control register 0n (TMC0n)

TMC0n is an 8-bit register that sets the 16-bit timer/event counter 0n operation mode, TM0n clear mode, and output timing, and detects an overflow.

Rewriting TMC0n is prohibited during operation (when TMC0n3 and TMC0n2 = other than 00). However, it can be changed when TMC0n3 and TMC0n2 are cleared to 00 (stopping operation) and when OVF0n is cleared to 0.

TMC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TMC0n to 00H.

Caution 16-bit timer/event counter 0n starts operation at the moment TMC0n2 and TMC0n3 are set to values other than 00 (operation stop mode), respectively. Set TMC0n2 and TMC0n3 to 00 to stop the operation.

Remark n = 0 to 2 (This is, however, in the case of the CRC0n and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-7. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FFBAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	Operation enable of 16-bit timer/event counter 00
0	0	Disables 16-bit timer/event counter 00 operation. Stops supplying operating clock. Clears 16-bit timer counter 00 (TM00).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI000 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM00 and CR000

TMC001	Condition to reverse timer output (TO00)
0	• Match between TM00 and CR000 or match between TM00 and CR010
1	• Match between TM00 and CR000 or match between TM00 and CR010 • Trigger input of TI000 pin valid edge

OVF00	TM00 overflow flag
Clear (0)	Clears OVF00 to 0 or TMC003 and TMC002 = 00
Set (1)	Overflow occurs.
OVF00 is set to 1 when the value of TM00 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI000 pin valid edge input, and clear & start mode entered upon a match between TM00 and CR000). It can also be set to 1 by writing 1 to OVF00.	

Note The TI000 pin valid edge is set by bits 5 and 4 (ES001, ES000) of prescaler mode register 00 (PRM00).

Figure 6-8. Format of 16-Bit Timer Mode Control Register 01 (TMC01)

Address: FF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC01	0	0	0	0	TMC013	TMC012	TMC011	OVF01

TMC013	TMC012	Operation enable of 16-bit timer/event counter 01
0	0	Disables 16-bit timer/event counter 01 operation. Stops supplying operating clock. Clears 16-bit timer counter 01 (TM01).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI001 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM01 and CR001

TMC011	Condition to reverse timer output (TO01)
0	• Match between TM01 and CR001 or match between TM01 and CR011
1	• Match between TM01 and CR001 or match between TM01 and CR011 • Trigger input of TI001 pin valid edge

OVF01	TM01 overflow flag
Clear (0)	Clears OVF01 to 0 or TMC013 and TMC012 = 00
Set (1)	Overflow occurs.
OVF01 is set to 1 when the value of TM01 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI001 pin valid edge input, and clear & start mode entered upon a match between TM01 and CR001). It can also be set to 1 by writing 1 to OVF01.	

Note The TI001 pin valid edge is set by bits 5 and 4 (ES011, ES010) of prescaler mode register 01 (PRM01).

Figure 6-9. Format of 16-Bit Timer Mode Control Register 02 (TMC02)

Address: FFA8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC02	0	0	0	0	TMC023	TMC022	TMC021	OVF02

TMC023	TMC022	Operation enable of 16-bit timer/event counter 02
0	0	Disables 16-bit timer/event counter 02 operation. Stops supplying operating clock. Clears 16-bit timer counter 02 (TM02).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI002 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM02 and CR002

TMC021	Condition to reverse timer output (TO02)
0	• Match between TM02 and CR002 or match between TM02 and CR012
1	• Match between TM02 and CR002 or match between TM02 and CR012 • Trigger input of TI002 pin valid edge

OVF02	TM02 overflow flag
Clear (0)	Clears OVF02 to 0 or TMC023 and TMC022 = 00
Set (1)	Overflow occurs.
OVF02 is set to 1 when the value of TM02 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI002 pin valid edge input, and clear & start mode entered upon a match between TM02 and CR002). It can also be set to 1 by writing 1 to OVF02.	

Note The TI002 pin valid edge is set by bits 5 and 4 (ES021, ES020) of prescaler mode register 02 (PRM02).

(2) Capture/compare control register 0n (CRC0n)

CRC0n is the register that controls the operation of CR00n and CR01n.

Changing the value of CRC0n is prohibited during operation (when TMC0n3 and TMC0n2 = other than 00).

CRC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CRC0n to 00H.

Caution The CRC01 and CRC02 registers are provided only for the 78K0/KE2-C. For the 78K0/KC2-C, the CR001, CR011, CR002, and CR012 registers can be used only as compare registers.

Remark n = 0 to 2 (This is, however, in the case of 78K0/KC2-C: n = 0).

Figure 6-10. Format of Capture/Compare Control Register 00 (CRC00)

Address: FFBCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operates as compare register
1	Operates as capture register

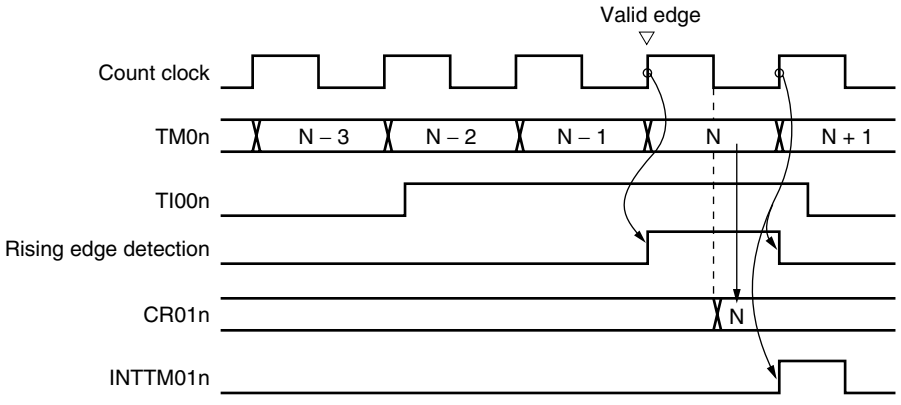
CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010 pin
1	Captures on valid edge of TI000 pin by reverse phase[Note]
The valid edge of the TI010 and TI000 pin is set by PRM00. If ES001 and ES000 are set to 11 (both edges) when CRC001 is 1, the valid edge of the TI000 pin cannot be detected.	

CRC000	CR000 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC003 and TMC002 are set to 11 (clear & start mode entered upon a match between TM00 and CR000), be sure to set CRC000 to 0.	

Note When the valid edge is detected from the TI010 pin, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

Figure 6-11. Example of CR01n Capture Operation (When Rising Edge Is Specified)



Remark n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n) in the 78K0/KC2-C: n = 0).

Figure 6-12. Format of Capture/Compare Control Register 01 (CRC01) (78K0/KE2-C only)

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC01	0	0	0	0	0	CRC012	CRC011	CRC010

CRC012	CR011 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC011	CR001 capture trigger selection
0	Captures on valid edge of TI011 pin
1	Captures on valid edge of TI001 pin by reverse phase[Note]
The valid edge of the TI011 and TI001 pin is set by PRM01. If ES011 and ES010 are set to 11 (both edges) when CRC011 is 1, the valid edge of the TI001 pin cannot be detected.	

CRC010	CR001 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC013 and TMC012 are set to 11 (clear & start mode entered upon a match between TM01 and CR001), be sure to set CRC010 to 0.	

Note When the valid edge is detected from the TI011 pin, the capture operation is not performed but the INTTM001 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 01 (PRM01) (see Figure 6-11 Example of CR01n Capture Operation (When Rising Edge Is Specified)).

Figure 6-13. Format of Capture/Compare Control Register 02 (CRC02) (78K0/KE2-C only)

Address: FFAAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC02	0	0	0	0	0	CRC022	CRC021	CRC020

CRC022	CR012 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC012	CR002 capture trigger selection
0	Captures on valid edge of TI012 pin
1	Captures on valid edge of TI002 pin by reverse phase[Note]
The valid edge of the TI012 and TI002 pin is set by PRM02. If ES021 and ES020 are set to 11 (both edges) when CRC021 is 1, the valid edge of the TI002 pin cannot be detected.	

CRC020	CR002 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC023 and TMC022 are set to 11 (clear & start mode entered upon a match between TM02 and CR002), be sure to set CRC020 to 0.	

Note When the valid edge is detected from the TI012 pin, the capture operation is not performed but the INTTM002 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 02 (PRM02) (see Figure 6-11 Example of CR01n Capture Operation (When Rising Edge Is Specified)).

(3) 16-bit timer output control register 0n (TOC0n)

TOC0n is an 8-bit register that controls the TO0n output.

TOC0n can be rewritten while only OSPT0n is operating (when TMC0n3 and TMC0n2 = other than 00). Rewriting the other bits is prohibited during operation.

However, TOC0n4 can be rewritten during timer operation as a means to rewrite CR01n (see **6.5.1 Rewriting CR01n during TM0n operation**).

TOC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TOC0n to 00H.

Cautions 1. The TOC01 and TOC02 registers are provided only for the 78K0/KE2-C. The 78K0/KC2-C is not provided with the KR4 and KR5 pins.

2. Be sure to set TOC0n using the following procedure.

<1> Set TOC0n4 and TOC0n1 to 1.

<2> Set only TOE0n to 1.

<3> Set either of LVS0n or LVR0n to 1.

Remark n = 0 to 2 (This is, however, in the case of the 78K0/KC2-C: n = 0).

Figure 6-14. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FFBDH After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always “0” when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM00 is cleared and started.	

OSPE00	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI000 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.	

TOC004	TO00 output control on match between CR010 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM010) is generated even when TOC004 = 0.	

LVS00	LVR00	Setting of TO00 output status
0	0	No change
0	1	Initial value of TO00 output is low level (TO00 output is cleared to 0).
1	0	Initial value of TO00 output is high level (TO00 output is set to 1).
1	1	Setting prohibited

- LVS00 and LVR00 can be used to set the initial value of the TO00 output level. If the initial value does not have to be set, leave LVS00 and LVR00 as 00.
- Be sure to set LVS00 and LVR00 when TOE00 = 1.
  LVS00, LVR00, and TOE00 being simultaneously set to 1 is prohibited.
- LVS00 and LVR00 are trigger bits. By setting these bits to 1, the initial value of the TO00 output level can be set. Even if these bits are cleared to 0, TO00 output is not affected.
- The values of LVS00 and LVR00 are always 0 when they are read.
- For how to set LVS00 and LVR00, see **6.5.2 Setting LVS0n and LVR0n**.
- The actual TO00/TI010/P01 pin output is determined depending on PM01 and P01, besides TO00 output.

TOC001	TO00 output control on match between CR000 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM000) is generated even when TOC001 = 0.	

TOE00	TO00 output control
0	Disables output (TO00 output fixed to low level)
1	Enables output

Figure 6-15. Format of 16-Bit Timer Output Control Register 01 (TOC01) (78K0/KE2-C only)

Address: FF9DH After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC01	0	OSPT01	OSPE01	TOC014	LVS01	LVR01	TOC011	TOE01

OSPT01	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always 0 when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM01 is cleared and started.	

OSPE01	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI001 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM01 and CR001.	

TOC014	TO01 output control on match between CR011 and TM01
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM011) is generated even when TOC014 = 0.	

LVS01	LVR01	Setting of TO01 output status
0	0	No change
0	1	Initial value of TO01 output is low level (TO01 output is cleared to 0).
1	0	Initial value of TO01 output is high level (TO01 output is set to 1).
1	1	Setting prohibited

- LVS01 and LVR01 can be used to set the initial value of the TO01 output level. If the initial value does not have to be set, leave LVS01 and LVR01 as 00.
- Be sure to set LVS01 and LVR01 when TOE01 = 1.
  LVS01, LVR01, and TOE01 being simultaneously set to 1 is prohibited.
- LVS01 and LVR01 are trigger bits. By setting these bits to 1, the initial value of the TO01 output level can be set. Even if these bits are cleared to 0, TO01 output is not affected.
- The values of LVS01 and LVR01 are always 0 when they are read.
- For how to set LVS01 and LVR01, see **6.5.2 Setting LVS0n and LVR0n**.
- The actual TO01/TI011/P06 pin output is determined depending on PM06 and P06, besides TO01 output.

TOC011	TO01 output control on match between CR001 and TM01
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM001) is generated even when TOC011 = 0.	

TOE01	TO01 output control
0	Disables output (TO01 output is fixed to low level)
1	Enables output

Figure 6-16. Format of 16-Bit Timer Output Control Register 02 (TOC02) (78K0/KE2-C only)

Address: FFABH After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC02	0	OSPT02	OSPE02	TOC024	LVS02	LVR02	TOC021	TOE02

OSPT02	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always 0 when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM02 is cleared and started.	

OSPE02	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI002 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM02 and CR002.	

TOC024	TO02 output control on match between CR012 and TM02
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM012) is generated even when TOC024 = 0.	

LVS02	LVR02	Setting of TO02 output status
0	0	No change
0	1	Initial value of TO02 output is low level (TO02 output is cleared to 0).
1	0	Initial value of TO02 output is high level (TO02 output is set to 1).
1	1	Setting prohibited

- LVS02 and LVR02 can be used to set the initial value of the TO02 output level. If the initial value does not have to be set, leave LVS02 and LVR02 as 00.
- Be sure to set LVS02 and LVR02 when TOE02 = 1.
  LVS02, LVR02, and TOE02 being simultaneously set to 1 is prohibited.
- LVS02 and LVR02 are trigger bits. By setting these bits to 1, the initial value of the TO02 output level can be set. Even if these bits are cleared to 0, TO02 output is not affected.
- The values of LVS02 and LVR02 are always 0 when they are read.
- For how to set LVS02 and LVR02, see **6.5.2 Setting LVS0n and LVR0n**.
- The actual TO02/TI012/P51 pin output is determined depending on PM51 and P51, besides TO02 output.

TOC021	TO02 output control on match between CR002 and TM02
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM002) is generated even when TOC021 = 0.	

TOE02	TO02 output control
0	Disables output (TO02 output is fixed to low level)
1	Enables output

(4) Prescaler mode register 0n (PRM0n)

PRM0n is the register that sets the TM0n count clock and TI00n and TI01n pin input valid edges.

Rewriting PRM0n is prohibited during operation (when TMC0n3 and TMC0n2 = other than 00).

PRM0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears PRM0n to 00H.

Cautions 1. Do not apply the following setting when setting the PRM0n1 and PRM0n0 bits to 11 (to specify the valid edge of the TI00n pin as a count clock).

- **Clear & start mode entered by the TI00n pin valid edge**
- **Setting the TI00n pin as a capture trigger**

2. If the operation of the 16-bit timer/event counter 0n is enabled when the TI00n or TI01n pin is at high level and when the valid edge of the TI00n or TI01n pin is specified to be the rising edge or both edges, the high level of the TI00n or TI01n pin is detected as a rising edge. Note this when the TI00n or TI01n pin is pulled up. However, the rising edge is not detected when the timer operation has been once stopped and then is enabled again.

3. The valid edge of TI010 and timer output (TO00) cannot be used for the P01 pin at the same time, the valid edge of TI011 and timer output (TO01) cannot be used for the P06 pin at the same time, and the valid edge of TI012 and timer output (TO02) cannot be used for the P51 pin at the same time. Select either of the functions.

Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n,TI01n) in the 78K0/KC2-C: n = 0).

Figure 6-17. Format of Prescaler Mode Register 00 (PRM00)

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES001	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM001	PRM000	Count clock selection				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.12 kHz
1	1	TI000 valid edge (Notes 1, 2)				

Notes 1. The external clock from the TI000 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

2. Do not start timer operation with the external clock from the TI000 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 6-18. Format of Prescaler Mode Register 01 (PRM01)

Address: FF9BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM01	ES111	ES110	ES011	ES010	0	0	PRM011	PRM010

ES111	ES110	TI011 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES011	ES010	TI001 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM011	PRM010	Count clock selection				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	$f_{PRS}/2^6$	31.25 kHz	78.125 kHz	156.25 kHz	312.5 kHz
1	1	TI001 valid edge[Notes 1, 2]				

Notes 1. The external clock from the TI001 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

2. Do not start timer operation with the external clock from the TI001 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 6-19. Format of Prescaler Mode Register 02 (PRM02)

Address: FF9BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM02	ES121	ES120	ES021	ES020	0	0	PRM021	PRM020

ES121	ES120	TI012 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES021	ES020	TI002 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM021	PRM020	Count clock selection				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	$f_{PRS}/2^6$	31.25 kHz	78.125 kHz	156.25 kHz	312.5 kHz
1	1	TI002 valid edge Notes 1, 2				

Notes 1. The external clock from the TI002 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

2. Do not start timer operation with the external clock from the TI002 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Remark f_{PRS}: Peripheral hardware clock frequency

(5) Port mode registers (PM0, PM5)

This register sets ports 0 and 5 input/output in 1-bit units.

When using the P01/TO00/TI010, P06/TO01/TI011, and P51/TO02/TI012 pins for timer output, set PM01, PM06, PM51, and the output latches of P01, P06, P51 to 0.

When using the P00/TI000, P01/TO00/TI010, P05/TI001, P06/TO01/TI011, P50/TI002, and P51/TO02/TI012 pins for timer input, set PM00, PM01, PM05, PM06, PM50, and PM51 to 1. At this time, the output latches of P00, P01, P05, P06, P50, and P51 may be 0 or 1.

PM0 and PM5 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM0 to FFH.

Figure 6-20. Format of Port Mode Register (78K0/KC2-C)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	1	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 6-21. Format of Port Mode Register (78K0/KE2-C)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	1	1	PM53	PM52	PM51	PM50

PMmn	Pmn pin I/O mode selection (m = 0, 5; n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Operation of 16-Bit Timer/Event Counters 00, 01, and 02

6.4.1 Interval timer operation

If bits 3 and 2 (TMC0n3 and TMC0n2) of the 16-bit timer mode control register (TMC0n) are set to 11 (clear & start mode entered upon a match between TM0n and CR00n), the count operation is started in synchronization with the count clock.

When the value of TM0n later matches the value of CR00n, TM0n is cleared to 0000H and a match interrupt signal (INTTM00n) is generated. This INTTM00n signal enables TM0n to operate as an interval timer.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

Figure 6-22. Block Diagram of Interval Timer Operation

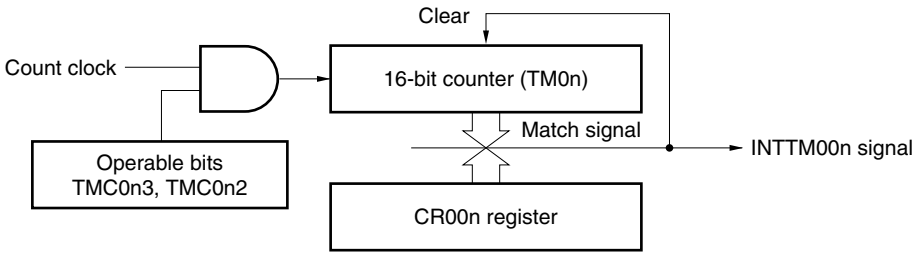
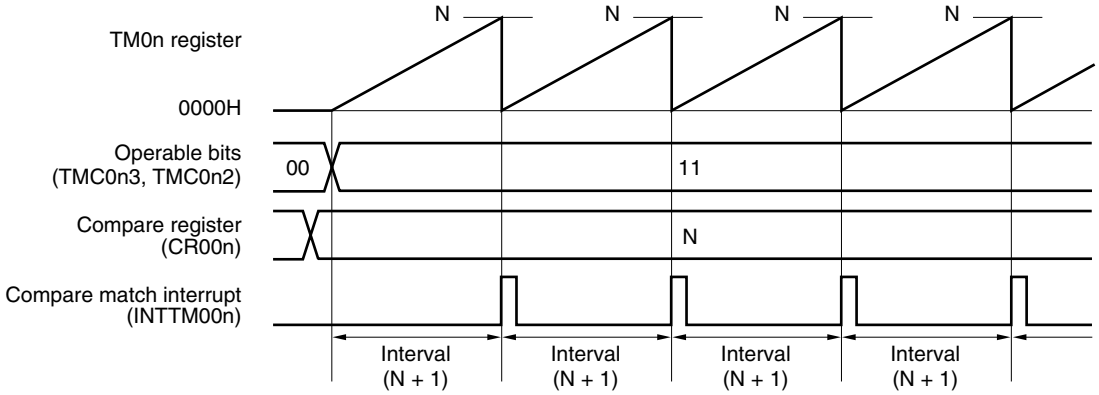


Figure 6-23. Basic Timing Example of Interval Timer Operation



Remark n = 0 to 2

Figure 6-24. Example of Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 0n (TMC0n)

				TMC0n3	TMC0n2	TMC0n1	OVF0n
0	0	0	0	1	1	0	0

Clears and starts on match between TM0n and CR00n.

(b) Capture/compare control register 0n (CRC0n)

					CRC0n2	CRC0n1	CRC0n0
0	0	0	0	0	0	0	0

CR00n used as compare register

(c) 16-bit timer output control register 0n (TOC0n)

	OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n
0	0	0	0	0	0	0	0

(d) Prescaler mode register 0n (PRM0n)

ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
0	0	0	0	0	0	0/1	0/1

Selects count clock

(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

If M is set to CR00n, the interval time is as follows.

- Interval time = (M + 1) × Count clock cycle

Setting CR00n to 0000H is prohibited.

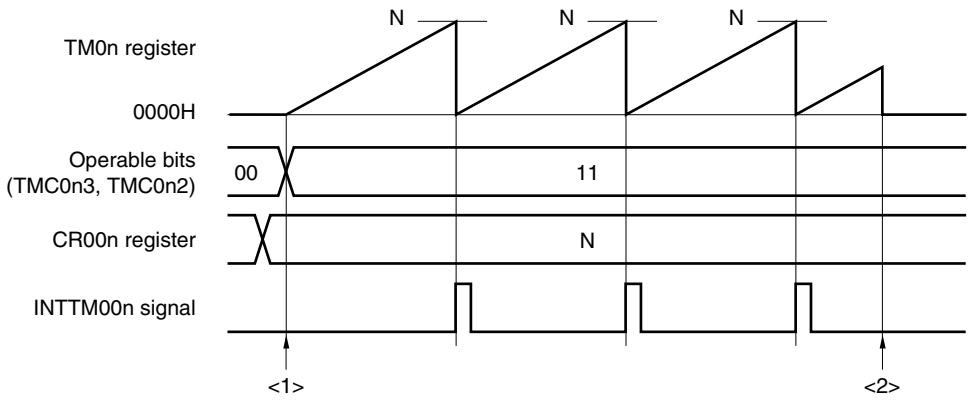
(g) 16-bit capture/compare register 01n (CR01n)

Usually, CR01n is not used for the interval timer function. However, a compare match interrupt (INTTM01n) is generated when the set value of CR01n matches the value of TM0n.

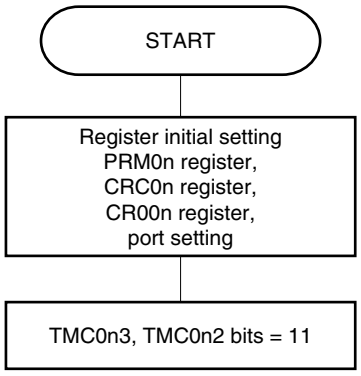
Therefore, mask the interrupt request by using the interrupt mask flag (TMMK01n).

Remark n = 0 to 2 (This is, however, in the case of the CRC0n and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-25. Example of Software Processing for Interval Timer Function



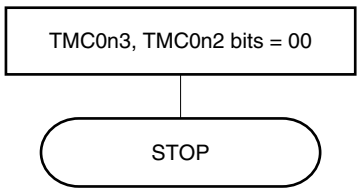
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC0n3 and TMC0n2 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC0n3 and TMC0n2 bits to 00.

Remark n = 0 to 2 (This is, however, in the case of the CRC0n register in the 78K0/KC2-C: n = 0).

6.4.2 Square-wave output operation

When 16-bit timer/event counter 0n operates as an interval timer (see **6.4.1**), a square wave can be output from the TO0n pin by setting the 16-bit timer output control register 0n (TOC0n) to 03H.

When TMC0n3 and TMC0n2 are set to 11 (count clear & start mode entered upon a match between TM0n and CR00n), the counting operation is started in synchronization with the count clock.

When the value of TM0n later matches the value of CR00n, TM0n is cleared to 0000H, an interrupt signal (INTTM00n) is generated, and TO0n output is inverted. This TO0n output that is inverted at fixed intervals enables TO0n to output a square wave.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

Figure 6-26. Block Diagram of Square-Wave Output Operation

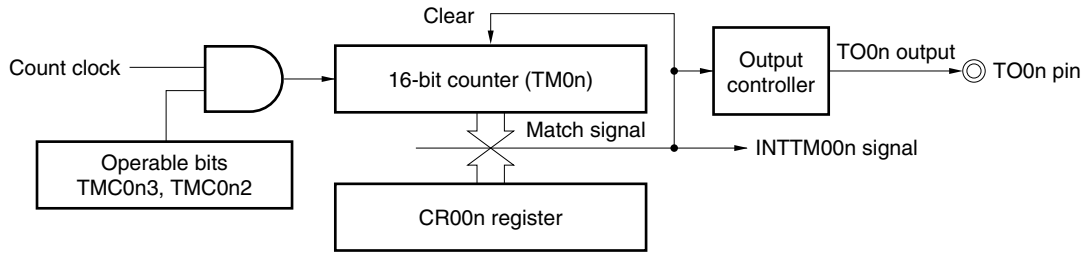
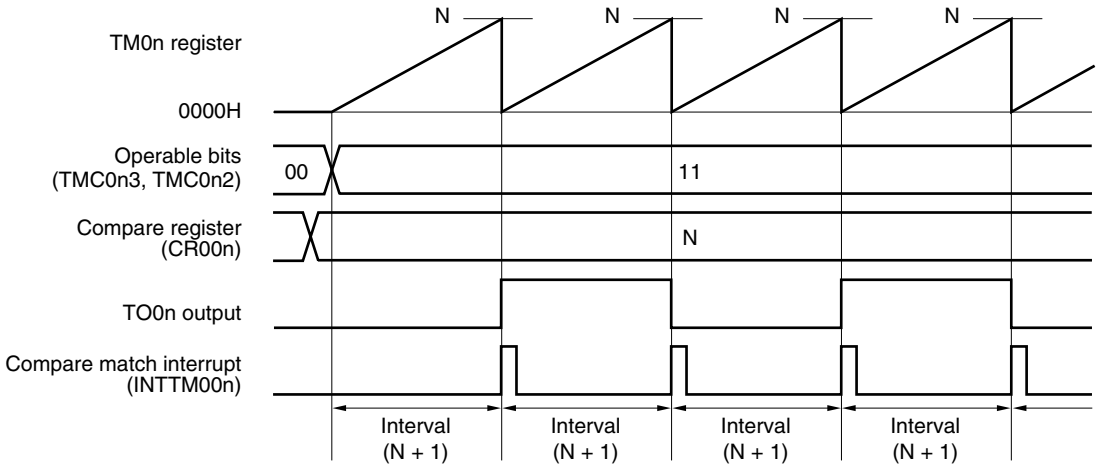


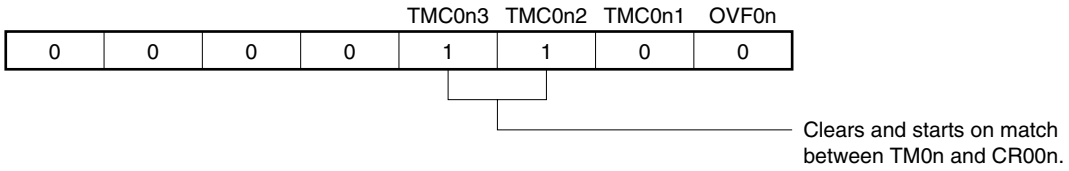
Figure 6-27. Basic Timing Example of Square-Wave Output Operation



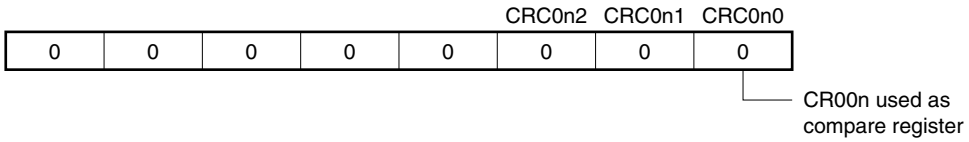
Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n) and TOC0n register in the 78K0/KC2-C: n = 0).

Figure 6-28. Example of Register Settings for Square-Wave Output Operation

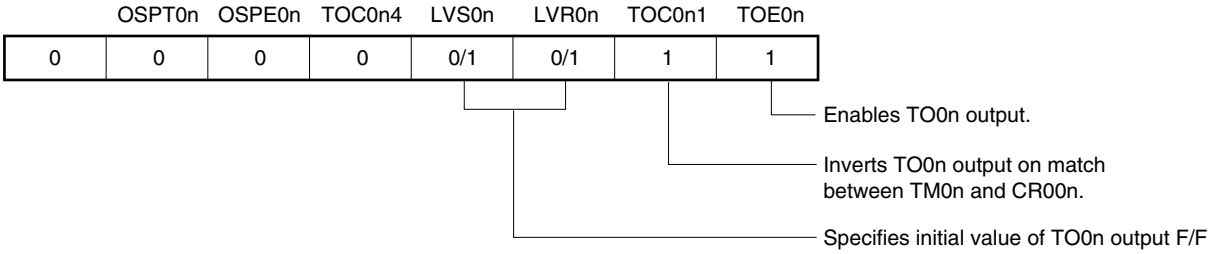
(a) 16-bit timer mode control register 0n (TMC0n)



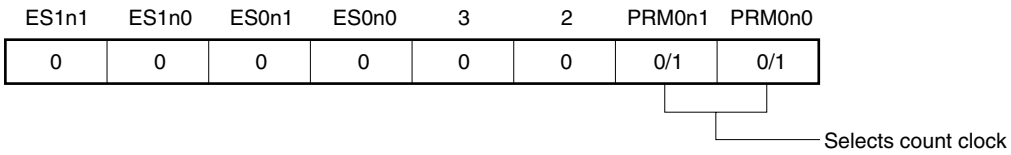
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)



(d) Prescaler mode register 0n (PRM0n)



(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

If M is set to CR00n, the interval time is as follows.

- Square wave frequency = 1 / [2 × (M + 1) × Count clock cycle]

Setting CR00n to 0000H is prohibited.

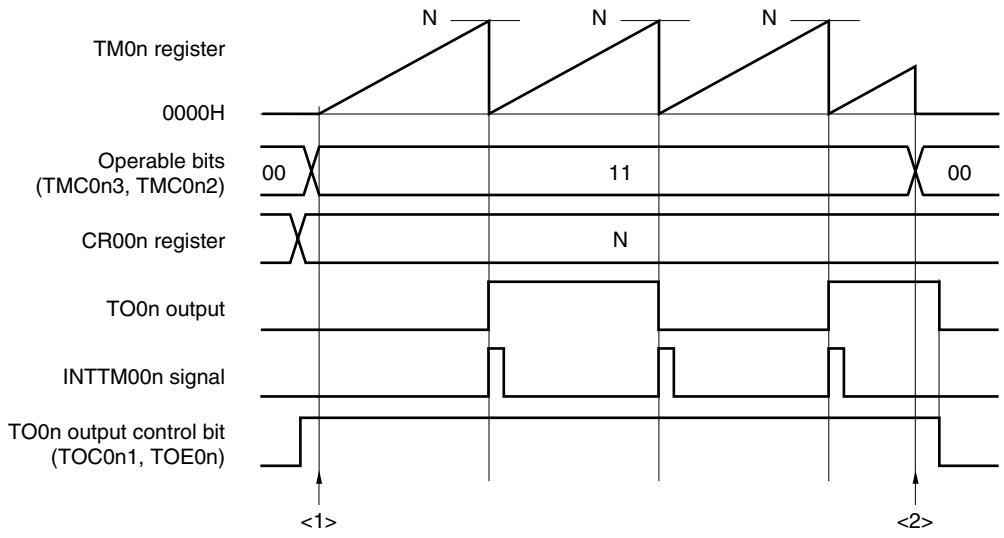
(g) 16-bit capture/compare register 01n (CR01n)

Usually, CR01n is not used for the square-wave output function. However, a compare match interrupt (INTTM01n) is generated when the set value of CR01n matches the value of TM0n.

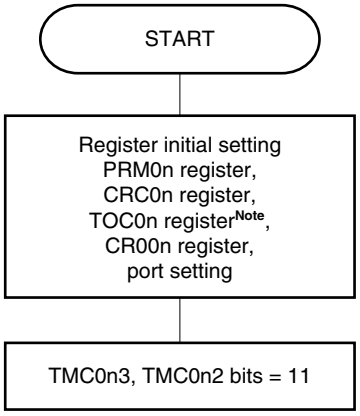
Therefore, mask the interrupt request by using the interrupt mask flag (TMMK01n).

Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-29. Example of Software Processing for Square-Wave Output Function



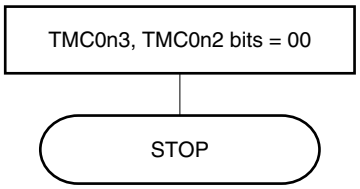
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC0n3 and TMC0n2 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC0n3 and TMC0n2 bits to 00.

Note Care must be exercised when setting TOC0n. For details, see **6.3 (3) 16-bit timer output control register 0n (TOC0n)**.

Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

6.4.3 External event counter operation

When bits 1 and 0 (PRM0n1 and PRM0n0) of the prescaler mode register 0n (PRM0n) are set to 11 (for counting up with the valid edge of the TI00n pin) and bits 3 and 2 (TMC0n3 and TMC0n2) of 16-bit timer mode control register 0n (TMC0n) are set to 11, the valid edge of an external event input is counted, and a match interrupt signal indicating matching between TM0n and CR00n (INTTM00n) is generated.

To input the external event, the TI00n pin is used. Therefore, the timer/event counter cannot be used as an external event counter in the clear & start mode entered by the TI00n pin valid edge input (when TMC0n3 and TMC0n2 = 10).

The INTTM00n signal is generated with the following timing.

- Timing of generation of INTTM00n signal (second time or later)
  = Number of times of detection of valid edge of external event × (Set value of CR00n + 1)

However, the first match interrupt immediately after the timer/event counter has started operating is generated with the following timing.

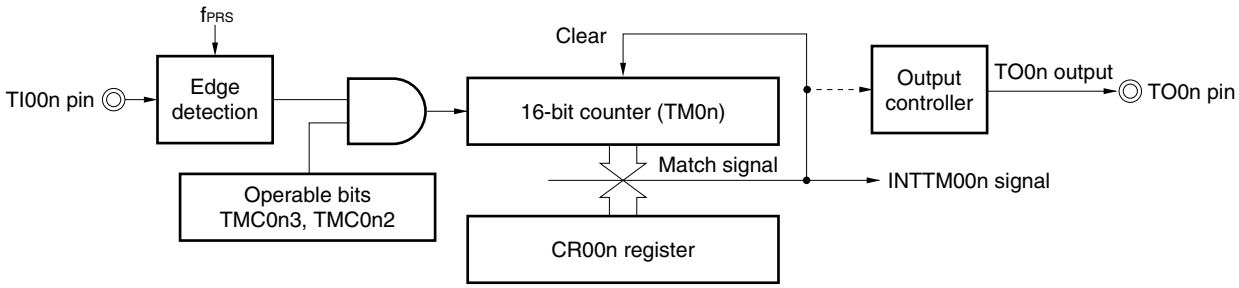
- Timing of generation of INTTM00n signal (first time only)
  = Number of times of detection of valid edge of external event input × (Set value of CR00n + 2)

To detect the valid edge, the signal input to the TI00n pin is sampled during the clock cycle of f_{PRS}. The valid edge is not detected until it is detected two times in a row. Therefore, a noise with a short pulse width can be eliminated.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

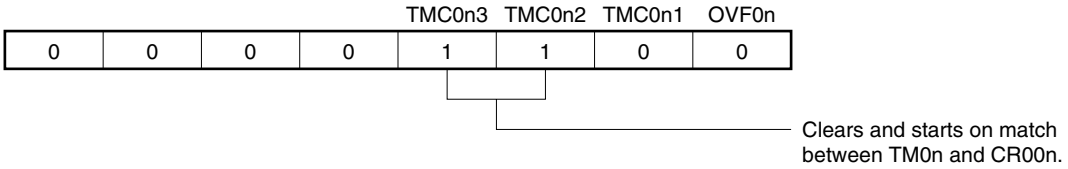
Figure 6-30. Block Diagram of External Event Counter Operation



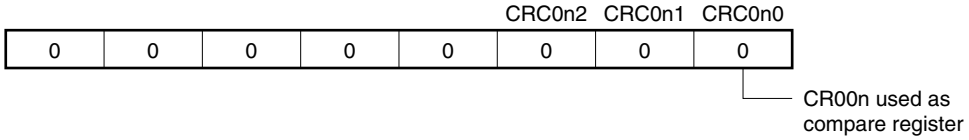
Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n) in the 78K0/KC2-C: n = 0).

Figure 6-31. Example of Register Settings in External Event Counter Mode (1/2)

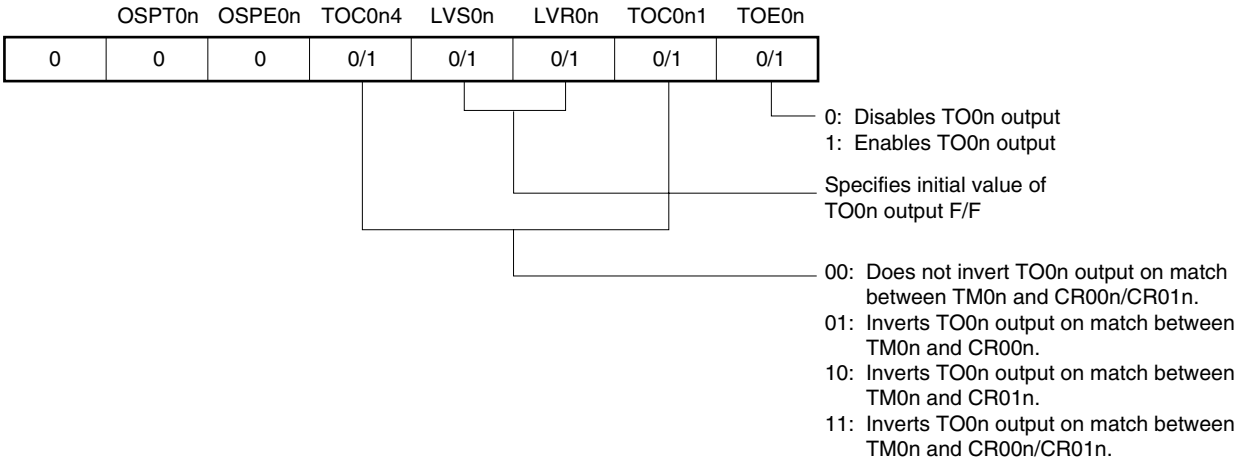
(a) 16-bit timer mode control register 0n (TMC0n)



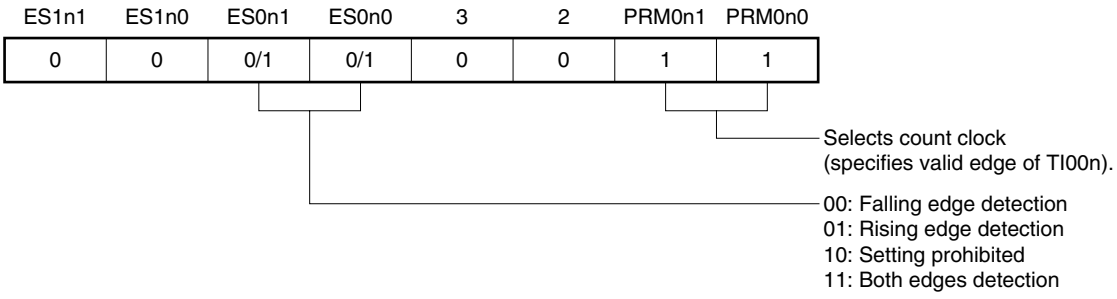
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)



(d) Prescaler mode register 0n (PRM0n)



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-31. Example of Register Settings in External Event Counter Mode (2/2)

(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

If M is set to CR00n, the interrupt signal (INTTM00n) is generated when the number of external events reaches (M + 1).

Setting CR00n to 0000H is prohibited.

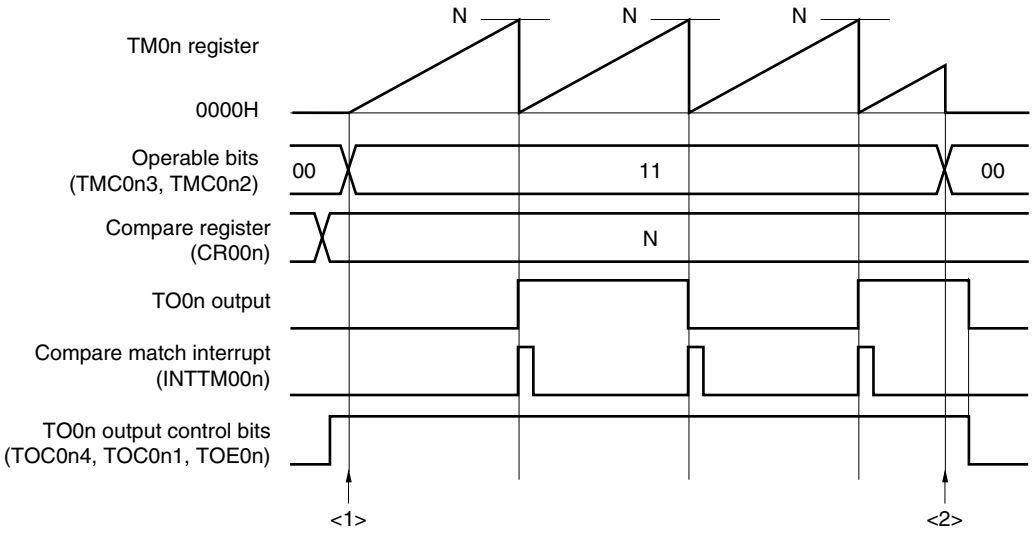
(g) 16-bit capture/compare register 01n (CR01n)

Usually, CR01n is not used in the external event counter mode. However, a compare match interrupt (INTTM01n) is generated when the set value of CR01n matches the value of TM0n.

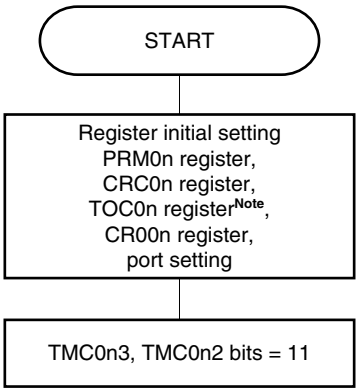
Therefore, mask the interrupt request by using the interrupt mask flag (TMMK01n).

Remark n = 0 to 2

Figure 6-32. Example of Software Processing in External Event Counter Mode



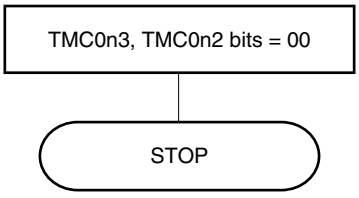
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC0n3 and TMC0n2 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC0n3 and TMC0n2 bits to 00.

Note Care must be exercised when setting TOC0n. For details, see **6.3 (3) 16-bit timer output control register 0n (TOC0n)**.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

6.4.4 Operation in clear & start mode entered by TI00n pin valid edge input

When bits 3 and 2 (TMC0n3 and TMC0n2) of 16-bit timer mode control register 0n (TMC0n) are set to 10 (clear & start mode entered by the TI00n pin valid edge input) and the count clock (set by PRM0n) is supplied to the timer/event counter, TM0n starts counting up. When the valid edge of the TI00n pin is detected during the counting operation, TM0n is cleared to 0000H and starts counting up again. If the valid edge of the TI00n pin is not detected, TM0n overflows and continues counting.

The valid edge of the TI00n pin is a cause to clear TM0n. Starting the counter is not controlled immediately after the start of the operation.

CR00n and CR01n are used as compare registers and capture registers.

(a) When CR00n and CR01n are used as compare registers

Signals INTTM00n and INTTM01n are generated when the value of TM0n matches the value of CR00n and CR01n.

(b) When CR00n and CR01n are used as capture registers

The count value of TM0n is captured to CR00n and the INTTM00n signal is generated when the valid edge is input to the TI01n pin (or when the phase reverse to that of the valid edge is input to the TI00n pin).

When the valid edge is input to the TI00n pin, the count value of TM0n is captured to CR01n and the INTTM01n signal is generated. As soon as the count value has been captured, the counter is cleared to 0000H.

Caution Do not set the count clock as the valid edge of the TI00n pin (PRM0n1 and PRM0n0 = 11). When PRM0n1 and PRM0n0 = 11, TM0n is cleared.

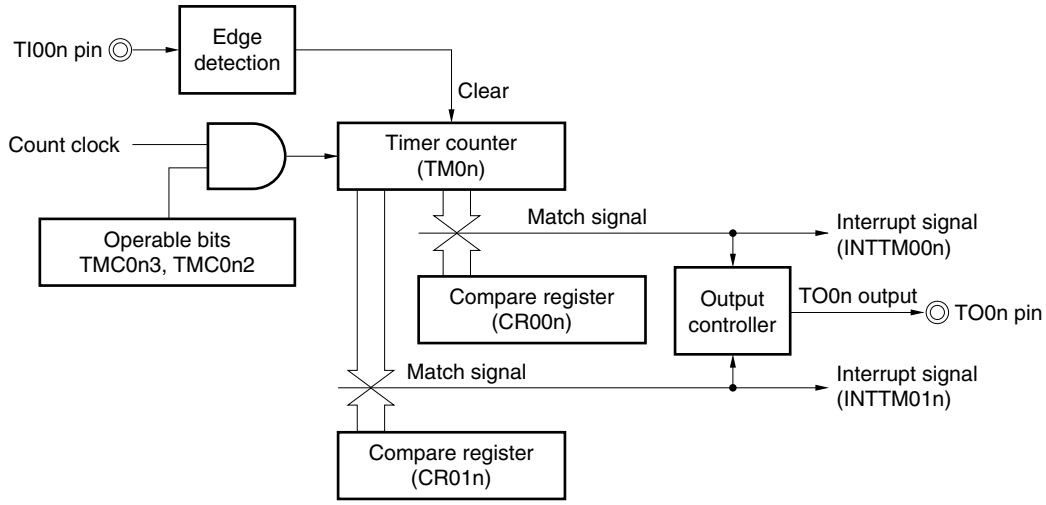
Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

3. n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n) in the 78K0/KC2-C: n = 0).

(1) Operation in clear & start mode entered by TI00n pin valid edge input (CR00n: compare register, CR01n: compare register)

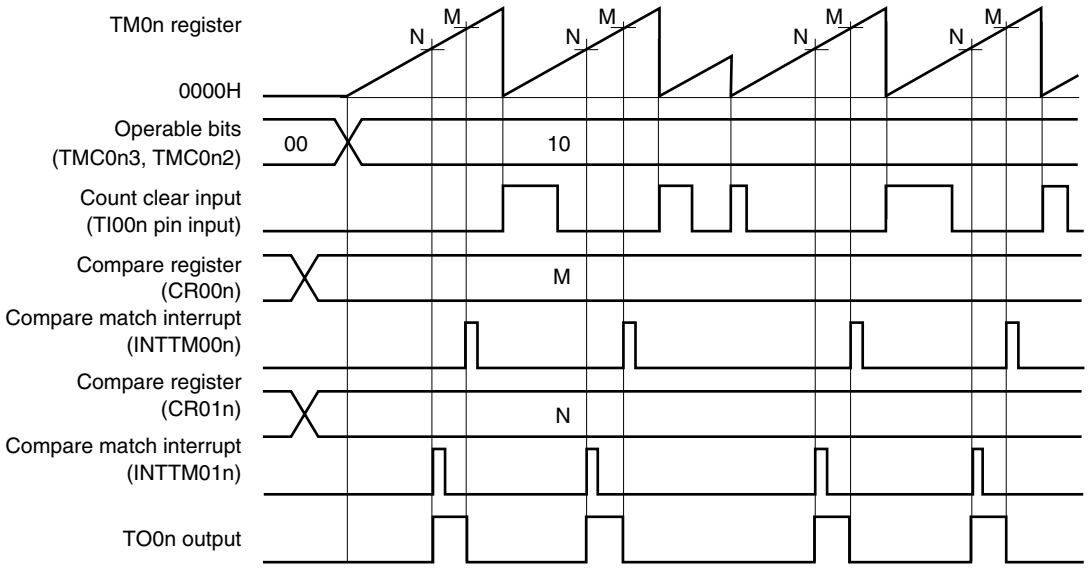
Figure 6-33. Block Diagram of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Compare Register, CR01n: Compare Register)



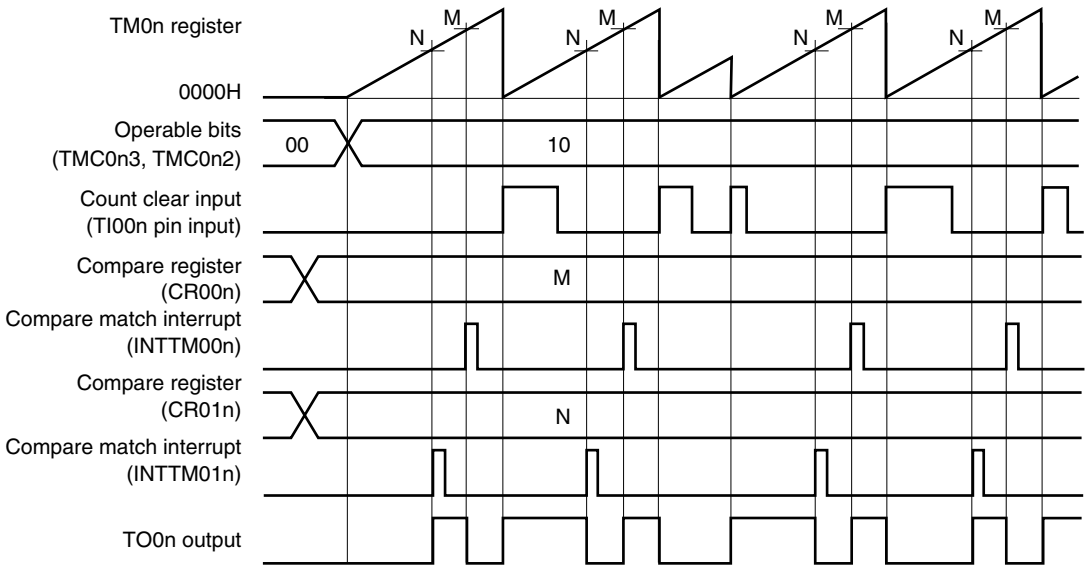
Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n) in the 78K0/KC2-C: n = 0).

Figure 6-34. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Compare Register, CR01n: Compare Register)

(a) TOC0n = 13H, PRM0n = 10H, CRC0n, = 00H, TMC0n = 08H



(b) TOC0n = 13H, PRM0n = 10H, CRC0n, = 00H, TMC0n = 0AH



(a) and (b) differ as follows depending on the setting of bit 1 (TMC0n1) of the 16-bit timer mode control register 0n (TMC0n).

(a) The TO0n output level is inverted when TM0n matches a compare register.

(b) The TO0n output level is inverted when TM0n matches a compare register or when the valid edge of the TI00n pin is detected.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

(2) Operation in clear & start mode entered by TI00n pin valid edge input
(CR00n: compare register, CR01n: capture register)

Figure 6-35. Block Diagram of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Compare Register, CR01n: Capture Register)

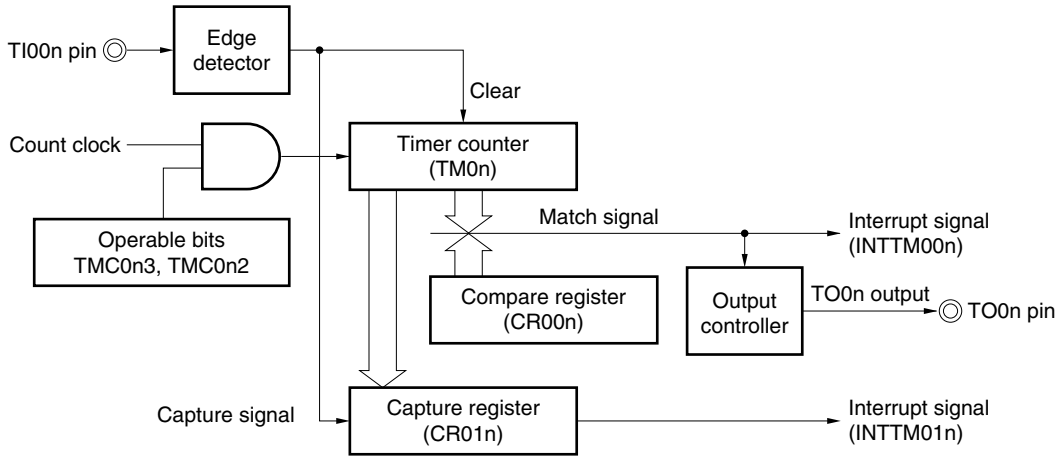
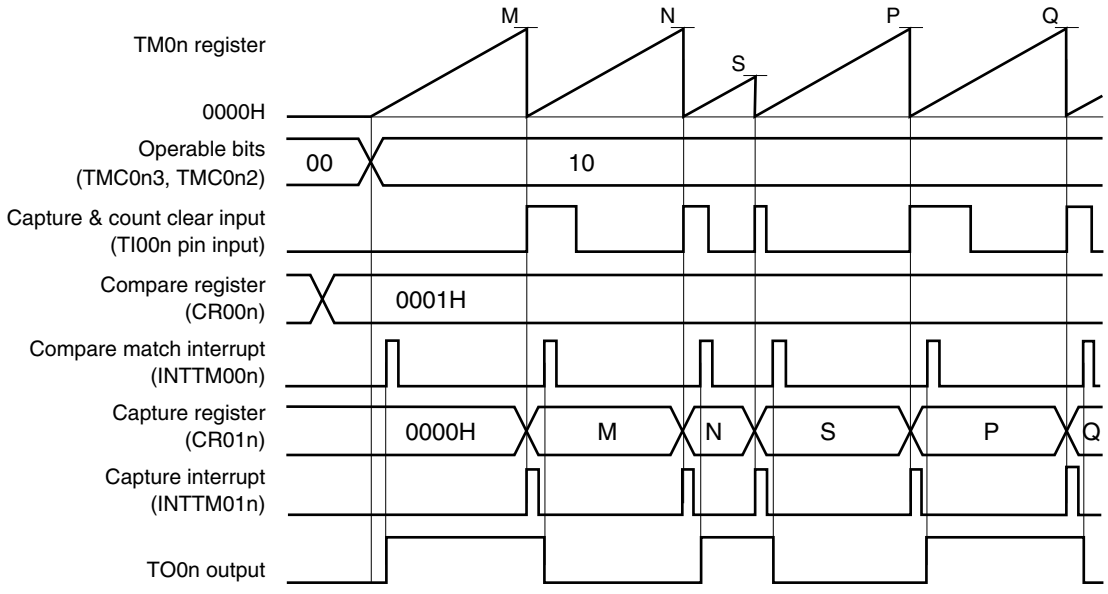


Figure 6-36. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Compare Register, CR01n: Capture Register) (1/2)

(a) TOC0n = 13H, PRM0n = 10H, CRC0n, = 04H, TMC0n = 08H, CR00n = 0001H



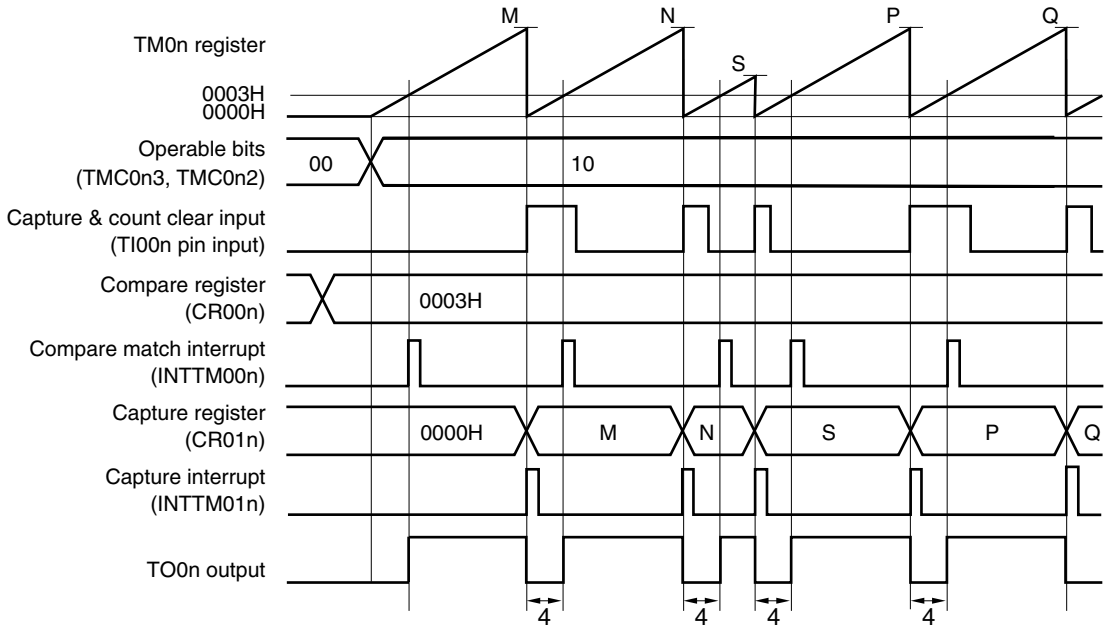
This is an application example where the TO0n output level is inverted when the count value has been captured & cleared.

The count value is captured to CR01n and TM0n is cleared (to 0000H) when the valid edge of the TI00n pin is detected. When the count value of TM0n is 0001H, a compare match interrupt signal (INTTM00n) is generated, and the TO0n output level is inverted.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-36. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Compare Register, CR01n: Capture Register) (2/2)

(b) TOC0n = 13H, PRM0n = 10H, CRC0n, = 04H, TMC0n = 0AH, CR00n = 0003H



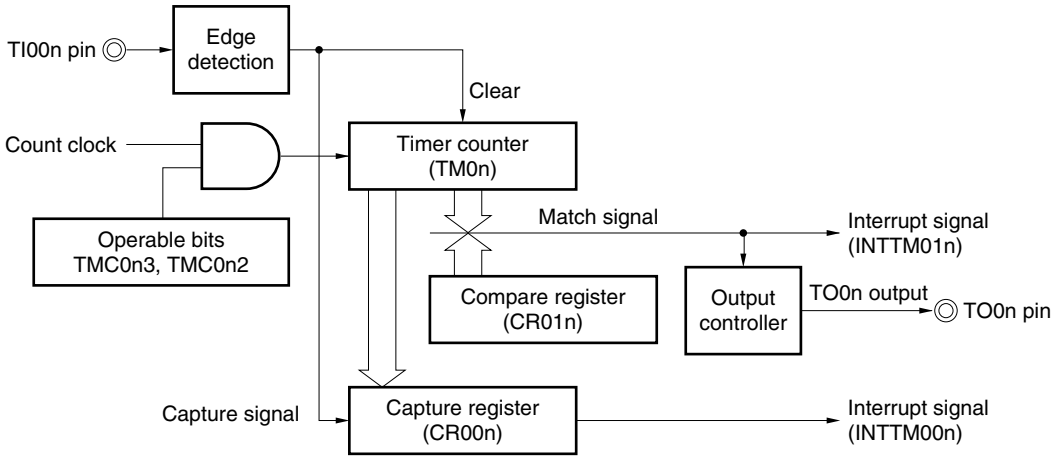
This is an application example where the width set to CR00n (4 clocks in this example) is to be output from the TO0n pin when the count value has been captured & cleared.

The count value is captured to CR01n, a capture interrupt signal (INTTM01n) is generated, TM0n is cleared (to 0000H), and the TO0n output level is inverted when the valid edge of the TI00n pin is detected. When the count value of TM0n is 0003H (four clocks have been counted), a compare match interrupt signal (INTTM00n) is generated and the TO0n output level is inverted.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

(3) Operation in clear & start mode by entered TI00n pin valid edge input (CR00n: capture register, CR01n: compare register)

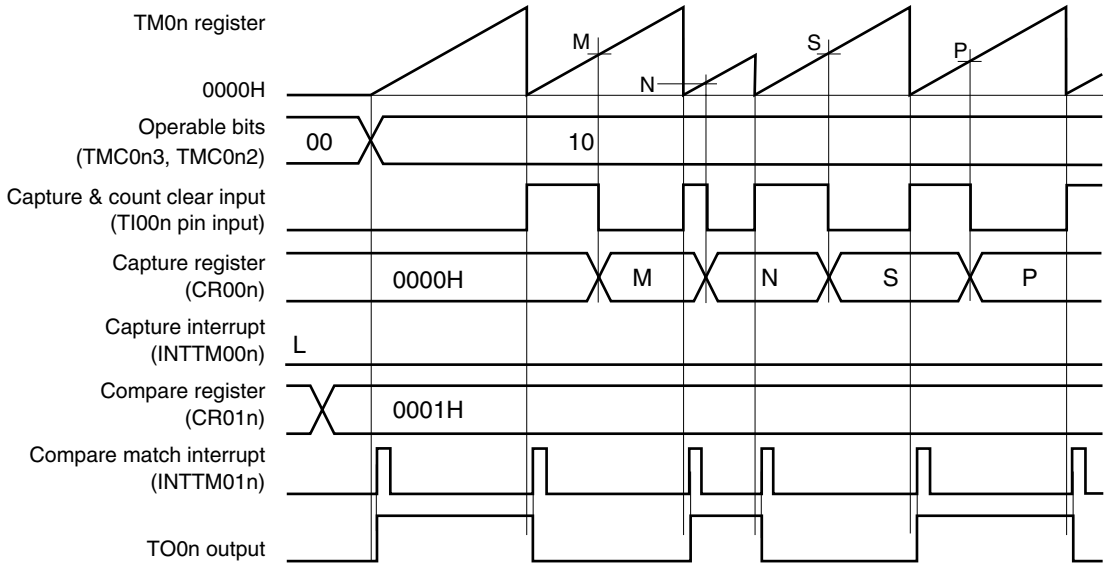
Figure 6-37. Block Diagram of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Capture Register, CR01n: Compare Register)



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n) in the 78K0/KC2-C: n = 0).

Figure 6-38. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Capture Register, CR01n: Compare Register) (1/2)

(a) TOC0n = 13H, PRM0n = 10H, CRC0n, = 03H, TMC0n = 08H, CR01n = 0001H



This is an application example where the TO0n output level is to be inverted when the count value has been captured & cleared.

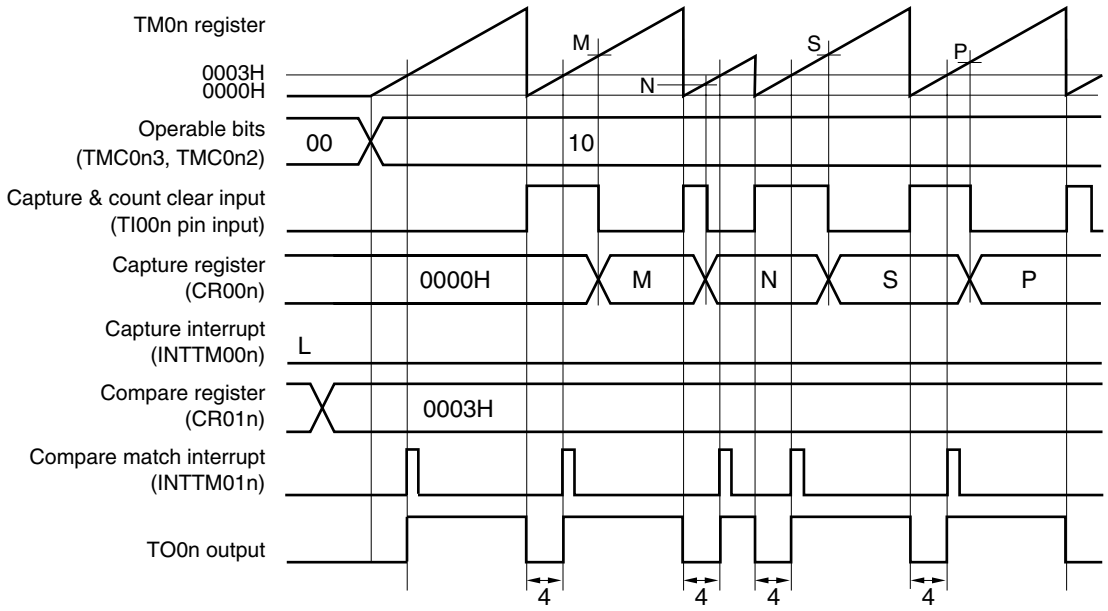
TM0n is cleared at the rising edge detection of the TI00n pin and it is captured to CR00n at the falling edge detection of the TI00n pin.

When bit 1 (CRC0n1) of capture/compare control register 0n (CRC0n) is set to 1, the count value of TM0n is captured to CR00n in the phase reverse to that of the signal input to the TI00n pin, but the capture interrupt signal (INTTM00n) is not generated. However, the INTTM00n signal is generated when the valid edge of the TI01n pin is detected. Mask the INTTM00n signal when it is not used.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-38. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Capture Register, CR01n: Compare Register) (2/2)

(b) TOC0n = 13H, PRM0n = 10H, CRC0n, = 03H, TMC0n = 0AH, CR01n = 0003H



This is an application example where the width set to CR01n (4 clocks in this example) is to be output from the TO0n pin when the count value has been captured & cleared.

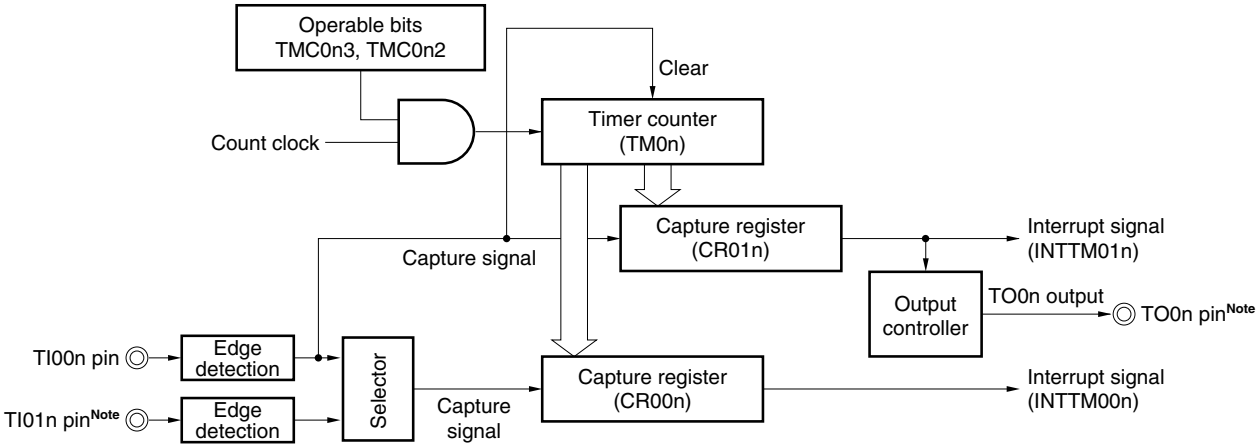
TM0n is cleared (to 0000H) at the rising edge detection of the TI00n pin and captured to CR00n at the falling edge detection of the TI00n pin. The TO0n output level is inverted when TM0n is cleared (to 0000H) because the rising edge of the TI00n pin has been detected or when the value of TM0n matches that of a compare register (CR01n).

When bit 1 (CRC0n1) of capture/compare control register 0n (CRC0n) is 1, the count value of TM0n is captured to CR00n in the phase reverse to that of the input signal of the TI00n pin, but the capture interrupt signal (INTTM00n) is not generated. However, the INTTM00n interrupt is generated when the valid edge of the TI01n pin is detected. Mask the INTTM00n signal when it is not used.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

(4) Operation in clear & start mode entered by TI00n pin valid edge input (CR00n: capture register, CR01n: capture register)

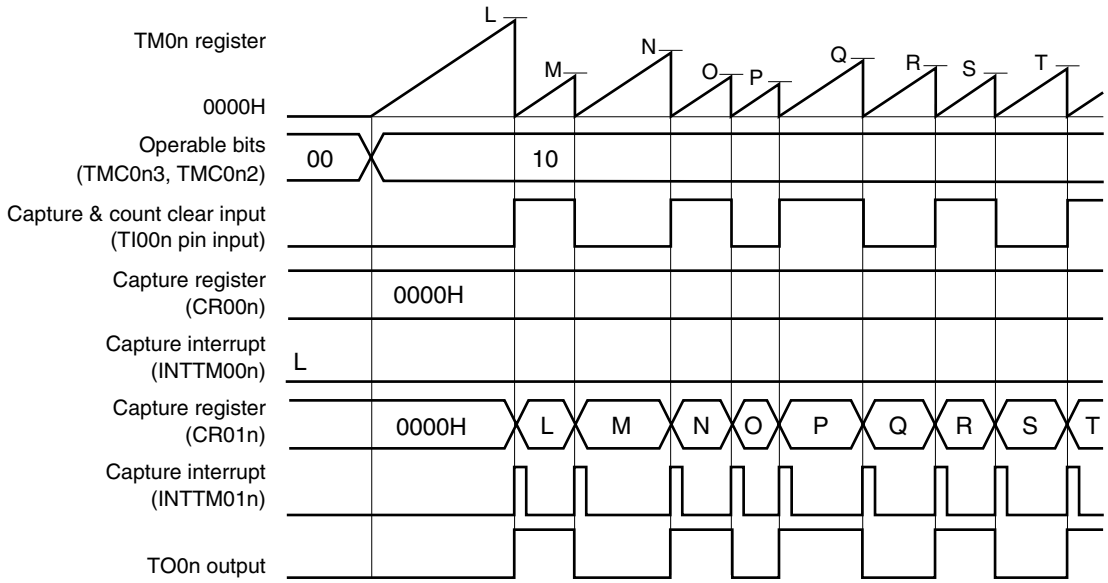
Figure 6-39. Block Diagram of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Capture Register, CR01n: Capture Register)



Note The timer output (TO0n) cannot be used when detecting the valid edge of the TI01n pin is used.

Figure 6-40. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Capture Register, CR01n: Capture Register) (1/3)

(a) TOC0n = 13H, PRM0n = 30H, CRC0n = 05H, TMC0n = 0AH



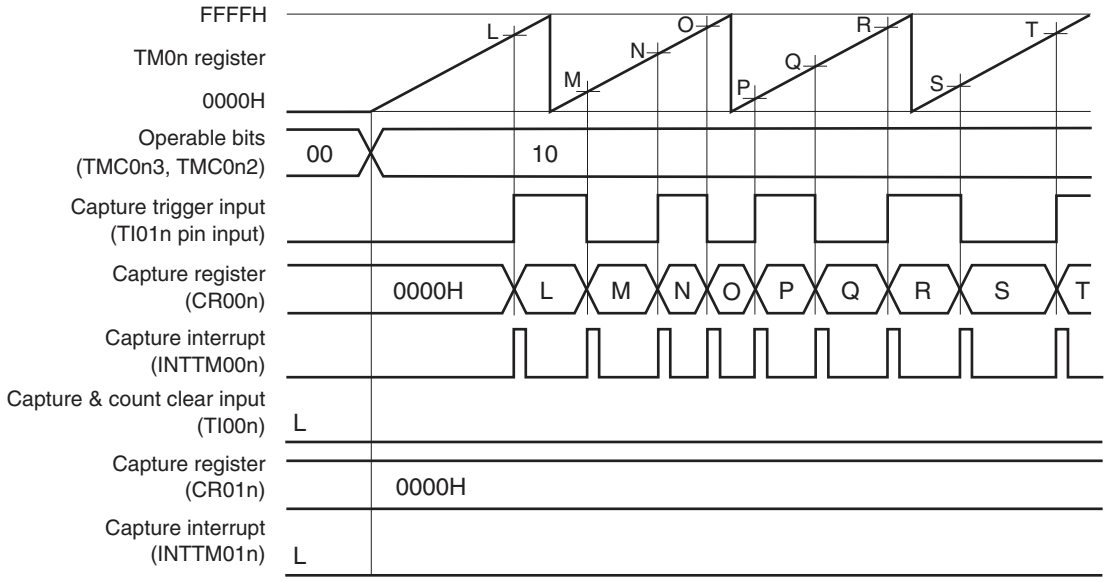
This is an application example where the count value is captured to CR01n, TM0n is cleared, and the TO0n output is inverted when the rising or falling edge of the TI00n pin is detected.

When the edge of the TI01n pin is detected, an interrupt signal (INTTM00n) is generated. Mask the INTTM00n signal when it is not used.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-40. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Capture Register, CR01n: Capture Register) (2/3)

(b) TOC0n = 13H, PRM0n = C0H, CRC0n = 05H, TMC0n = 0AH

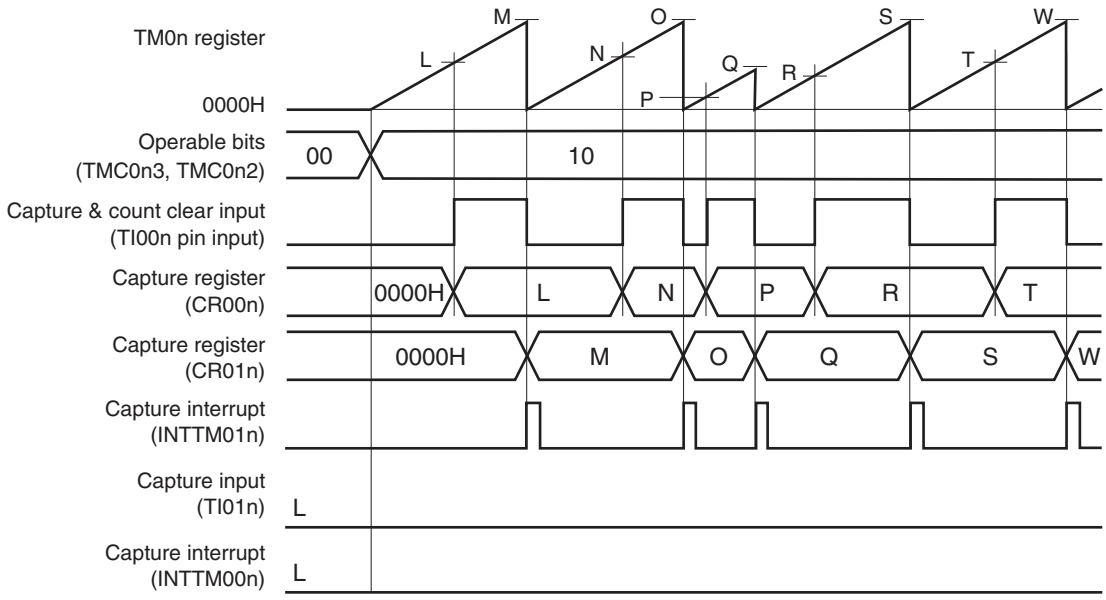


This is a timing example where an edge is not input to the TI00n pin, in an application where the count value is captured to CR00n when the rising or falling edge of the TI01n pin is detected.

Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-40. Timing Example of Clear & Start Mode Entered by TI00n Pin Valid Edge Input (CR00n: Capture Register, CR01n: Capture Register) (3/3)

(c) TOC0n = 13H, PRM0n = 00H, CRC0n = 07H, TMC0n = 0AH



This is an application example where the pulse width of the signal input to the TI00n pin is measured.
By setting CRC0n, the count value can be captured to CR00n in the phase reverse to the falling edge of the TI00n pin (i.e., rising edge) and to CR01n at the falling edge of the TI00n pin.
The high- and low-level widths of the input pulse can be calculated by the following expressions.

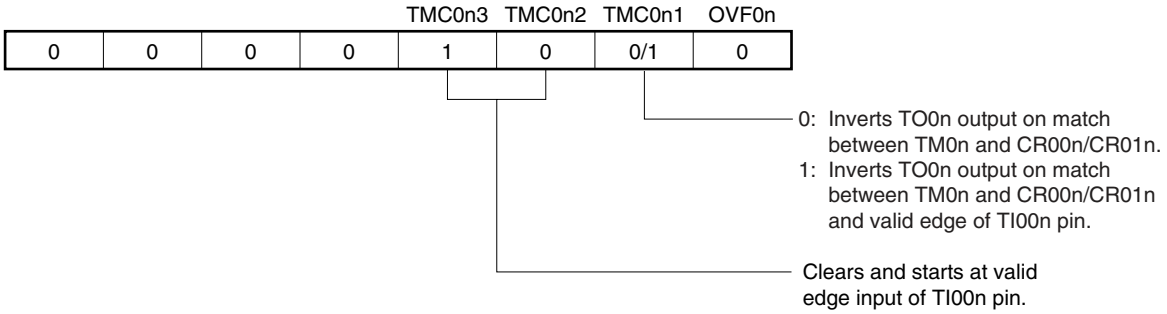
- High-level width = [CR01n value] – [CR00n value] × [Count clock cycle]
- Low-level width = [CR00n value] × [Count clock cycle]

If the reverse phase of the TI00n pin is selected as a trigger to capture the count value to CR00n, the INTTM00n signal is not generated. Read the values of CR00n and CR01n to measure the pulse width immediately after the INTTM01n signal is generated.
However, if the valid edge specified by bits 6 and 5 (ES1n1 and ES1n0) of prescaler mode register 0n (PRM0n) is input to the TI01n pin, the count value is not captured but the INTTM00n signal is generated. To measure the pulse width of the TI00n pin, mask the INTTM00n signal when it is not used.

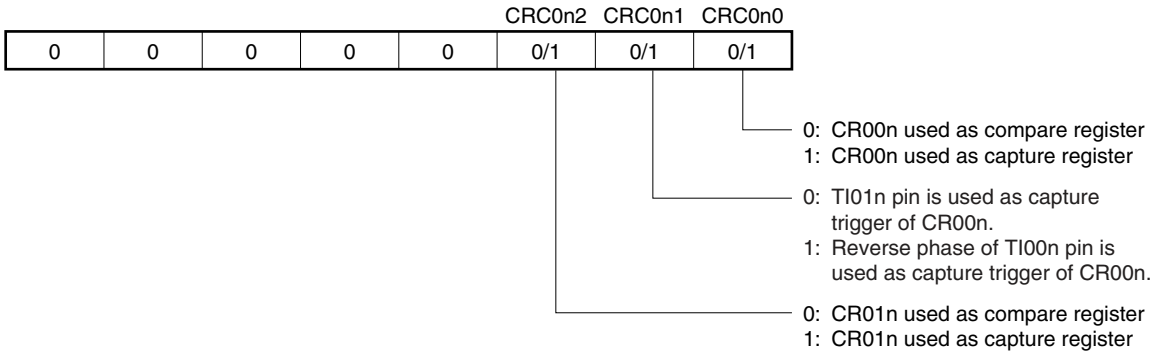
Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-41. Example of Register Settings in Clear & Start Mode Entered by TI00n Pin Valid Edge Input (1/2)

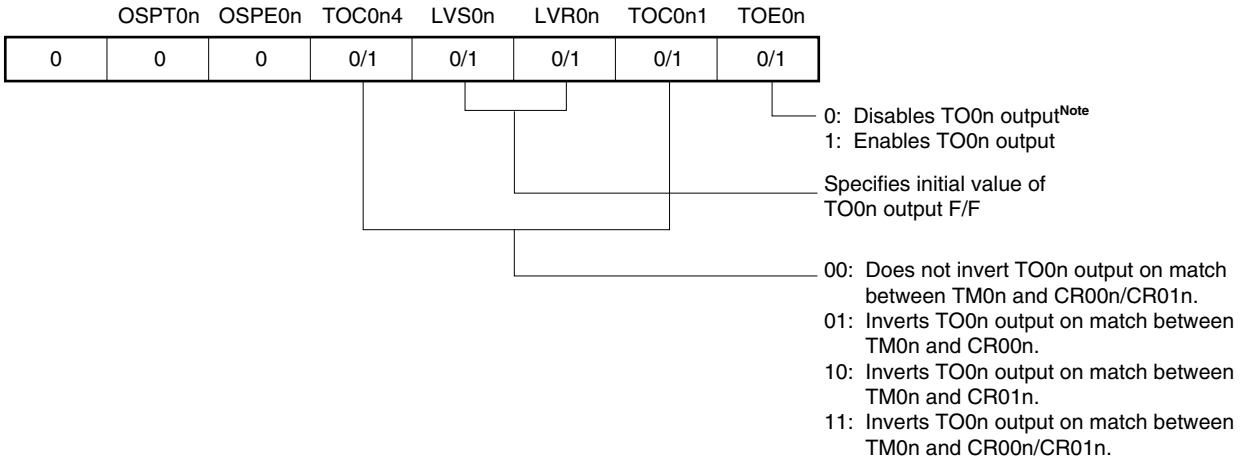
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)

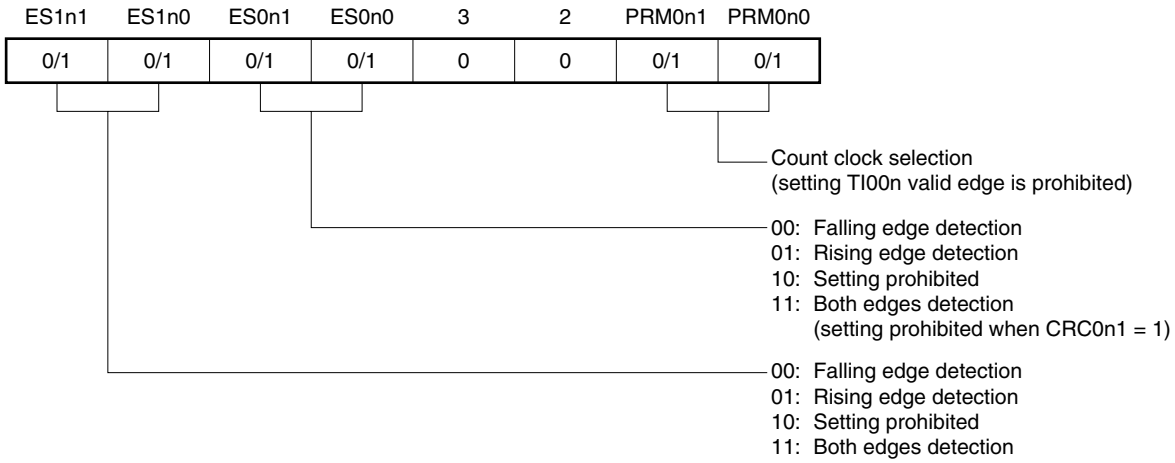


Note The timer output (TO0n) cannot be used when detecting the valid edge of the TI01n pin is used.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-41. Example of Register Settings in Clear & Start Mode Entered by TI00n Pin Valid Edge Input (2/2)

(d) Prescaler mode register 0n (PRM0n)



(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

When this register is used as a compare register and when its value matches the count value of TM0n, an interrupt signal (INTTM00n) is generated. The count value of TM0n is not cleared.

To use this register as a capture register, select either the TI00n or TI01n pin[Note] input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM0n is stored in CR00n.

Note The timer output (TO0n) cannot be used when detection of the valid edge of the TI01n pin is used.

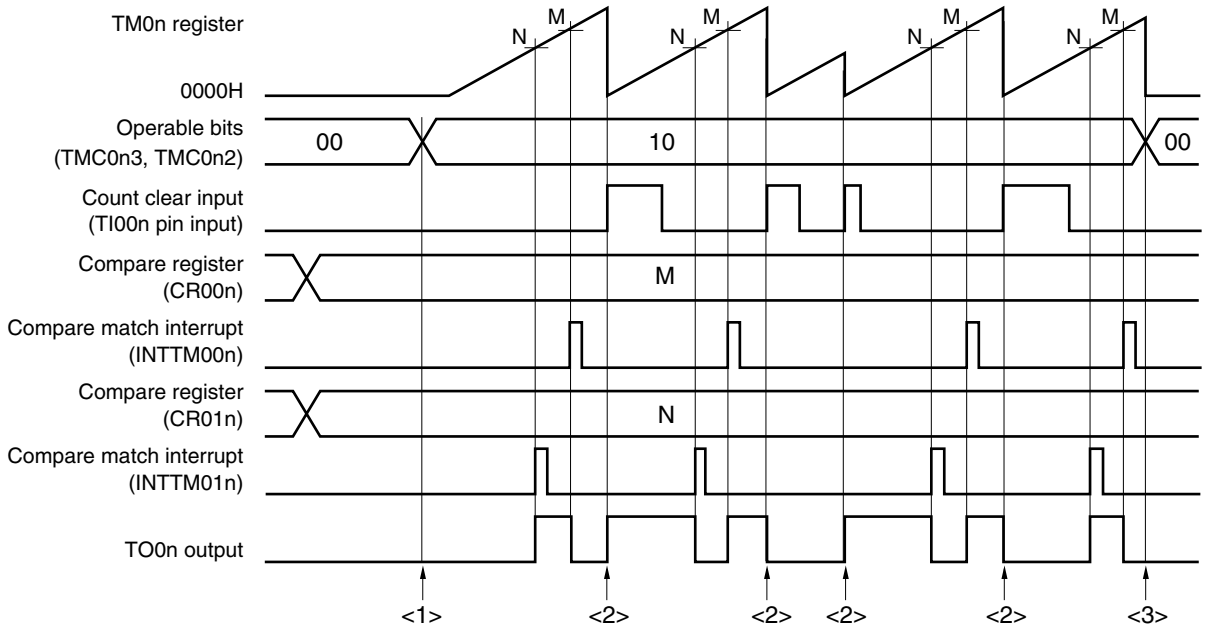
(g) 16-bit capture/compare register 01n (CR01n)

When this register is used as a compare register and when its value matches the count value of TM0n, an interrupt signal (INTTM01n) is generated. The count value of TM0n is not cleared.

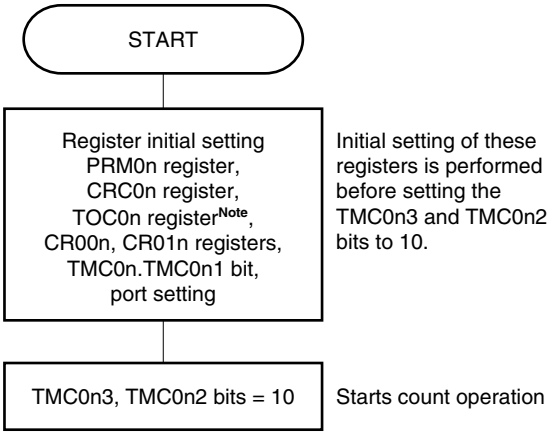
When this register is used as a capture register, the TI00n pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM0n is stored in CR01n.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n) in the 78K0/KC2-C: n = 0).

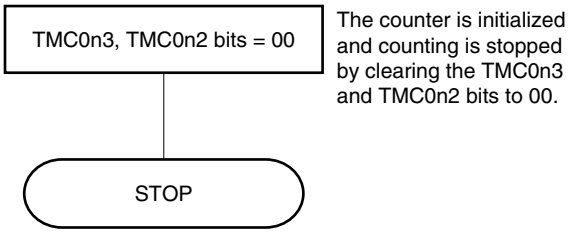
Figure 6-42. Example of Software Processing in Clear & Start Mode Entered by TI00n Pin Valid Edge Input



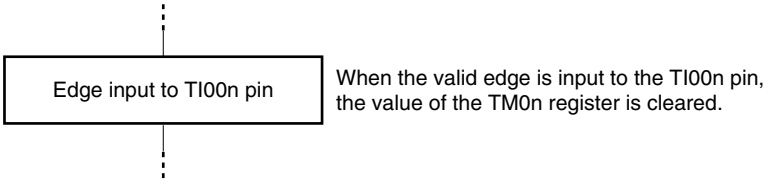
<1> Count operation start flow



<3> Count operation stop flow



<2> TM0n register clear & start flow



Note Care must be exercised when setting TOC0n. For details, see **6.3 (3) 16-bit timer output control register 0n (TOC0n)**.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

6.4.5 Free-running timer operation

When bits 3 and 2 (TMC0n3 and TMC0n2) of 16-bit timer mode control register 0n (TMC0n) are set to 01 (free-running timer mode), 16-bit timer/event counter 0n continues counting up in synchronization with the count clock. When it has counted up to FFFFH, the overflow flag (OVF0n) is set to 1 at the next clock, and TM0n is cleared (to 0000H) and continues counting. Clear OVF0n to 0 by executing the CLR instruction via software.

The following three types of free-running timer operations are available.

- Both CR00n and CR01n are used as compare registers.
- One of CR00n or CR01n is used as a compare register and the other is used as a capture register.
- Both CR00n and CR01n are used as capture registers.

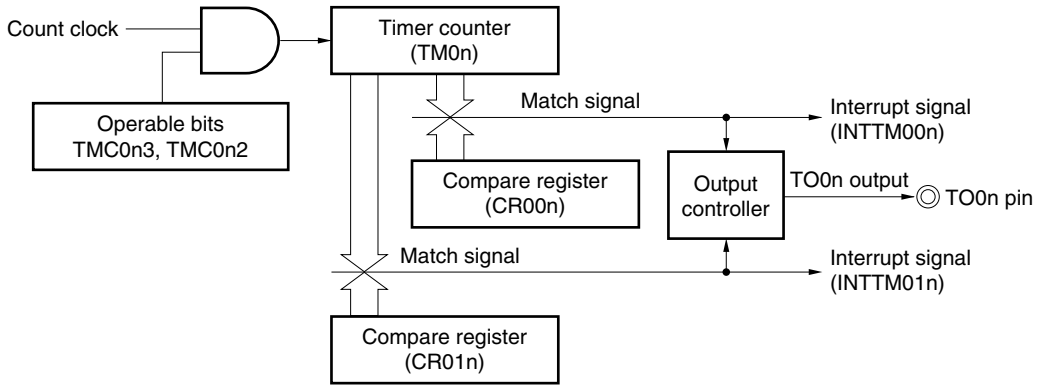
Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

(1) Free-running timer mode operation

(CR00n: compare register, CR01n: compare register)

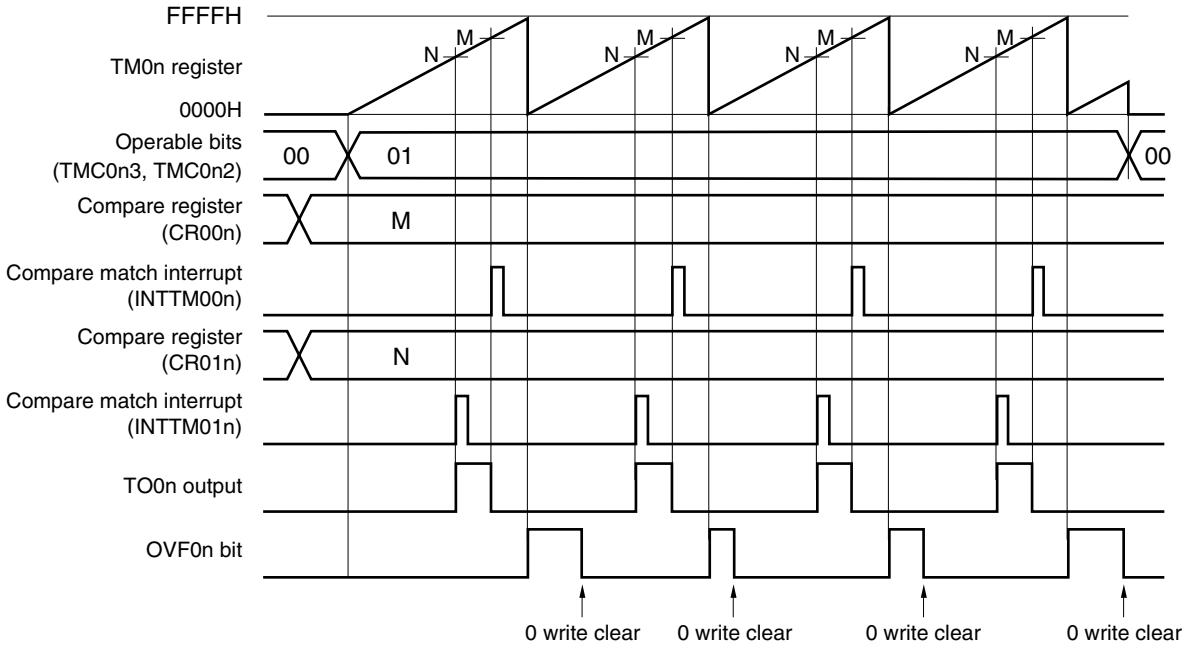
Figure 6-43. Block Diagram of Free-Running Timer Mode
(CR00n: Compare Register, CR01n: Compare Register)



Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n) in the 78K0/KC2-C: n = 0).

Figure 6-44. Timing Example of Free-Running Timer Mode
(CR00n: Compare Register, CR01n: Compare Register)

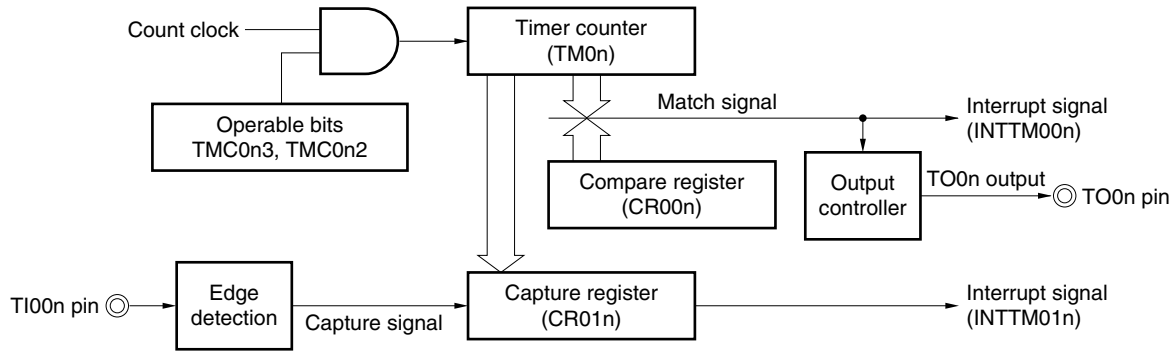
• TOC0n = 13H, PRM0n = 00H, CRC0n = 00H, TMC0n = 04H



This is an application example where two compare registers are used in the free-running timer mode.
The TO0n output level is reversed each time the count value of TM0n matches the set value of CR00n or CR01n. When the count value matches the register value, the INTTM00n or INTTM01n signal is generated.

(2) Free-running timer mode operation
(CR00n: compare register, CR01n: capture register)

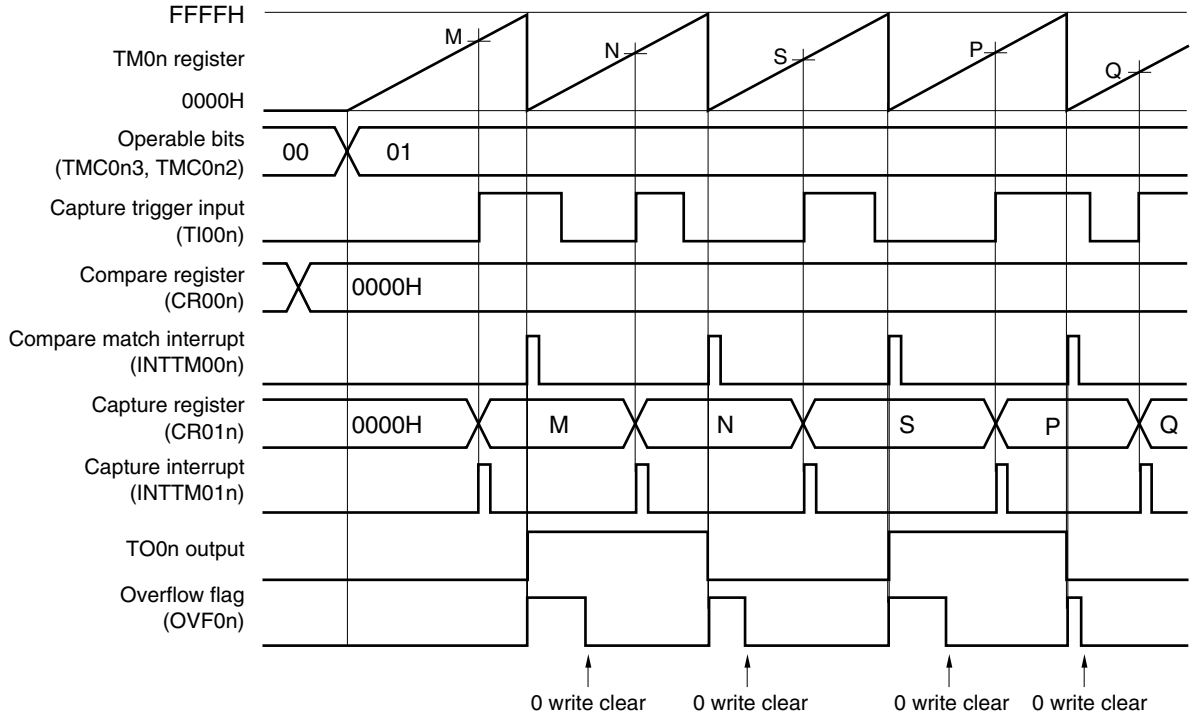
Figure 6-45. Block Diagram of Free-Running Timer Mode
(CR00n: Compare Register, CR01n: Capture Register)



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-46. Timing Example of Free-Running Timer Mode
(CR00n: Compare Register, CR01n: Capture Register)

• TOC0n = 13H, PRM0n = 10H, CRC0n = 04H, TMC0n = 04H



This is an application example where a compare register and a capture register are used at the same time in the free-running timer mode.

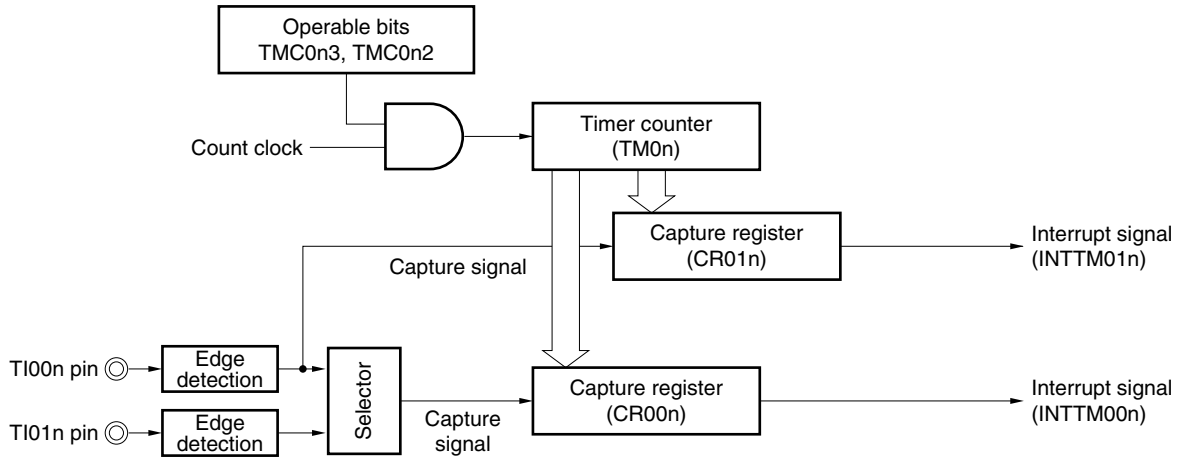
In this example, the INTTM00n signal is generated and the TO0n output level is reversed each time the count value of TM0n matches the set value of CR00n (compare register). In addition, the INTTM01n signal is generated and the count value of TM0n is captured to CR01n each time the valid edge of the TI00n pin is detected.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

(3) Free-running timer mode operation

(CR00n: capture register, CR01n: capture register)

Figure 6-47. Block Diagram of Free-Running Timer Mode
(CR00n: Capture Register, CR01n: Capture Register)



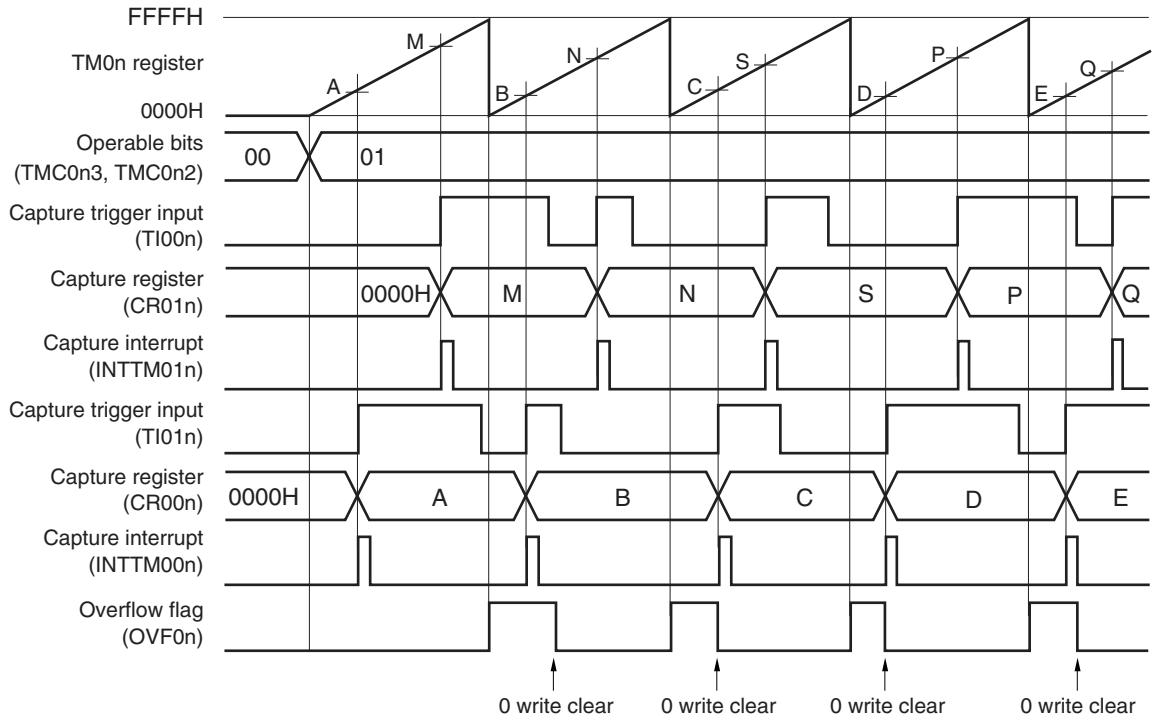
Remarks 1. If both CR00n and CR01n are used as capture registers in the free-running timer mode, the TO0n output level is not inverted.

However, it can be inverted each time the valid edge of the TI00n pin is detected if bit 1 (TMC0n1) of 16-bit timer mode control register 0n (TMC0n) is set to 1.

2. n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n) in the 78K0/KC2-C: n = 0).

Figure 6-48. Timing Example of Free-Running Timer Mode (CR00n: Capture Register, CR01n: Capture Register) (1/2)

(a) TOC0n = 13H, PRM0n = 50H, CRC0n = 05H, TMC0n = 04H

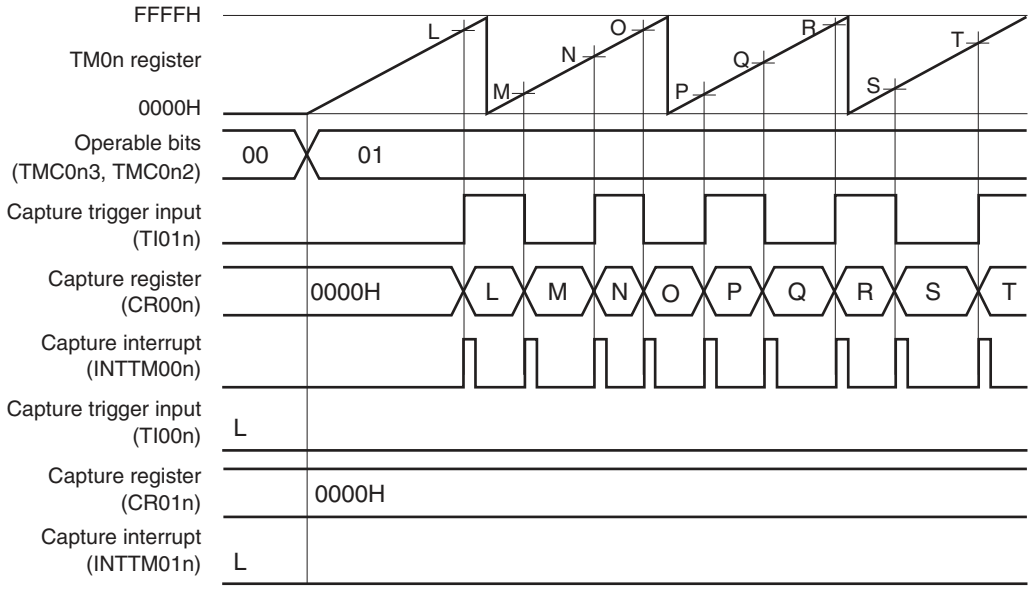


This is an application example where the count values that have been captured at the valid edges of separate capture trigger signals are stored in separate capture registers in the free-running timer mode.
The count value is captured to CR01n when the valid edge of the TI00n pin input is detected and to CR00n when the valid edge of the TI01n pin input is detected.

Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-48. Timing Example of Free-Running Timer Mode (CR00n: Capture Register, CR01n: Capture Register) (2/2)

(b) TOC0n = 13H, PRM0n = C0H, CRC0n = 05H, TMC0n = 04H



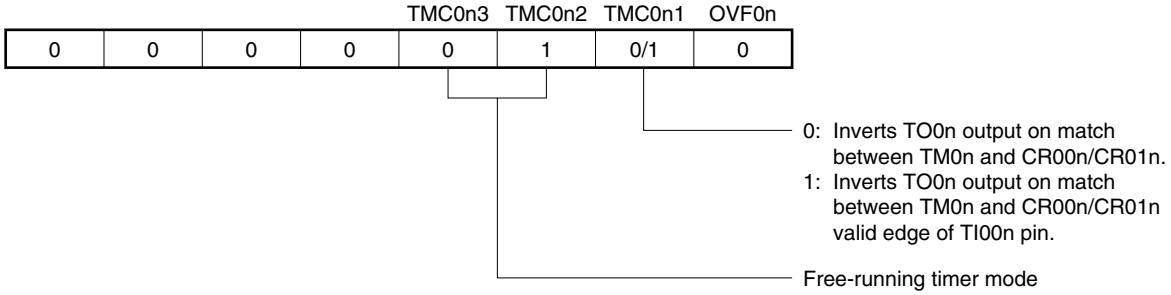
This is an application example where both the edges of the TI01n pin are detected and the count value is captured to CR00n in the free-running timer mode.

When both CR00n and CR01n are used as capture registers and when the valid edge of only the TI01n pin is to be detected, the count value cannot be captured to CR01n.

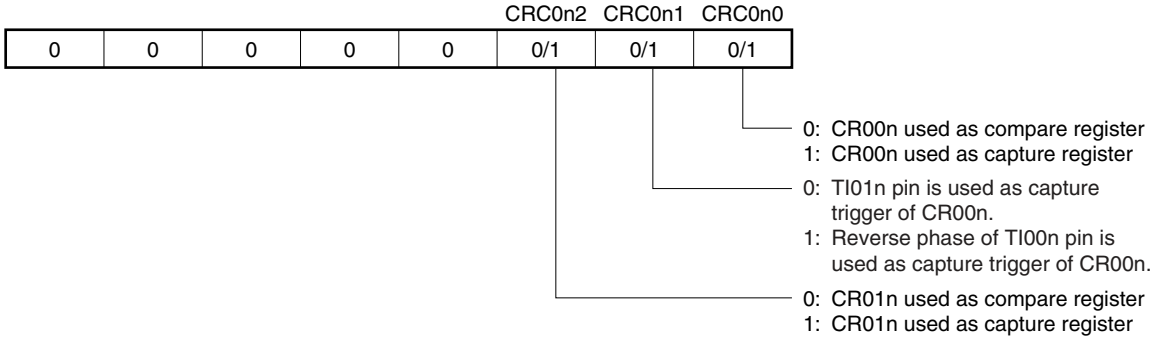
Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-49. Example of Register Settings in Free-Running Timer Mode (1/2)

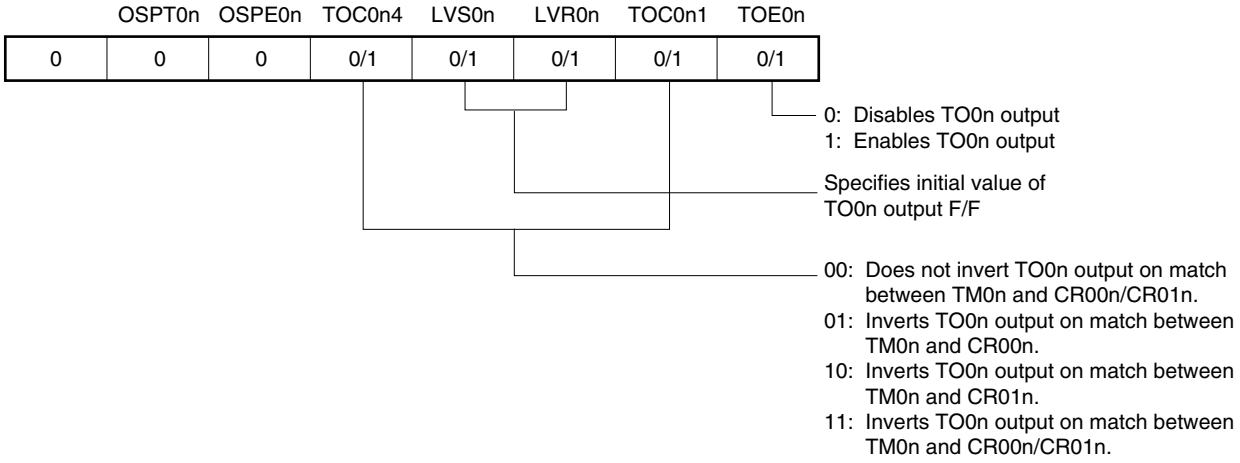
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



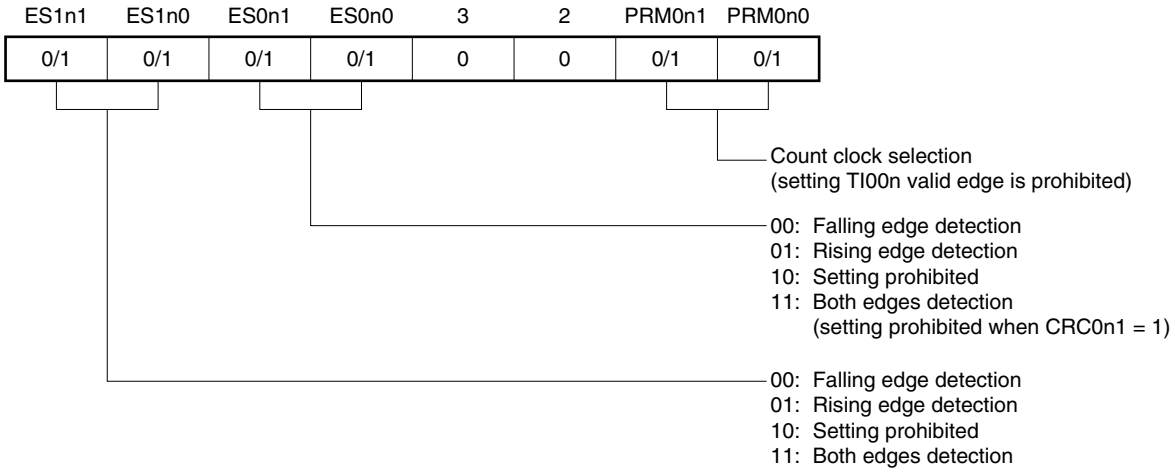
(c) 16-bit timer output control register 0n (TOC0n)



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-49. Example of Register Settings in Free-Running Timer Mode (2/2)

(d) Prescaler mode register 0n (PRM0n)



(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

When this register is used as a compare register and when its value matches the count value of TM0n, an interrupt signal (INTTM00n) is generated. The count value of TM0n is not cleared.

To use this register as a capture register, select either the TI00n or TI01n pin input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM0n is stored in CR00n.

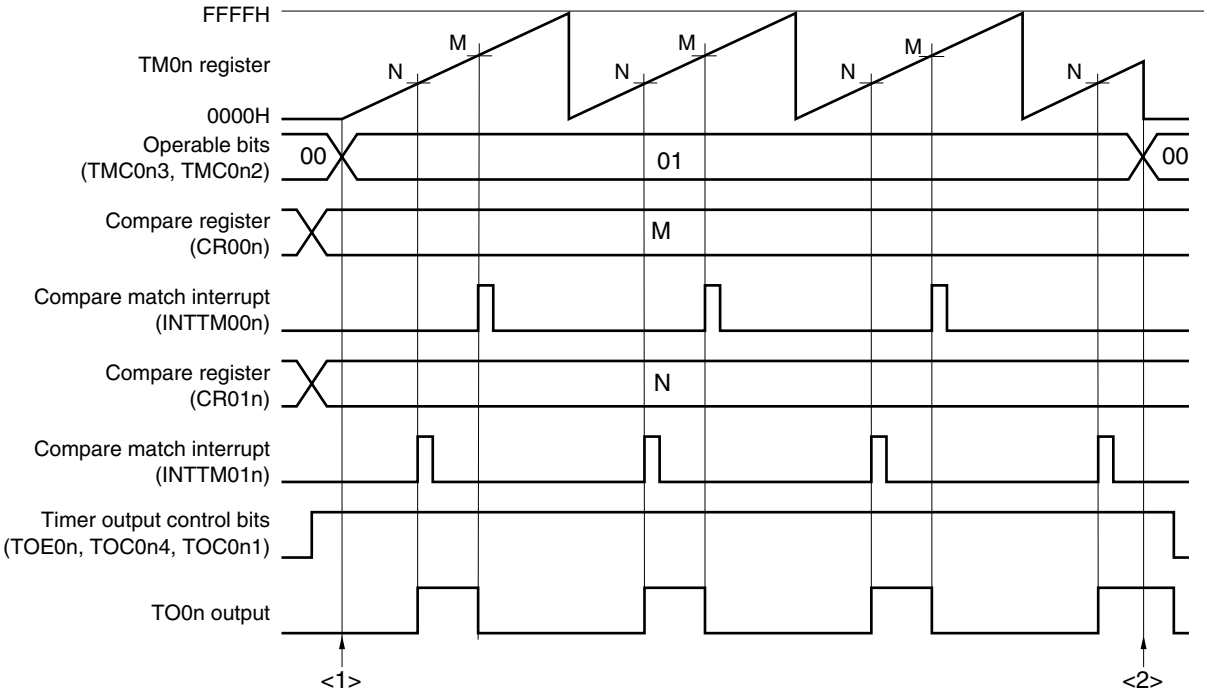
(g) 16-bit capture/compare register 01n (CR01n)

When this register is used as a compare register and when its value matches the count value of TM0n, an interrupt signal (INTTM01n) is generated. The count value of TM0n is not cleared.

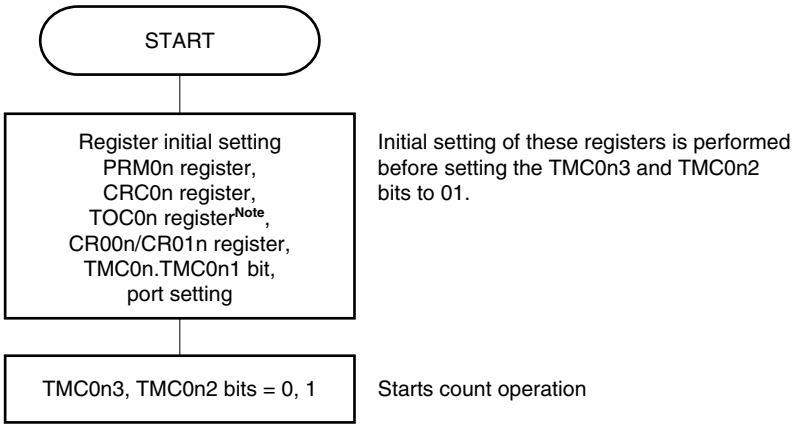
When this register is used as a capture register, the TI00n pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM0n is stored in CR01n.

Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n), and CRC0n register in the 78K0/KC2-C: n = 0).

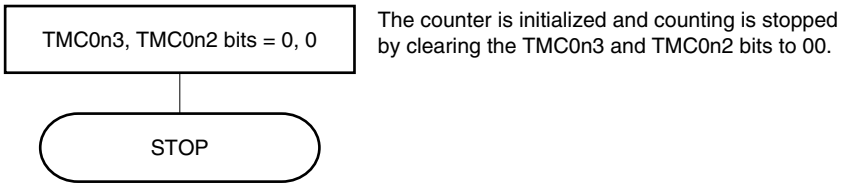
Figure 6-50. Example of Software Processing in Free-Running Timer Mode



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC0n. For details, see **6.3 (3) 16-bit timer output control register 0n (TOC0n)**.

Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

6.4.6 PPG output operation

A square wave having a pulse width set in advance by CR01n is output from the TO0n pin as a PPG (Programmable Pulse Generator) signal during a cycle set by CR00n when bits 3 and 2 (TMC0n3 and TMC0n2) of 16-bit timer mode control register 0n (TMC0n) are set to 11 (clear & start upon a match between TM0n and CR00n).

The pulse cycle and duty factor of the pulse generated as the PPG output are as follows.

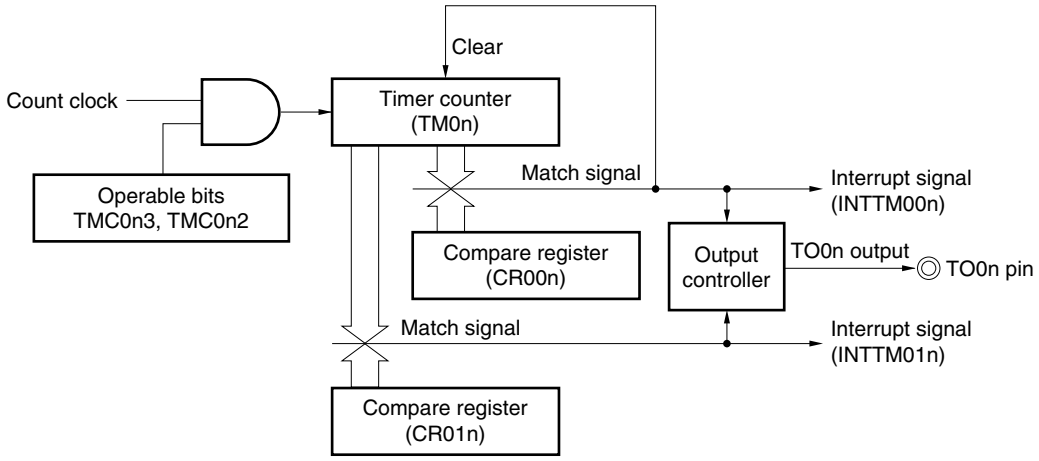
- Pulse cycle = (Set value of CR00n + 1) × Count clock cycle
- Duty = (Set value of CR01n + 1) / (Set value of CR00n + 1)

Caution To change the duty factor (value of CR01n) during operation, see 6.5.1 Rewriting CR01n during TM0n operation.

Remarks 1. For the setting of I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

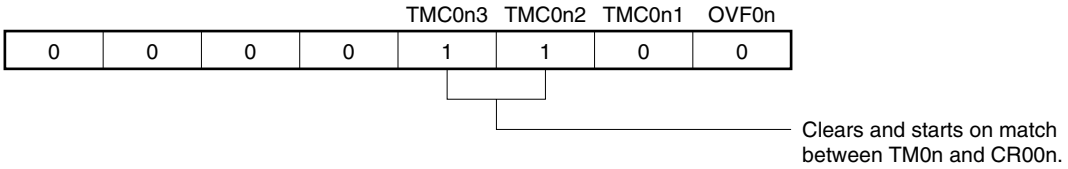
Figure 6-51. Block Diagram of PPG Output Operation



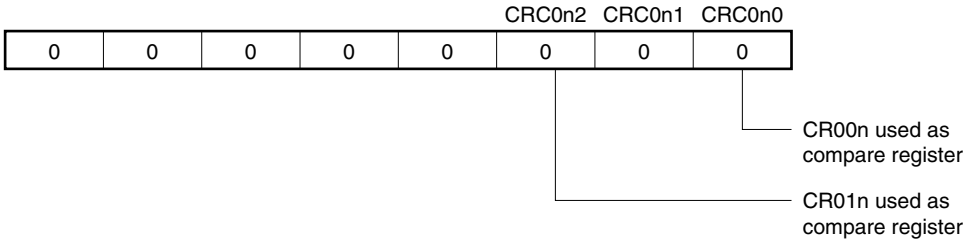
Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n) in the 78K0/KC2-C: n = 0).

Figure 6-52. Example of Register Settings for PPG Output Operation (1/2)

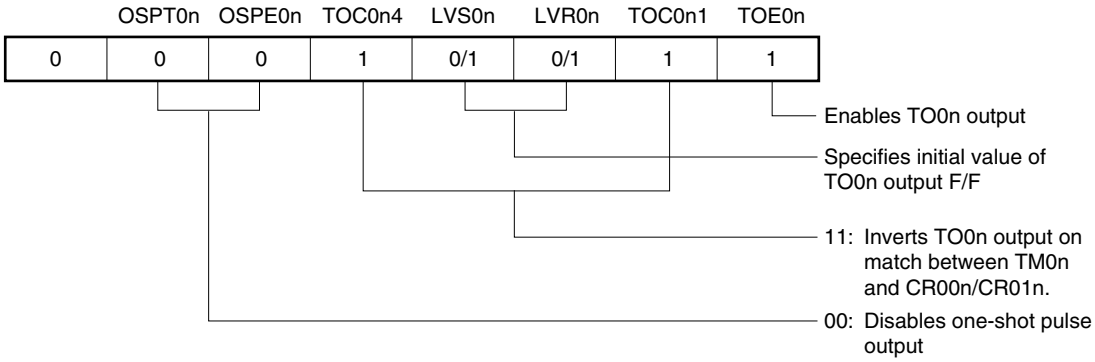
(a) 16-bit timer mode control register 0n (TMC0n)



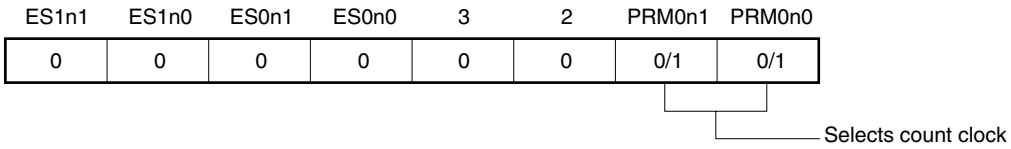
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)



(d) Prescaler mode register 0n (PRM0n)



Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-46. Example of Register Settings for PPG Output Operation (2/2)

(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

An interrupt signal (INTTM00n) is generated when the value of this register matches the count value of TM0n. The count value of TM0n is cleared.

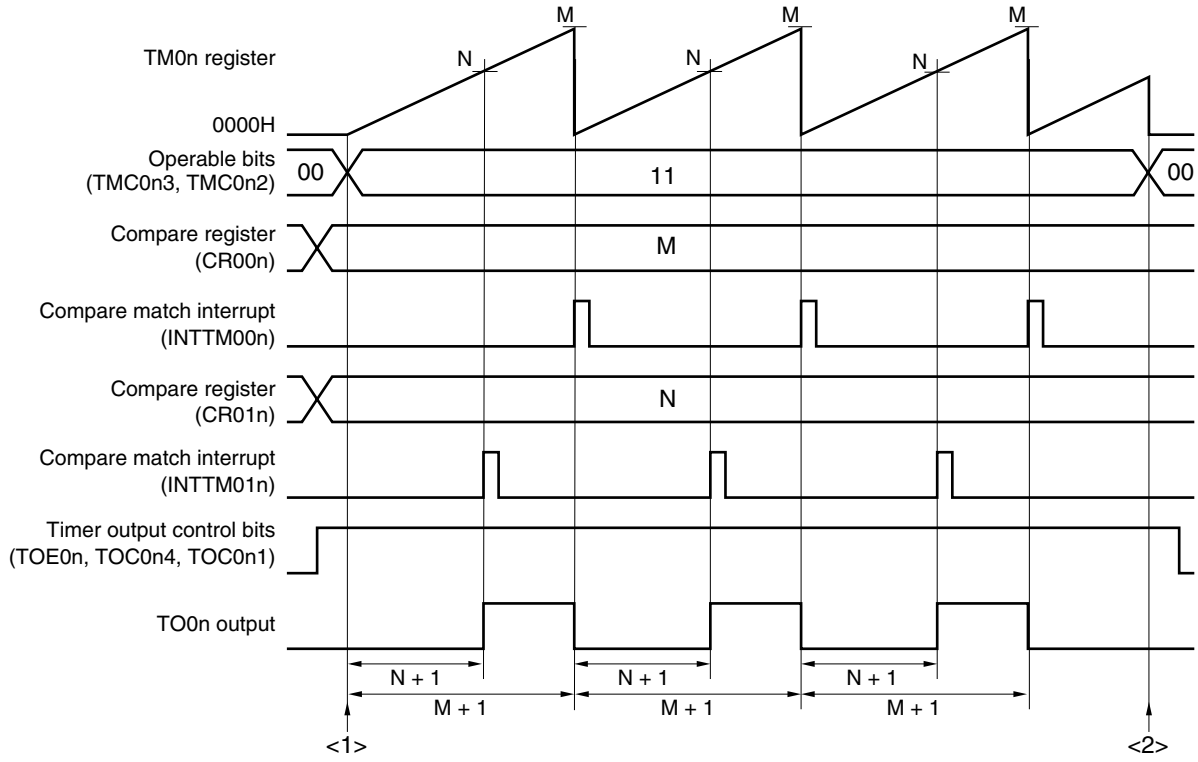
(g) 16-bit capture/compare register 01n (CR01n)

An interrupt signal (INTTM01n) is generated when the value of this register matches the count value of TM0n. The count value of TM0n is not cleared.

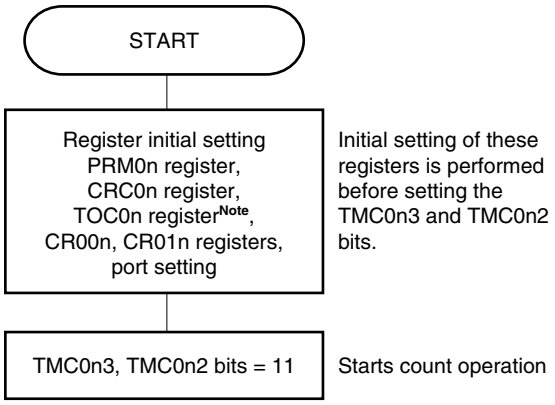
Caution Set values to CR00n and CR01n such that the condition 0000H ≤ CR01n < CR00n ≤ FFFFH is satisfied.

Remark n = 0 to 2

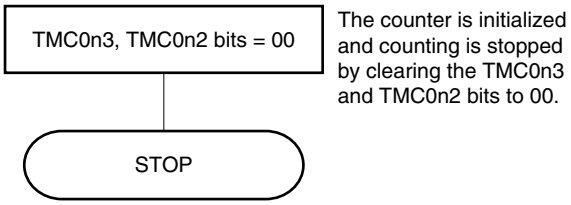
Figure 6-53. Example of Software Processing for PPG Output Operation



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC0n. For details, see **6.3 (3) 16-bit timer output control register 0n (TOC0n)**.

Remarks 1. PPG pulse cycle = (M + 1) × Count clock cycle
PPG duty = (N + 1)/(M + 1)

2. n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

6.4.7 One-shot pulse output operation

A one-shot pulse can be output by setting bits 3 and 2 (TMC0n3 and TMC0n2) of the 16-bit timer mode control register 0n (TMC0n) to 01 (free-running timer mode) or to 10 (clear & start mode entered by the TI00n pin valid edge) and setting bit 5 (OSPE0n) of 16-bit timer output control register 0n (TOC0n) to 1.

When bit 6 (OSPT0n) of TOC0n is set to 1 or when the valid edge is input to the TI00n pin during timer operation, clearing & starting of TM0n is triggered, and a pulse of the difference between the values of CR00n and CR01n is output only once from the TO0n pin.

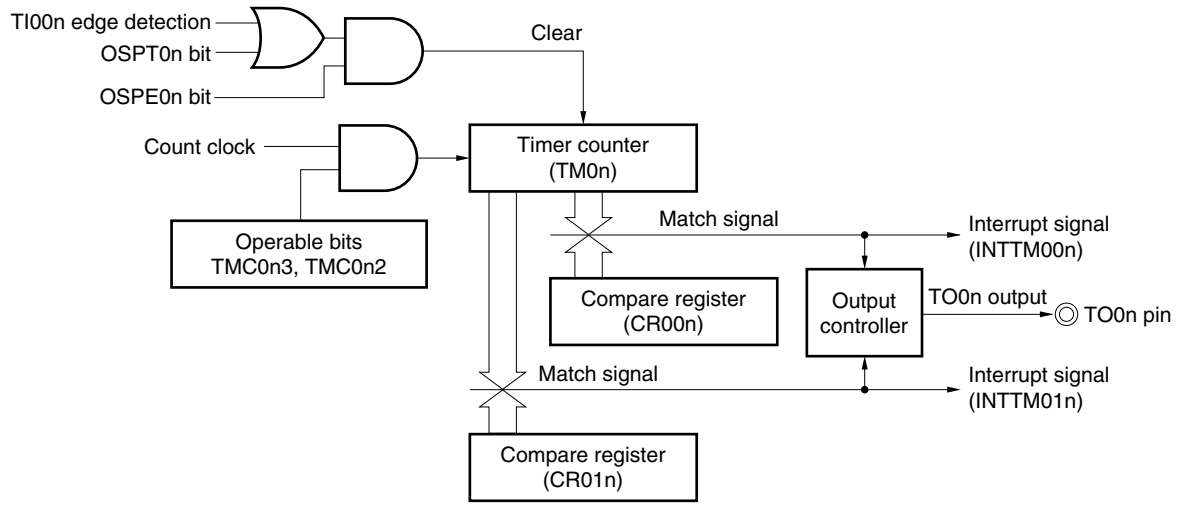
Cautions 1. Do not input the trigger again (setting OSPT0n to 1 or detecting the valid edge of the TI00n pin) while the one-shot pulse is output. To output the one-shot pulse again, generate the trigger after the current one-shot pulse output has completed.

2. To use only the setting of OSPT0n to 1 as the trigger of one-shot pulse output, do not change the level of the TI00n pin or its alternate function port pin. Otherwise, the pulse will be unexpectedly output.

Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

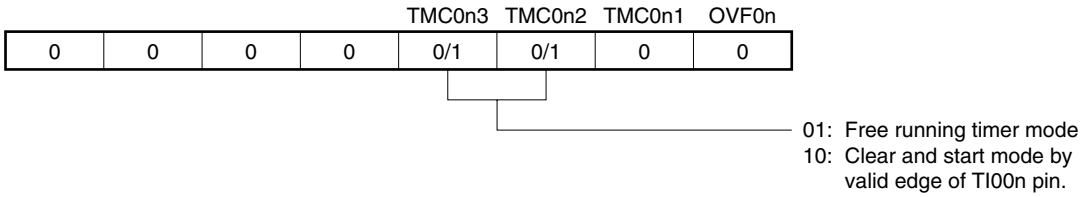
Figure 6-54. Block Diagram of One-Shot Pulse Output Operation



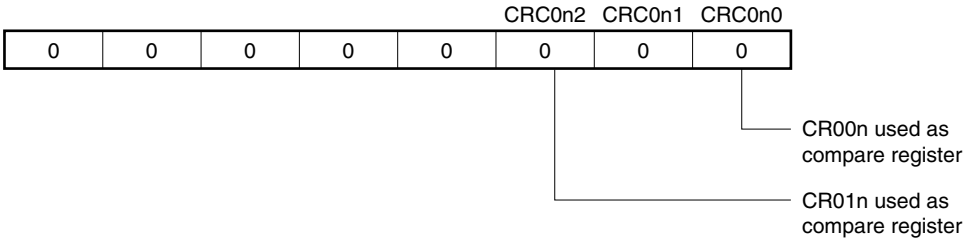
Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-55. Example of Register Settings for One-Shot Pulse Output Operation (1/2)

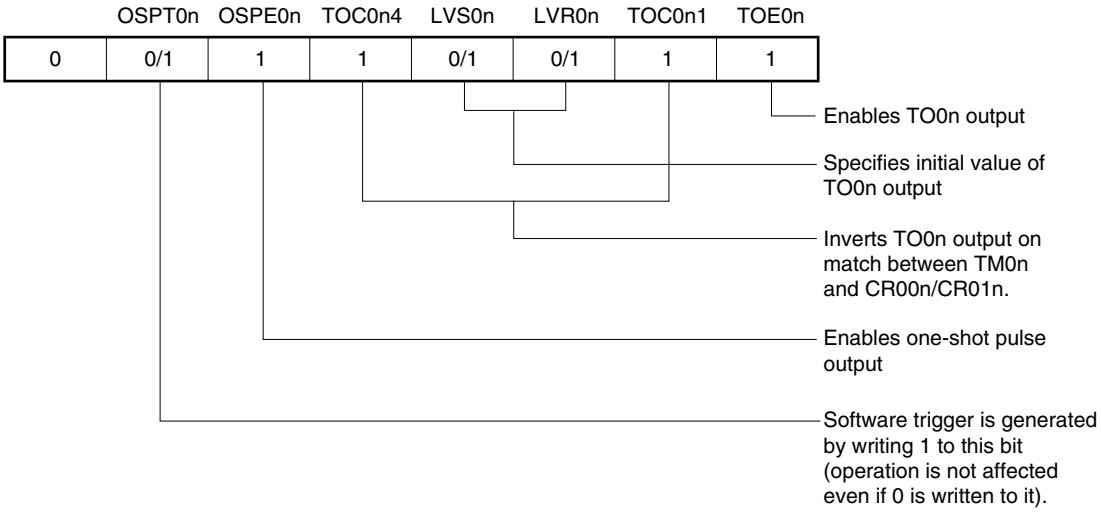
(a) 16-bit timer mode control register 0n (TMC0n)



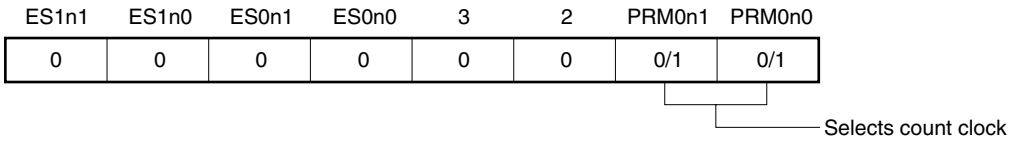
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)



(d) Prescaler mode register 0n (PRM0n)



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-55. Example of Register Settings for One-Shot Pulse Output Operation (2/2)

(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

This register is used as a compare register when a one-shot pulse is output. When the value of TM0n matches that of CR00n, an interrupt signal (INTTM00n) is generated and the TO0n output level is inverted.

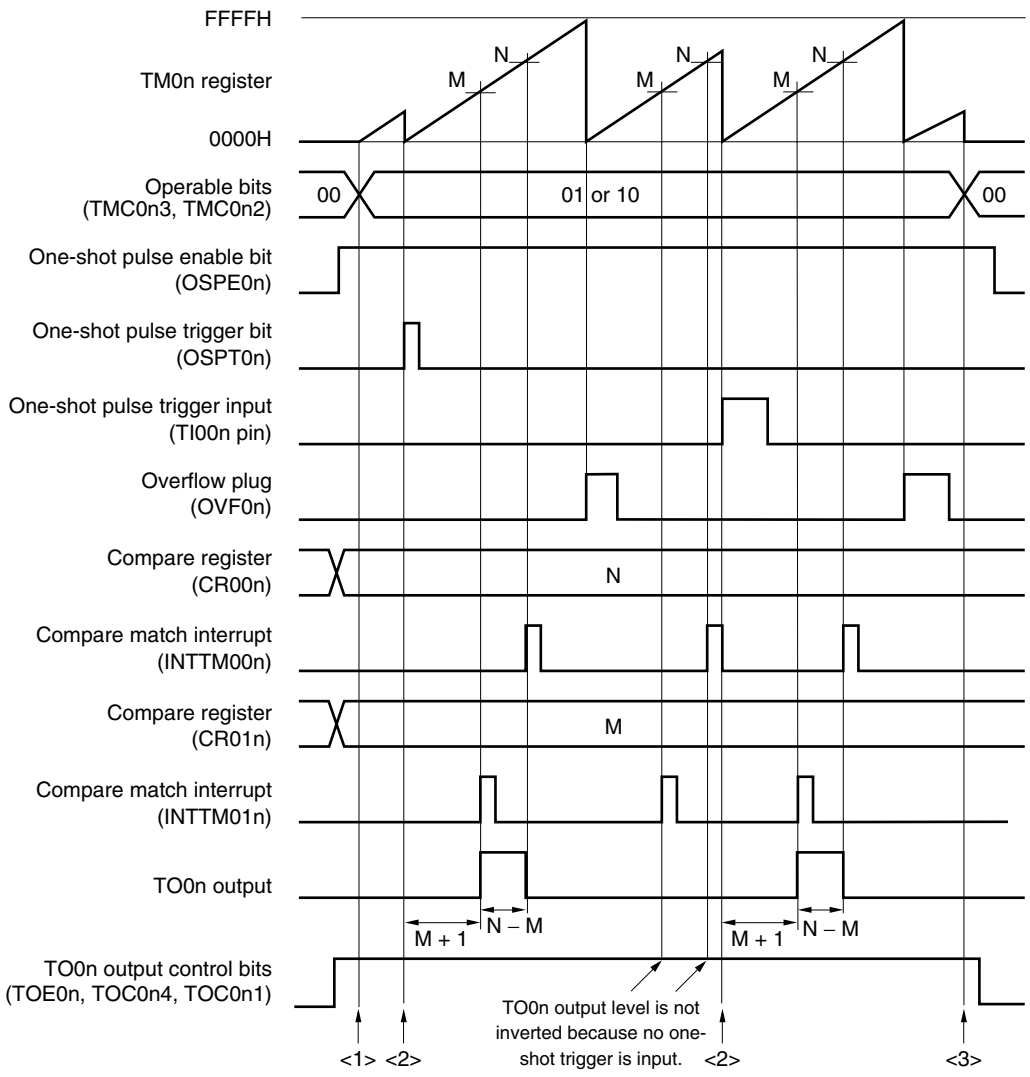
(g) 16-bit capture/compare register 01n (CR01n)

This register is used as a compare register when a one-shot pulse is output. When the value of TM0n matches that of CR01n, an interrupt signal (INTTM01n) is generated and the TO0n output level is inverted.

Caution Do not set the same value to CR00n and CR01n.

Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n) in the 78K0/KC2-C: n = 0).

Figure 6-56. Example of Software Processing for One-Shot Pulse Output Operation (1/2)

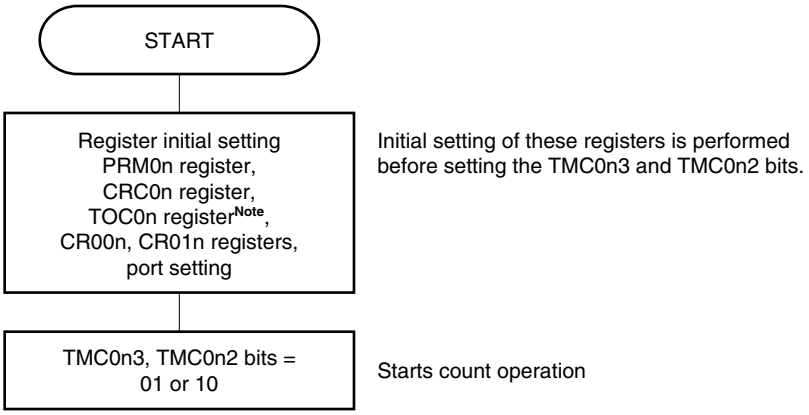


- Time from when the one-shot pulse trigger is input until the one-shot pulse is output
  = (M + 1) × Count clock cycle
- One-shot pulse output active level width
  = (N – M) × Count clock cycle

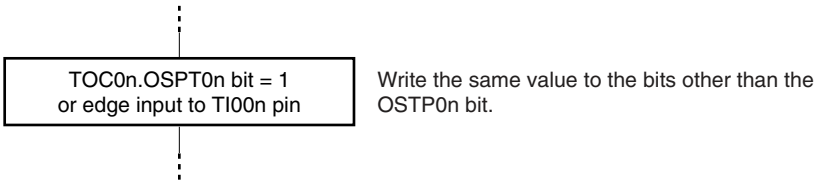
Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TO0n), and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-56. Example of Software Processing for One-Shot Pulse Output Operation (2/2)

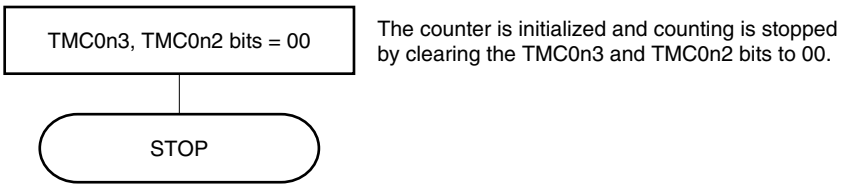
<1> Count operation start flow



<2> One-shot trigger input flow



<3> Count operation stop flow



Note Care must be exercised when setting TOC0n. For details, see **6.3 (3) 16-bit timer output control register 0n (TOC0n)**.

Remark n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

6.4.8 Pulse width measurement operation

TM0n can be used to measure the pulse width of the signal input to the TI00n and TI01n pins.

Measurement can be accomplished by operating the 16-bit timer/event counter 0n in the free-running timer mode or by restarting the timer in synchronization with the signal input to the TI00n pin.

When an interrupt is generated, read the value of the valid capture register and measure the pulse width. Check bit 0 (OVF0n) of 16-bit timer mode control register 0n (TMC0n). If it is set (to 1), clear it to 0 by software.

Figure 6-57. Block Diagram of Pulse Width Measurement (Free-Running Timer Mode)

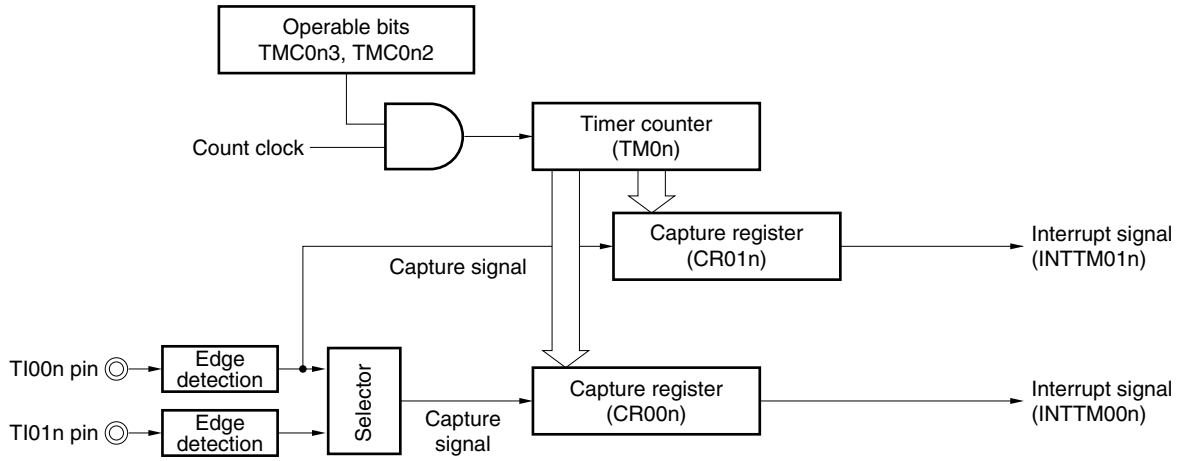
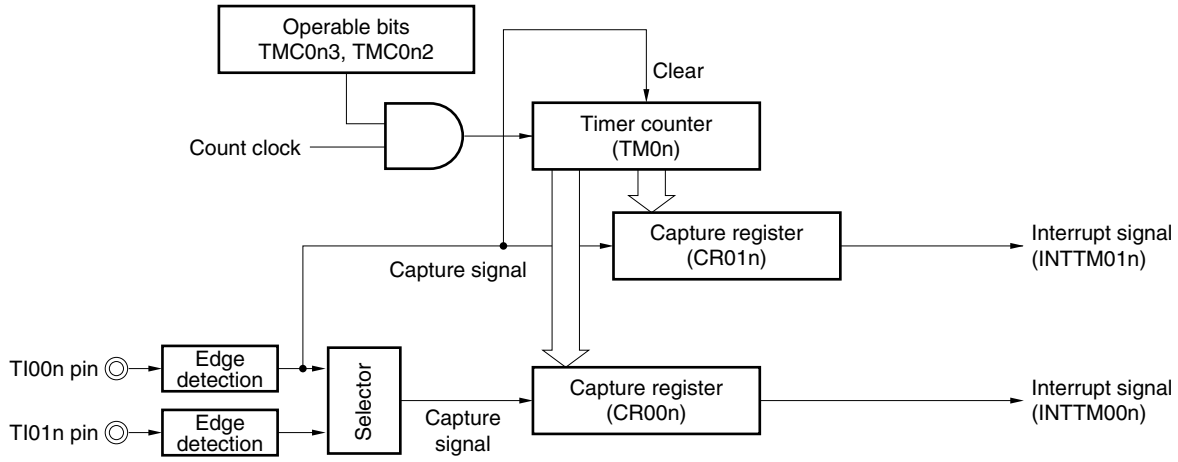


Figure 6-58. Block Diagram of Pulse Width Measurement
(Clear & Start Mode Entered by TI00n Pin Valid Edge Input)



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n) in the 78K0/KC2-C: n = 0).

A pulse width can be measured in the following three ways.

- Measuring the pulse width by using two input signals of the TI00n and TI01n pins (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI00n pin (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI00n pin (clear & start mode entered by the TI00n pin valid edge input)

Remarks 1. For the setting of the I/O pins, see **6.3 (5) Port mode registers 0, 5 (PM0, PM5)**.

2. For how to enable the INTTM00n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

3. n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n) in the 78K0/KC2-C: n = 0).

(1) Measuring the pulse width by using two input signals of the TI00n and TI01n pins (free-running timer mode)

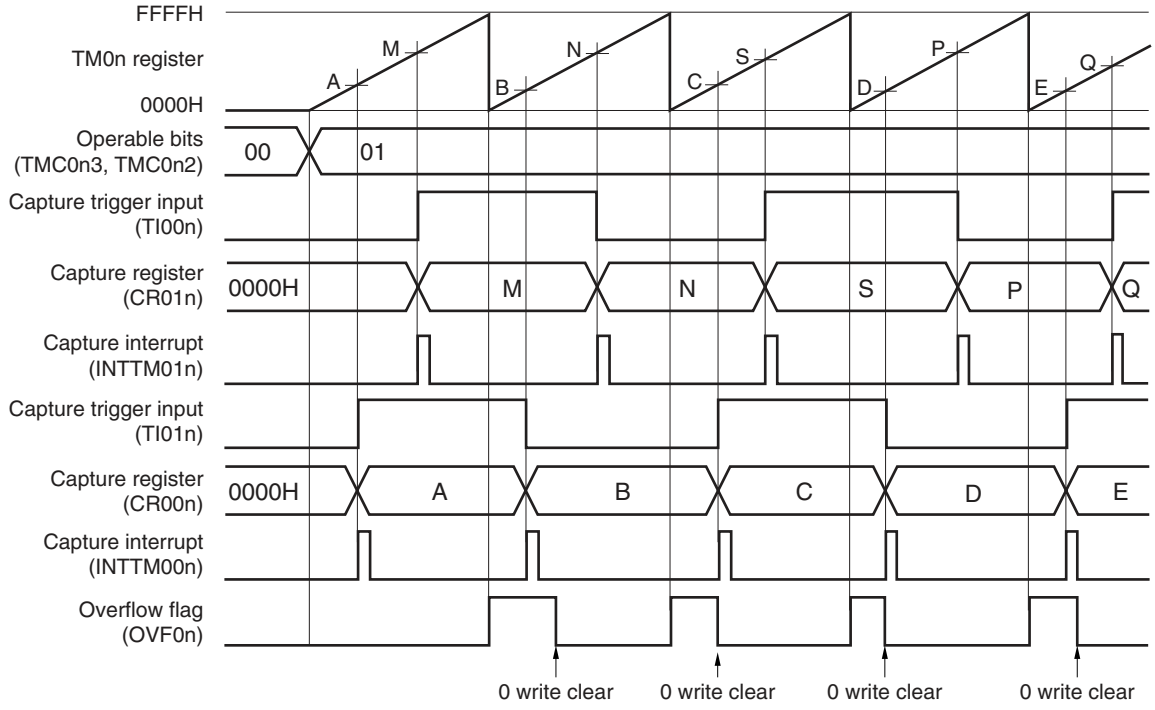
Set the free-running timer mode (TMC0n3 and TMC0n2 = 01). When the valid edge of the TI00n pin is detected, the count value of TM0n is captured to CR01n. When the valid edge of the TI01n pin is detected, the count value of TM0n is captured to CR00n. Specify detection of both the edges of the TI00n and TI01n pins.

By this measurement method, the previous count value is subtracted from the count value captured by the edge of each input signal. Therefore, save the previously captured value to a separate register in advance.

If an overflow occurs, the value becomes negative if the previously captured value is simply subtracted from the current captured value and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF0n) of 16-bit timer mode control register 0n (TMC0n) to 0.

Figure 6-59. Timing Example of Pulse Width Measurement (1)

• TMC0n = 04H, PRM0n = F0H, CRC0n = 05H



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n) in the 78K0/KC2-C: n = 0).

(2) Measuring the pulse width by using one input signal of the TI00n pin (free-running timer mode)

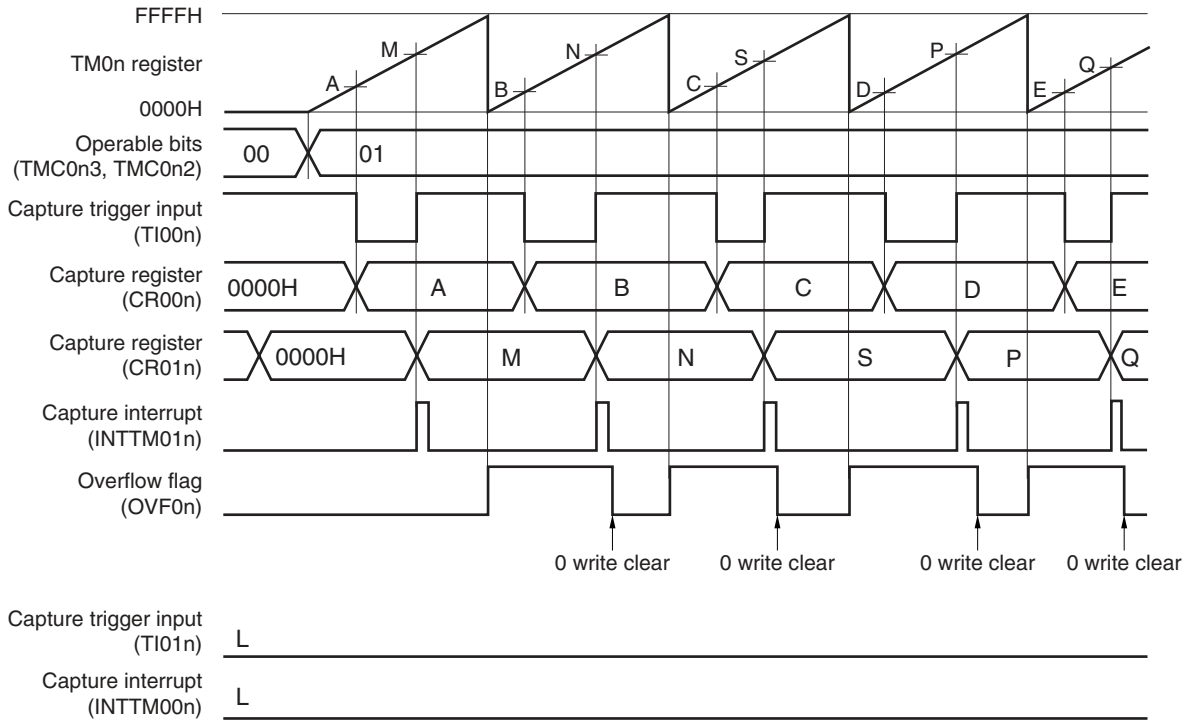
Set the free-running timer mode (TMC0n3 and TMC0n2 = 01). The count value of TM0n is captured to CR00n in the phase reverse to the valid edge detected on the TI00n pin. When the valid edge of the TI00n pin is detected, the count value of TM0n is captured to CR01n.

By this measurement method, values are stored in separate capture registers when a width from one edge to another is measured. Therefore, the capture values do not have to be saved. By subtracting the value of one capture register from that of another, a high-level width, low-level width, and cycle are calculated.

If an overflow occurs, the value becomes negative if one captured value is simply subtracted from another and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF0n) of 16-bit timer mode control register 0n (TMC0n) to 0.

Figure 6-60. Timing Example of Pulse Width Measurement (2)

• TMC0n = 04H, PRM0n = 10H, CRC0n = 07H



Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

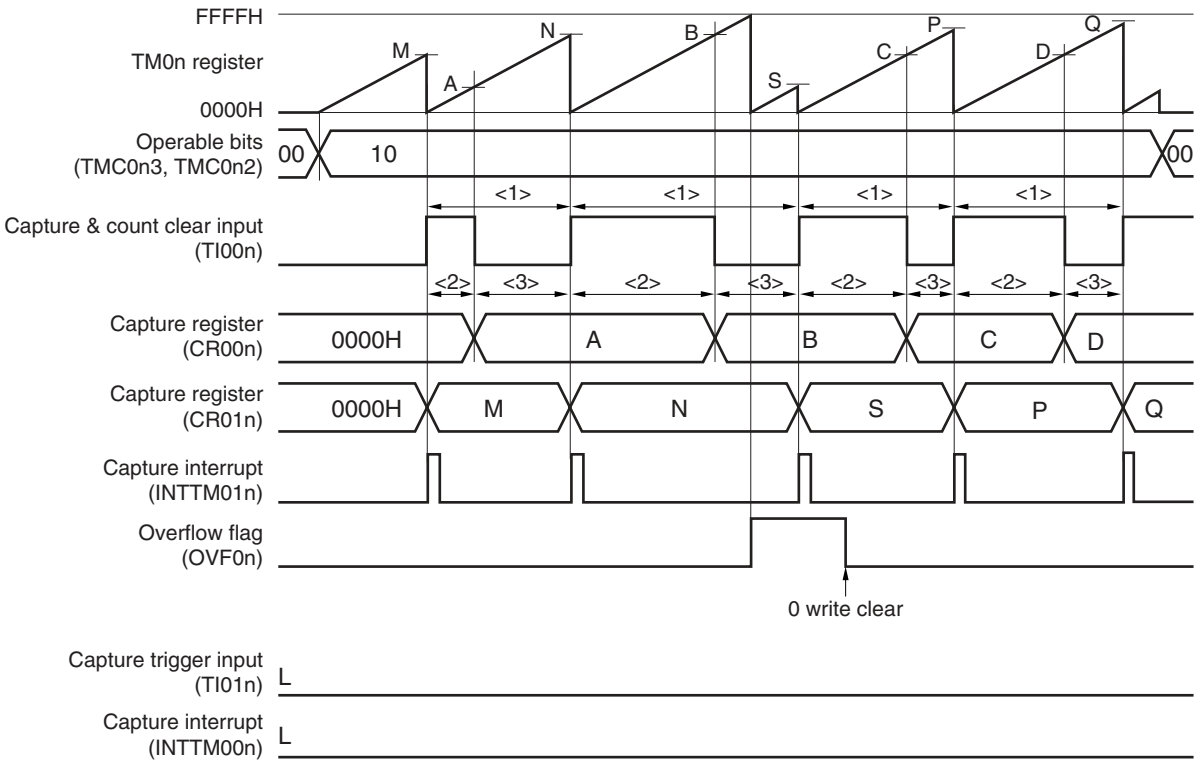
(3) Measuring the pulse width by using one input signal of the TI00n pin (clear & start mode entered by the TI00n pin valid edge input)

Set the clear & start mode entered by the TI00n pin valid edge (TMC0n3 and TMC0n2 = 10). The count value of TM0n is captured to CR00n in the phase reverse to the valid edge of the TI00n pin, and the count value of TM0n is captured to CR01n and TM0n is cleared (0000H) when the valid edge of the TI00n pin is detected. Therefore, a cycle is stored in CR01n if TM0n does not overflow.

If an overflow occurs, take the value that results from adding 10000H to the value stored in CR01n as a cycle. Clear bit 0 (OVF0n) of 16-bit timer mode control register 0n (TMC0n) to 0.

Figure 6-61. Timing Example of Pulse Width Measurement (3)

• TMC0n = 08H, PRM0n = 10H, CRC0n = 07H



<1> Pulse cycle = (10000H × Number of times OVF0n bit is set to 1 + Captured value of CR01n) × Count clock cycle

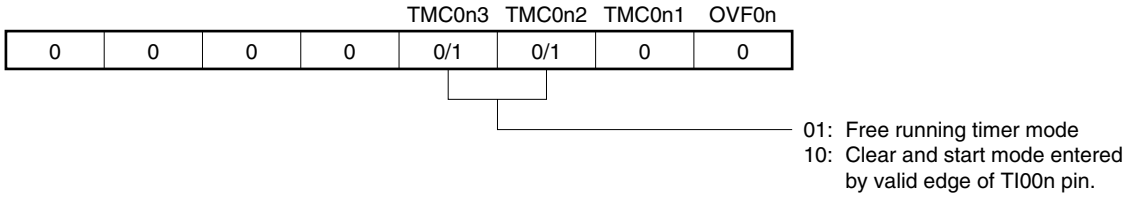
<2> High-level pulse width = (10000H × Number of times OVF0n bit is set to 1 + Captured value of CR00n) × Count clock cycle

<3> Low-level pulse width = (Pulse cycle − High-level pulse width)

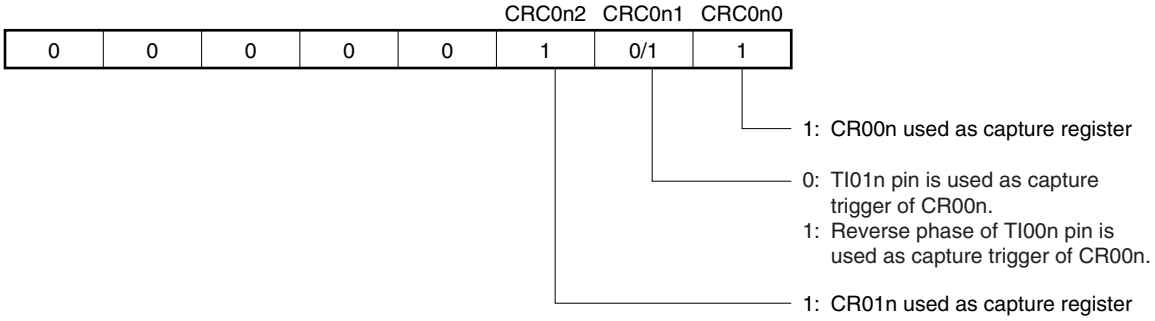
Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n), and CRC0n register in the 78K0/KC2-C: n = 0).

Figure 6-62. Example of Register Settings for Pulse Width Measurement (1/2)

(a) 16-bit timer mode control register 0n (TMC0n)



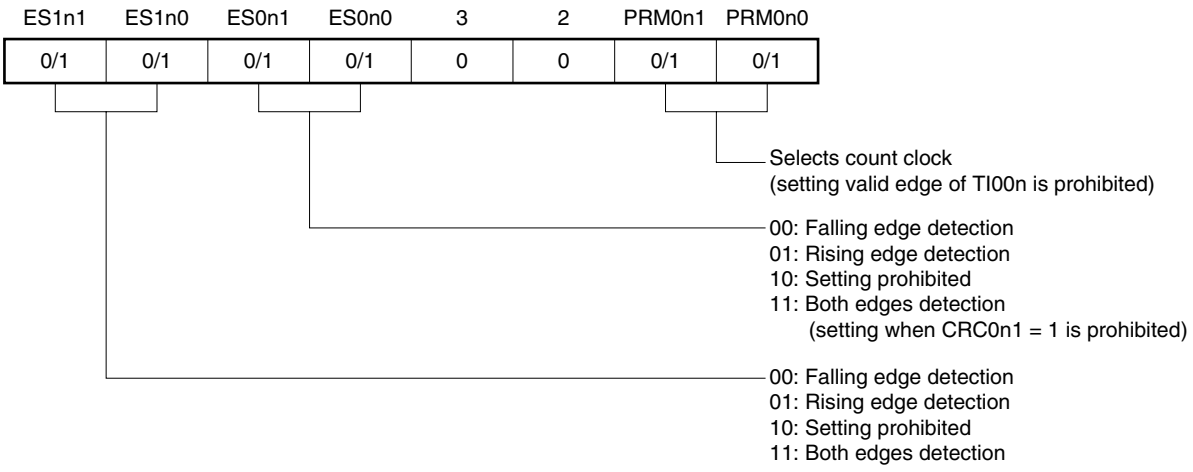
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)

	OSPT0n	OSPE0n	TOC0n4	LVS0n	LVR0n	TOC0n1	TOE0n
0	0	0	0	0	0	0	0

(d) Prescaler mode register 0n (PRM0n)



Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n), CRC0n, and TOC0n registers in the 78K0/KC2-C: n = 0).

Figure 6-62. Example of Register Settings for Pulse Width Measurement (2/2)

(e) 16-bit timer counter 0n (TM0n)

By reading TM0n, the count value can be read.

(f) 16-bit capture/compare register 00n (CR00n)

This register is used as a capture register. Either the TI00n or TI01n pin is selected as a capture trigger. When a specified edge of the capture trigger is detected, the count value of TM0n is stored in CR00n.

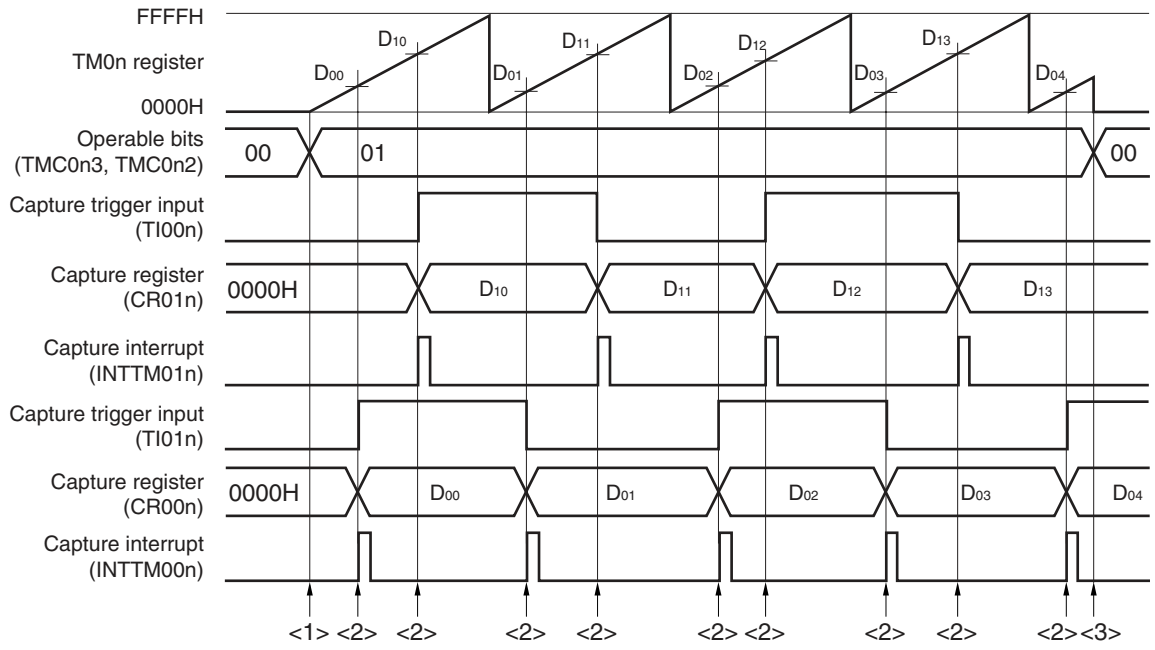
(g) 16-bit capture/compare register 01n (CR01n)

This register is used as a capture register. The signal input to the TI00n pin is used as a capture trigger. When the capture trigger is detected, the count value of TM0n is stored in CR01n.

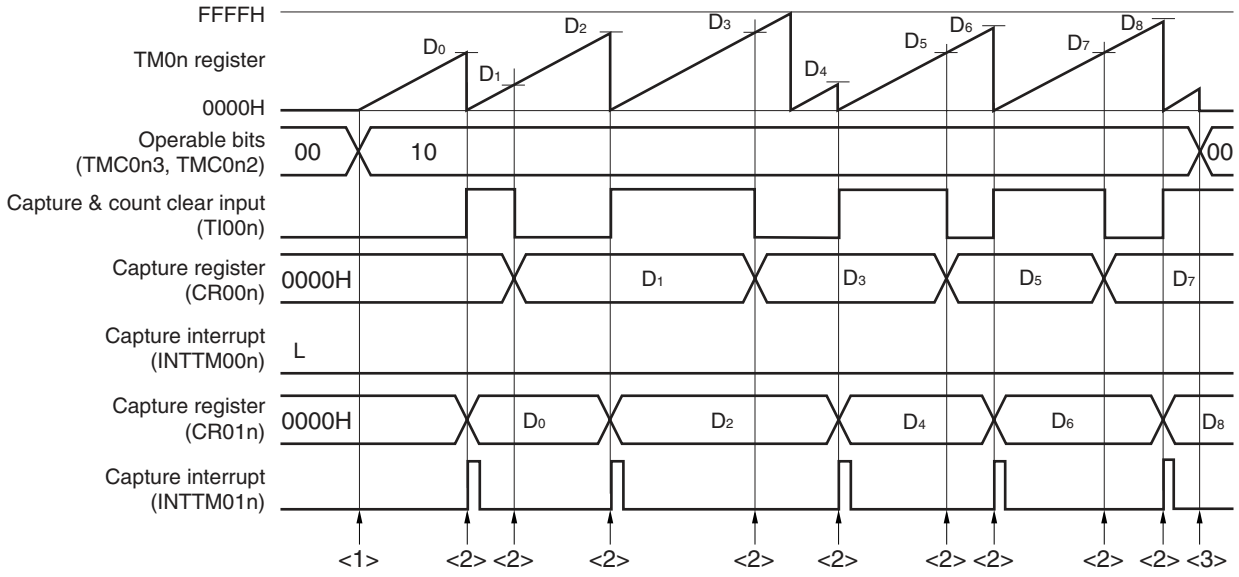
Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n) in the 78K0/KC2-C: n = 0).

Figure 6-63. Example of Software Processing for Pulse Width Measurement (1/2)

(a) Example of free-running timer mode



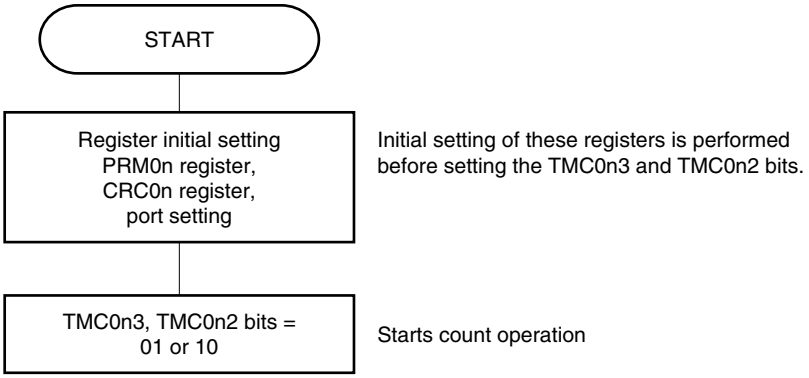
(b) Example of clear & start mode entered by TI00n pin valid edge



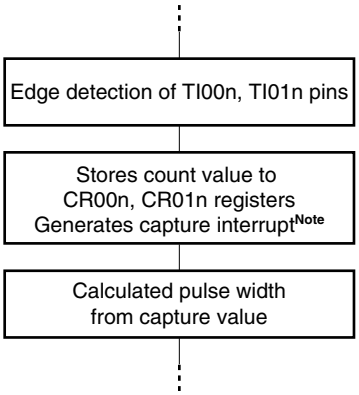
Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n) in the 78K0/KC2-C: n = 0).

Figure 6-63. Example of Software Processing for Pulse Width Measurement (2/2)

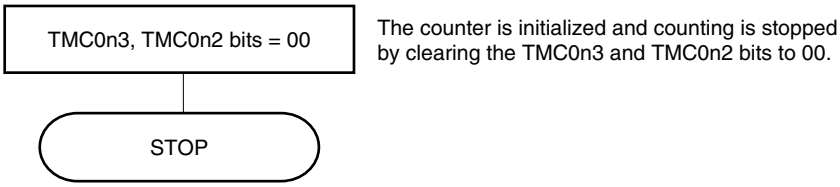
<1> Count operation start flow



<2> Capture trigger input flow



<3> Count operation stop flow



Note The capture interrupt signal (INTTM00n) is not generated when the reverse-phase edge of the TI00n pin input is selected to the valid edge of CR00n.

Remark n = 0 to 2 (This is, however, in the case of the timer input pins (TI00n, TI01n), and CRC0n register in the 78K0/KC2-C: n = 0).

6.5 Special Use of TM0n

6.5.1 Rewriting CR01n during TM0n operation

In principle, rewriting CR00n and CR01n of the 78K0/Kx2-C microcontrollers when they are used as compare registers is prohibited while TM0n is operating (TMC0n3 and TMC0n2 = other than 00).

However, the value of CR01n can be changed, even while TM0n is operating, using the following procedure if CR01n is used for PPG output and the duty factor is changed. (When changing the value of CR01n to a smaller value than the current one, rewrite it immediately after its value matches the value of TM0n. When changing the value of CR01n to a larger value than the current one, rewrite it immediately after the values of CR00n and TM0n match. If the value of CR01n is rewritten immediately before a match between CR01n and TM0n, or between CR00n and TM0n, an unexpected operation may be performed.).

Procedure for changing value of CR01n

<1> Disable interrupt INTTM01n (TMMK01n = 1).
<2> Disable reversal of the timer output when the value of TM0n matches that of CR01n (TOC0n4 = 0).
<3> Change the value of CR01n.
<4> Wait for one cycle of the count clock of TM0n.
<5> Enable reversal of the timer output when the value of TM0n matches that of CR01n (TOC0n4 = 1).
<6> Clear the interrupt flag of INTTM01n (TMIF01n = 0) to 0.
<7> Enable interrupt INTTM01n (TMMK01n = 0).

Remark For TMIF01n and TMMK01n, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

6.5.2 Setting LVS0n and LVR0n

(1) Usage of LVS0n and LVR0n

LVS0n and LVR0n are used to set the default value of the TO0n output and to invert the timer output without enabling the timer operation (TMC0n3 and TMC0n2 = 00). Clear LVS0n and LVR0n to 00 (default value: low-level output) when software control is unnecessary.

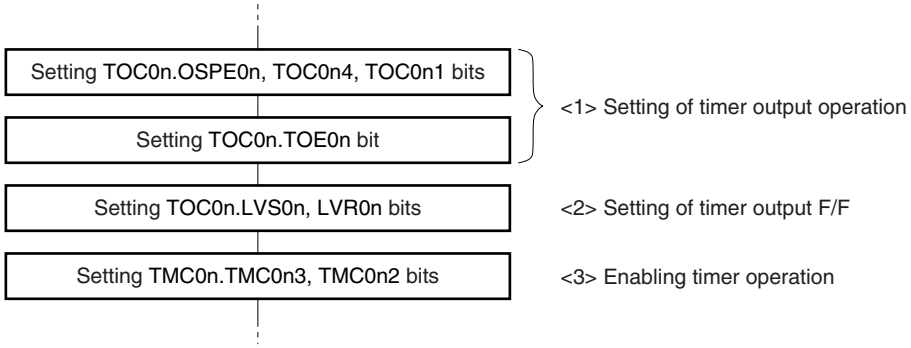
LVS0n	LVR0n	Timer Output Status
0	0	Not changed (low-level output)
0	1	Cleared (low-level output)
1	0	Set (high-level output)
1	1	Setting prohibited

Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n) and TOC0n register in the 78K0/KC2-C: n = 0).

(2) Setting LVS0n and LVR0n

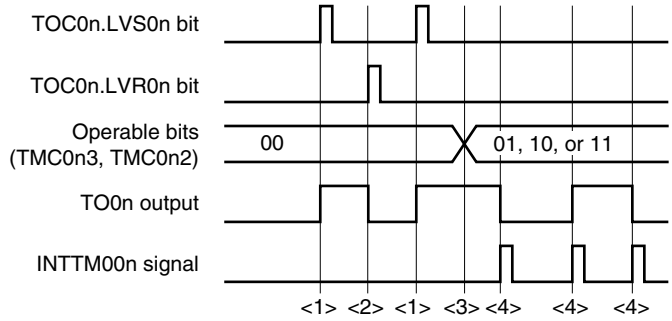
Set LVS0n and LVR0n using the following procedure.

Figure 6-64. Example of Flow for Setting LVS0n and LVR0n Bits



Caution Be sure to set LVS0n and LVR0n following steps <1>, <2>, and <3> above.
Step <2> can be performed after <1> and before <3>.

Figure 6-65. Timing Example of LVR0n and LVS0n



<1> The TO0n output goes high when LVS0n and LVR0n = 10.

<2> The TO0n output goes low when LVS0n and LVR0n = 01 (the pin output remains unchanged from the high level even if LVS0n and LVR0n are cleared to 00).

<3> The timer starts operating when TMC0n3 and TMC0n2 are set to 01, 10, or 11. Because LVS0n and LVR0n were set to 10 before the operation was started, the TO0n output starts from the high level. After the timer starts operating, setting LVS0n and LVR0n is prohibited until TMC0n3 and TMC0n2 = 00 (disabling the timer operation).

<4> The TO0n output level is inverted each time an interrupt signal (INTTM00n) is generated.

Remark n = 0 to 2 (This is, however, in the case of the timer output pin (TO0n) and TOC0n register in the 78K0/KC2-C: n = 0).

6.6 Cautions for 16-Bit Timer/Event Counters 00, 01, and 02

(1) Restrictions for each channel of 16-bit timer/event counter 0n

Table 6-3 shows the restrictions for each channel.

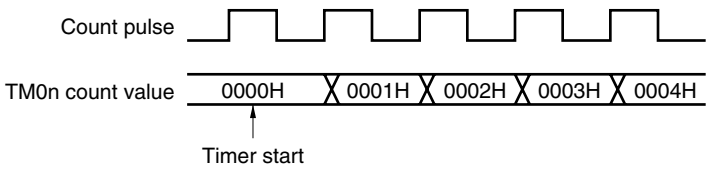
Table 6-3. Restrictions for Each Channel of 16-Bit Timer/Event Counter 0n

Operation	Restriction
As interval timer	–
As square-wave output	
As external event counter	
As clear & start mode entered by TI00n pin valid edge input	Using timer output (TO0n) is prohibited when detection of the valid edge of the TI01n pin is used. (TOC0n = 00H)
As free-running timer	–
As PPG output	0000H ≤ CP01n < CR00n ≤ FFFFH
As one-shot pulse output	Setting the same value to CR00n and CP01n is prohibited.
As pulse width measurement	Using timer output (TO0n) is prohibited (TOC0n = 00H)

(2) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because counting TM0n is started asynchronously to the count pulse.

Figure 6-66. Start Timing of TM0n Count



(3) Setting of CR00n and CR01n (clear & start mode entered upon a match between TM0n and CR00n)

Set a value other than 0000H to CR00n and CR01n (TM0n cannot count one pulse when it is used as an external event counter).

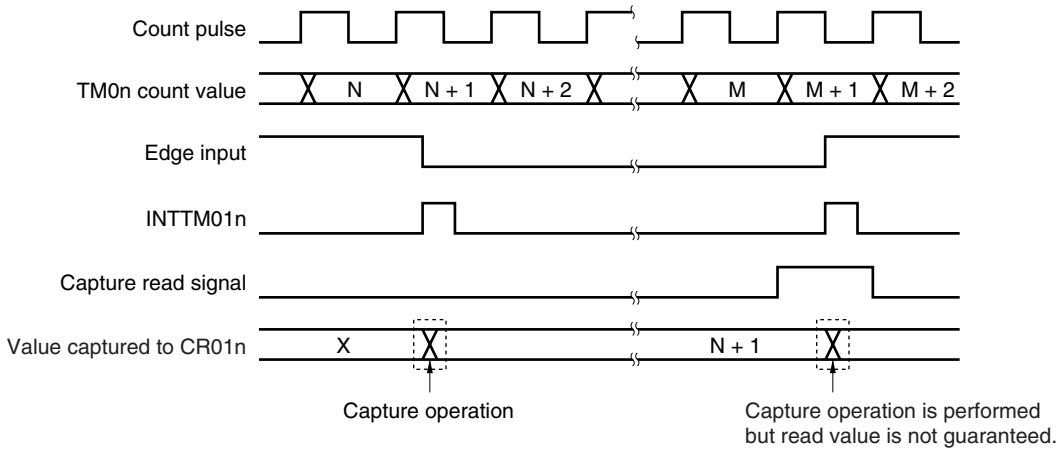
Remark n = 0 to 2 (This is, however, in the case of the timer I/O pins (TI00n, TI01n, TO0n), and TOC0n register in the 78K0/KC2-C: n = 0).

(4) Timing of holding data by capture register

(a) When the valid edge is input to the TI00n/TI01n pin and the reverse phase of the TI00n pin is detected while CR00n/CR01n is read, CR01n performs a capture operation but the read value of CR00n/CR01n is not guaranteed. At this time, an interrupt signal (INTTM00n/INTTM01n) is generated when the valid edge of the TI00n/TI01n pin is detected (the interrupt signal is not generated when the reverse-phase edge of the TI00n pin is detected).

When the count value is captured because the valid edge of the TI00n/TI01n pin was detected, read the value of CR00n/CR01n after INTTM00n/INTTM01n is generated.

Figure 6-67. Timing of Holding Data by Capture Register



(b) The values of CR00n and CR01n are not guaranteed after 16-bit timer/event counter 0n stops.

(5) Setting valid edge

Set the valid edge of the TI00n pin while the timer operation is stopped (TMC0n3 and TMC0n2 = 00). Set the valid edge by using ES0n0 and ES0n1.

(6) Re-triggering one-shot pulse

Make sure that the trigger is not generated while an active level is being output in the one-shot pulse output mode. Be sure to input the next trigger after the current active level is output.

Remark n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n, TI01n) in the 78K0/KC2-C: n = 0).

(7) Operation of OVF0n flag

(a) Setting OVF0n flag (1)

The OVF0n flag is set to 1 in the following case, as well as when TM0n overflows.

Select the clear & start mode entered upon a match between TM0n and CR00n.

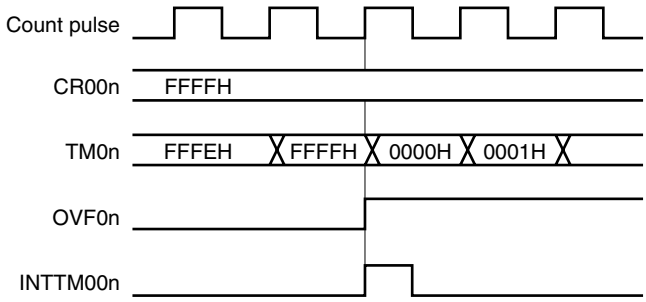
↓

Set CR00n to FFFFH.

↓

When TM0n matches CR00n and TM0n is cleared from FFFFH to 0000H

Figure 6-68. Operation Timing of OVF0n Flag



(b) Clearing OVF0n flag

Even if the OVF0n flag is cleared to 0 after TM0n overflows and before the next count clock is counted (before the value of TM0n becomes 0001H), it is set to 1 again and clearing is invalid.

(8) One-shot pulse output

One-shot pulse output operates correctly in the free-running timer mode or the clear & start mode entered by the TI00n pin valid edge. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM0n and CR00n.

Remark n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n, TI01n) in the 78K0/KC2-C: n = 0).

(9) Capture operation

(a) When valid edge of TI00n is specified as count clock

When the valid edge of TI00n is specified as the count clock, the capture register for which TI00n is specified as a trigger does not operate correctly.

(b) Pulse width to accurately capture value by signals input to TI01n and TI00n pins

To accurately capture the count value, the pulse input to the TI00n and TI01n pins as a capture trigger must be wider than two count clocks selected by PRM0n (see **Figure 6-11**).

(c) Generation of interrupt signal

The capture operation is performed at the falling edge of the count clock but the interrupt signals (INTTM00n and INTTM01n) are generated at the rising edge of the next count clock (see **Figure 6-11**).

(d) Note when CRC0n1 (bit 1 of capture/compare control register 0n (CRC0n)) is set to 1

When the count value of the TM0n register is captured to the CR00n register in the phase reverse to the signal input to the TI00n pin, the interrupt signal (INTTM00n) is not generated after the count value is captured. If the valid edge is detected on the TI01n pin during this operation, the capture operation is not performed but the INTTM00n signal is generated as an external interrupt signal. Mask the INTTM00n signal when the external interrupt is not used.

(10) Edge detection

(a) Specifying valid edge after reset

If the operation of the 16-bit timer/event counter 0n is enabled after reset and while the TI00n or TI01n pin is at high level and when the rising edge or both the edges are specified as the valid edge of the TI00n or TI01n pin, then the high level of the TI00n or TI01n pin is detected as the rising edge. Note this when the TI00n or TI01n pin is pulled up. However, the rising edge is not detected when the operation is once stopped and then enabled again.

(b) Sampling clock for eliminating noise

The sampling clock for eliminating noise differs depending on whether the valid edge of TI00n is used as the count clock or capture trigger. In the former case, the sampling clock is fixed to f_{PRS}. In the latter, the count clock selected by PRM0n is used for sampling.

When the signal input to the TI00n pin is sampled and the valid level is detected two times in a row, the valid edge is detected. Therefore, noise having a short pulse width can be eliminated (see **Figure 6-11**).

(11) Timer operation

The signal input to the TI00n/TI01n pin is not acknowledged while the timer is stopped, regardless of the operation mode of the CPU.

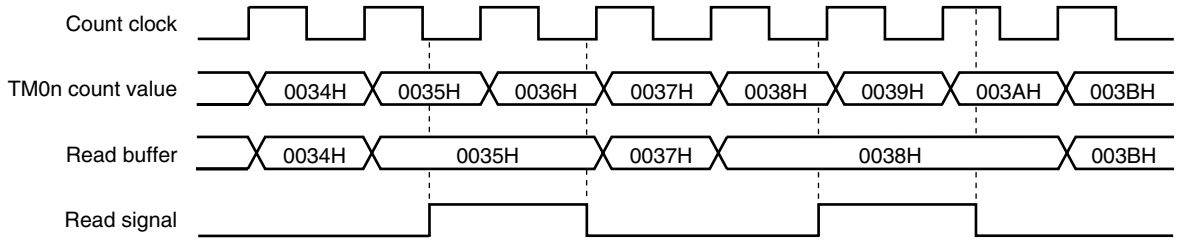
Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. n = 0 to 2 (This is, however, in the case of the timer input pin (TI00n, TI01n) and CRC0n register in the 78K0/KC2-C: n = 0).

(12) Reading of 16-bit timer counter 0n (TM0n)

TM0n can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 6-69. 16-bit Timer Counter 0n (TM0n) Read Timing



Remark n = 0 to 2

CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50 AND 51

7.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output
- Carrier generator (8-bit timer 51 only) Note

Note TM51 and TMH1 can be used in combination as a carrier generator mode.

7.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 include the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer input	TI5n
Timer output	TO5n
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 1 (PM1) or port mode register 3 (PM3) Port register 1 (P1) or port register 3 (P3)

Figures 7-1 and 7-2 show the block diagrams of 8-bit timer/event counters 50 and 51.

Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 50

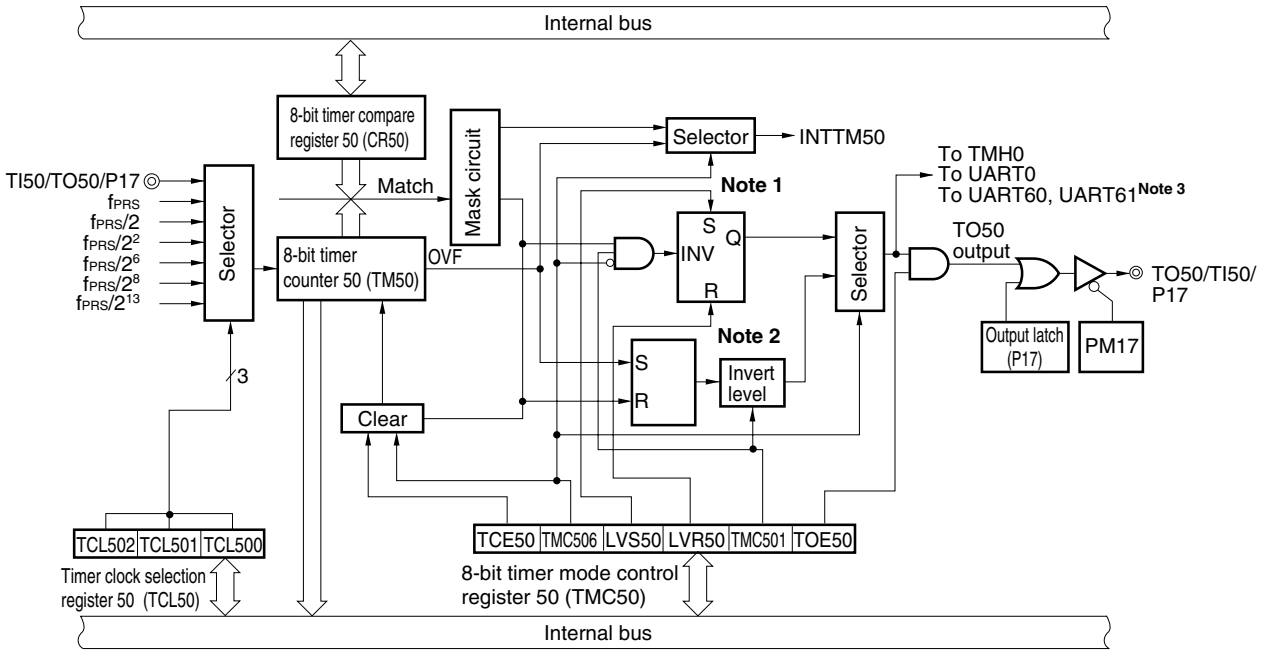
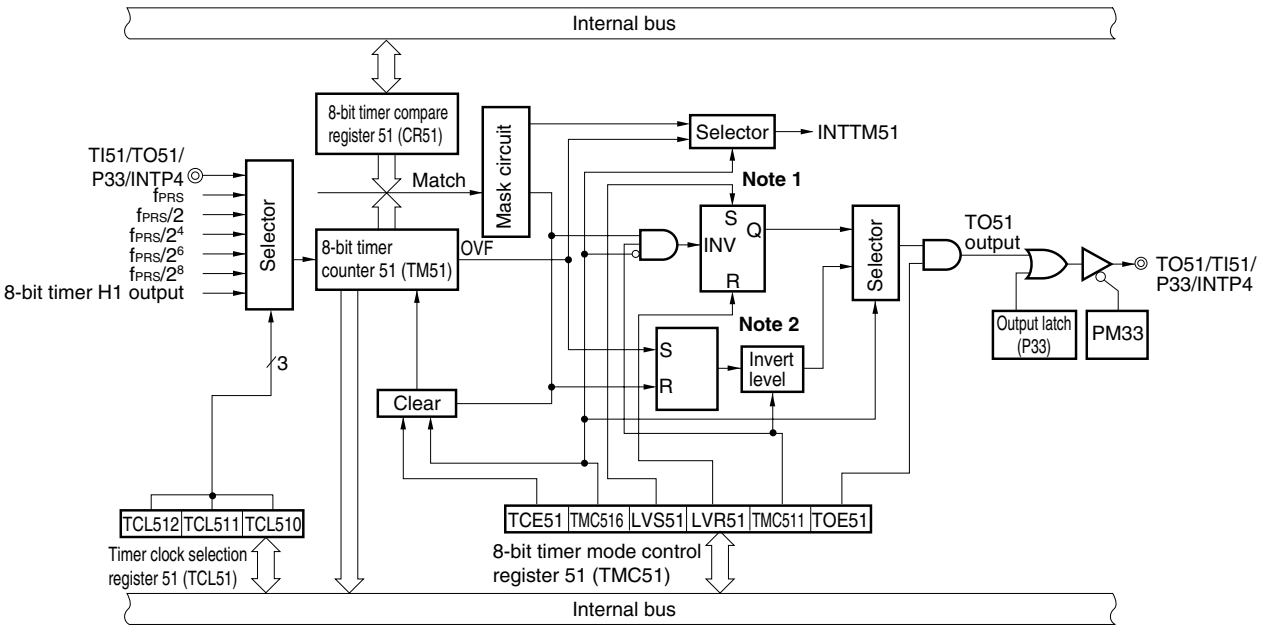


Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51



Notes 1. Timer output F/F

2. PWM output F/F

3. 78K0/KE2-C only

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit register that counts the count pulses and is read-only.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 7-3. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF16H (TM50), FF1FH (TM51) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0, 1)								

In the following situations, the count value is cleared to 00H.

<1> Reset signal generation

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

(2) 8-bit timer compare register 5n (CR5n)

CR5n can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match.

In the PWM mode, TO5n output becomes inactive when the values of TM5n and CR5n match, but no interrupt is generated.

The value of CR5n can be set within 00H to FFH.

Reset signal generation clears CR5n to 00H.

Figure 7-4. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF17H (CR50), FF41H (CR51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0, 1)								

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51

The following four registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 1 (PM1) or port mode register 3 (PM3)
- Port register 1 (P1) or port register 3 (P3)

(1) Timer clock selection register 5n (TCL5n)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of the TI5n pin input.

TCL5n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TCL5n to 00H.

Remark n = 0, 1

Figure 7-5. Format of Timer Clock Selection Register 50 (TCL50)

Address: FF6AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	TI50 pin falling edge[Note]				
0	0	1	TI50 pin rising edge[Note]				
0	1	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	1	1	$f_{PRS}/2^{13}$	0.24 kHz	0.61 kHz	1.22 kHz	2.44 kHz

Note Do not start timer operation with the external clock from the TI50 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

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

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 7-6. Format of Timer Clock Selection Register 51 (TCL51)

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	TI51 pin falling edge[Note]				
0	0	1	TI51 pin rising edge[Note]				
0	1	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	1	1	Output signal of the timer H1				

Note Do not start timer operation with the external clock from the TI51 pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

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

Remark f_{PRS}: Peripheral hardware clock frequency

(2) 8-bit timer mode control register 5n (TMC5n)

TMC5n is a register that performs the following five types of settings.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Timer output F/F (flip flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode.
<5> Timer output control

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 7-7. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF6BH After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Mode in which clear & start occurs on a match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F clear (0) (default value of TO50 output: low level)
1	0	Timer output F/F set (1) (default value of TO50 output: high level)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE50	Timer output control
0	Output disabled (TO50 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

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

Figure 7-8. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FF43H After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F clear (0) (default value of TO51 output: low)
1	0	Timer output F/F set (1) (default value of TO51 output: high)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TO51 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

Cautions 1. The settings of LVS5n and LVR5n are valid in other than PWM mode.

2. Perform <1> to <4> below in the following order, not at the same time.

<1> Set TMC5n1, TMC5n6: Operation mode setting

<2> Set TOE5n to enable output: Timer output enable

<3> Set LVS5n, LVR5n (see Caution 1): Timer F/F setting

<4> Set TCE5n

3. When TCE5n = 1, setting the other bits of TMC5n is prohibited.

4. The actual TO50/TI50/P17 and TO51/TI51/P33/INTP4 pin outputs are determined depending on PM17 and P17, and PM33 and P33, besides TO5n output.

Remarks 1. In PWM mode, PWM output is made inactive by clearing TCE5n to 0.

2. If LVS5n and LVR5n are read, the value is 0.

3. The values of the TMC5n6, LVS5n, LVR5n, TMC5n1, and TOE5n bits are reflected at the TO5n output regardless of the value of TCE5n.

4. n = 0, 1

(3) Port mode registers 1 and 3 (PM1, PM3)

These registers set port 1 and 3 input/output in 1-bit units.

When using the P17/TO50/TI50 and P33/TO51/TI51/INTP4 pins for timer output, clear PM17 and PM33 and the output latches of P17 and P33 to 0.

When using the P17/TO50/TI50 and P33/TO51/TI51/INTP4 pins for timer input, set PM17 and PM33 to 1. The output latches of P17 and P33 at this time may be 0 or 1.

PM1 and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 7-9. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 7-10. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operations of 8-Bit Timer/Event Counters 50 and 51

7.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

Setting

<1> Set the registers.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don't care)

<2> After TCE5n = 1 is set, the count operation starts.

<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

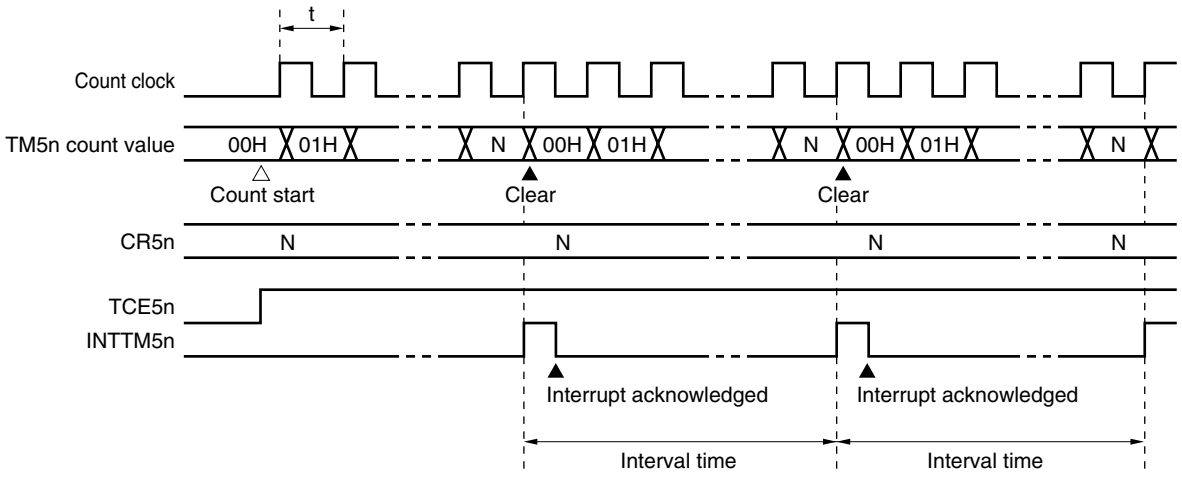
<4> INTTM5n is generated repeatedly at the same interval.
Set TCE5n to 0 to stop the count operation.

Caution Do not write other values to CR5n during operation.

Remark For how to enable the INTTM5n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

Figure 7-11. Interval Timer Operation Timing (1/2)

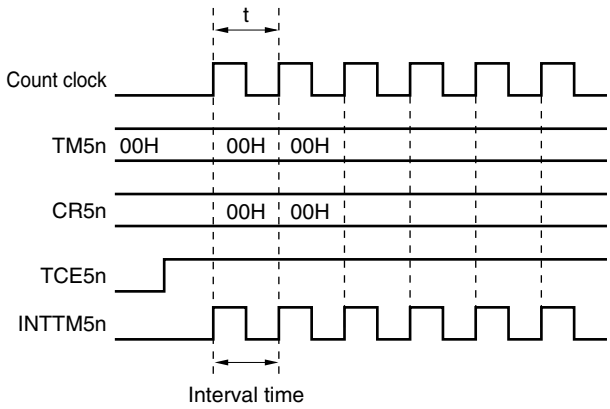
(a) Basic operation



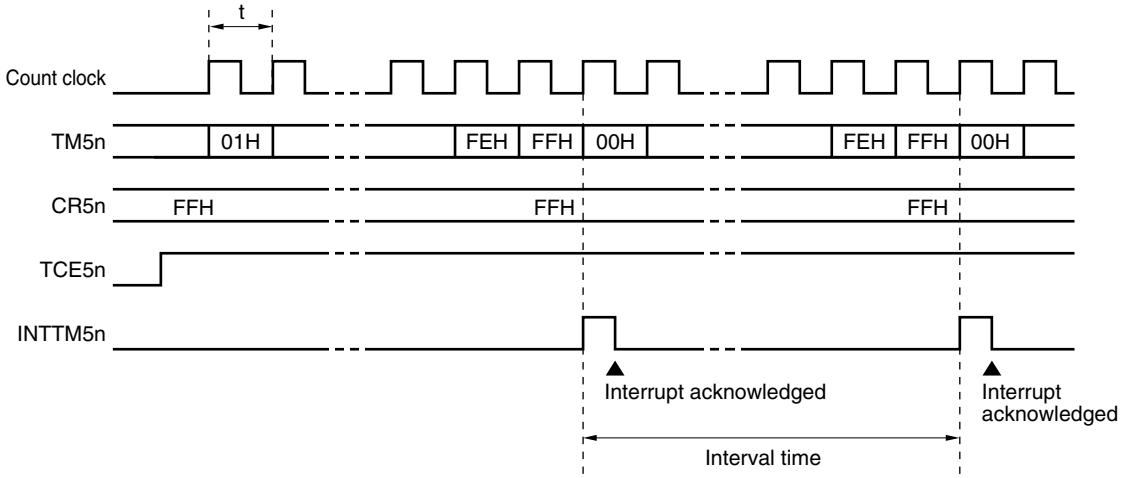
Remark Interval time = (N + 1) × t
N = 01H to FFH
n = 0, 1

Figure 7-11. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0, 1

7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TI5n pin by 8-bit timer counter 5n (TM5n).

TM5n is incremented each time the valid edge specified by timer clock selection register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

Whenever the TM5n value matches the value of CR5n, INTTM5n is generated.

Setting

<1> Set each register.

- Set the port mode register (PM17 or PM33)[Note] to 1.
- TCL5n: Select TI5n pin input edge.
  TI5n pin falling edge → TCL5n = 00H
  TI5n pin rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 00000000B)

<2> When TCE5n = 1 is set, the number of pulses input from the TI5n pin is counted.

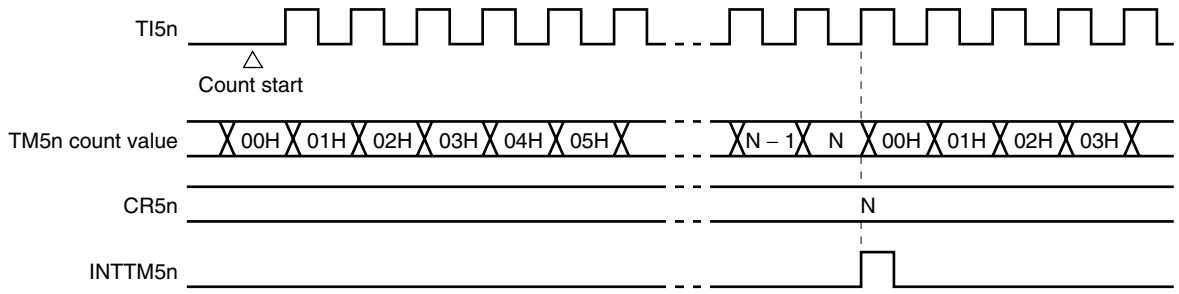
<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

Note 8-bit timer/event counter 50: PM17
8-bit timer/event counter 51: PM33

Remark For how to enable the INTTM5n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

Figure 7-12. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0, 1

7.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is inverted at intervals determined by the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- Clear the port output latch (P17 or P33)[Note] and port mode register (PM17 or PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.

LVS5n	LVR5n	Timer Output F/F Status Setting
0	1	Timer output F/F clear (0) (default value of TO5n output: low level)
1	0	Timer output F/F set (1) (default value of TO5n output: high level)

Timer output enabled
(TMC5n = 00001011B or 00000111B)

<2> After TCE5n = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n. The frequency is as follows.

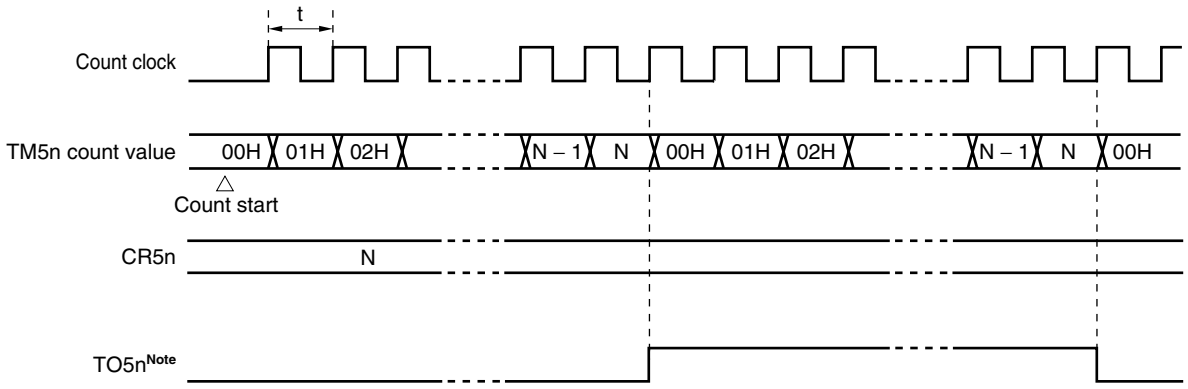
- Frequency = 1/2t (N + 1)
  (N: 00H to FFH)

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

Caution Do not write other values to CR5n during operation.

Remarks 1. For how to enable the INTTM5n signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.
2. n = 0, 1

Figure 7-13. Square-Wave Output Operation Timing



Note The initial value of TO5n output can be set by bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

7.4.4 PWM output operation

8-bit timer/event counter 5n operates as a PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

PWM output can be enabled/disabled with bit 0 (TOE5n) of TMC5n.

Caution In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

(1) PWM output basic operation

Setting

<1> Set each register.

- Clear the port output latch (P17 or P33)[Note] and port mode register (PM17 or PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Clear TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

PWM output operation

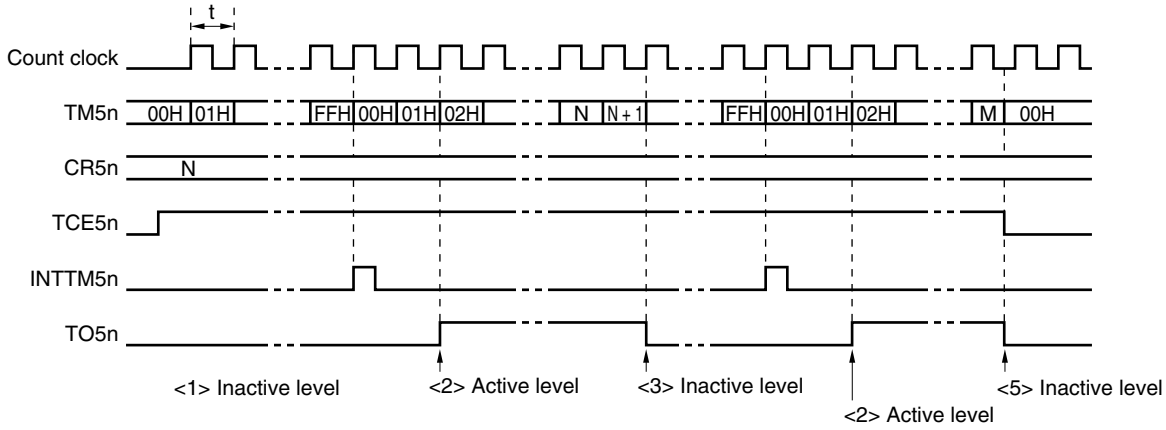
<1> PWM output (TO5n output) outputs an inactive level until an overflow occurs.
<2> When an overflow occurs, the active level is output. The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).
<3> After the CR5n matches the count value, the inactive level is output until an overflow occurs again.
<4> Operations <2> and <3> are repeated until the count operation stops.
<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.
For details of timing, see **Figures 7-14** and **7-15**.
The cycle, active-level width, and duty are as follows.

- Cycle = 2^8t
- Active-level width = Nt
- Duty = N/2^8
  (N = 00H to FFH)

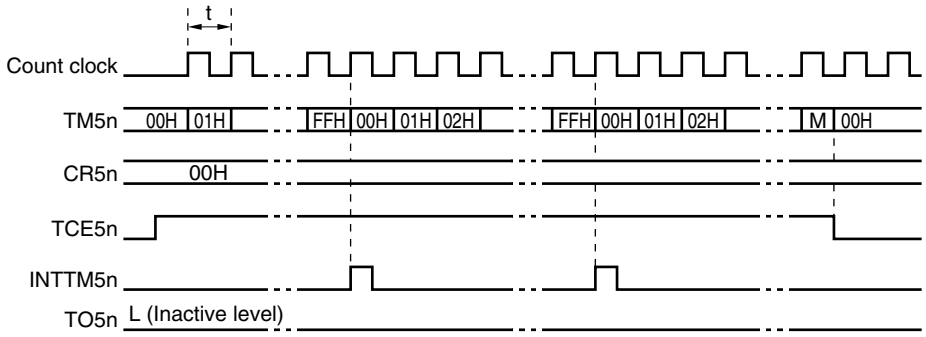
Remark n = 0, 1

Figure 7-14. PWM Output Operation Timing

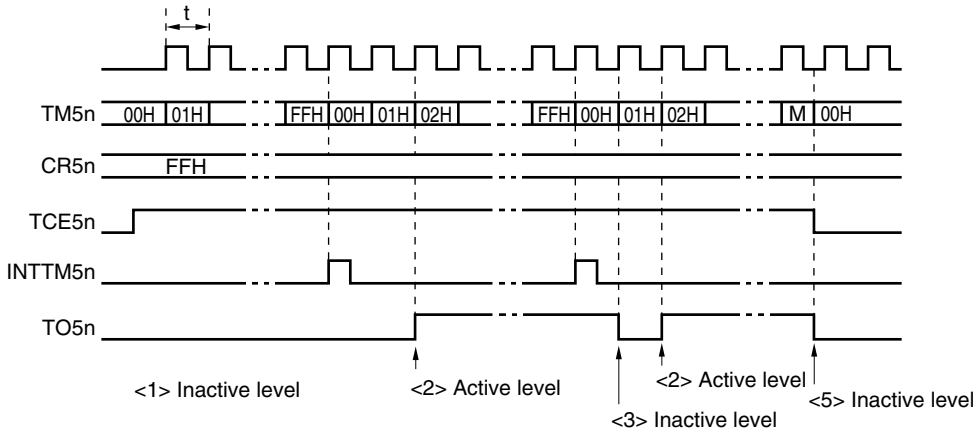
(a) Basic operation (active level = H)



(b) CR5n = 00H



(c) CR5n = FFH



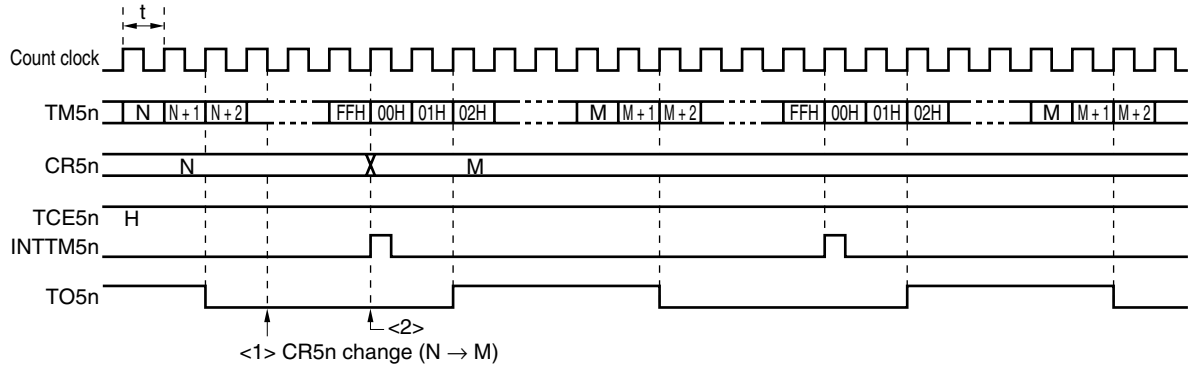
Remarks 1. <1> to <3> and <5> in Figure 7-14 (a) and (c) correspond to <1> to <3> and <5> in PWM output operation in **7.4.4 (1) PWM output basic operation**.

2. n = 0, 1

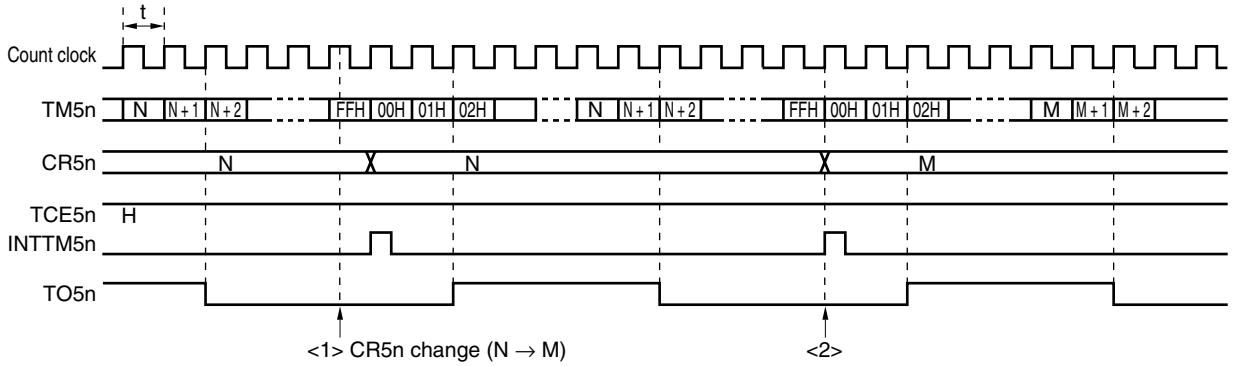
(2) Operation with CR5n changed

Figure 7-15. Timing of Operation with CR5n Changed

(a) CR5n value is changed from N to M before clock rising edge of FFH
→ Value is transferred to CR5n at overflow immediately after change.



(b) CR5n value is changed from N to M after clock rising edge of FFH
→ Value is transferred to CR5n at second overflow.



Caution When reading from CR5n between <1> and <2> in Figure 7-15, the value read differs from the actual value (read value: M, actual value of CR5n: N).

7.4.5 Carrier generator operation (8-bit timer 51 only)

TM51 and TMH1 can be used in combination as a carrier generator mode.

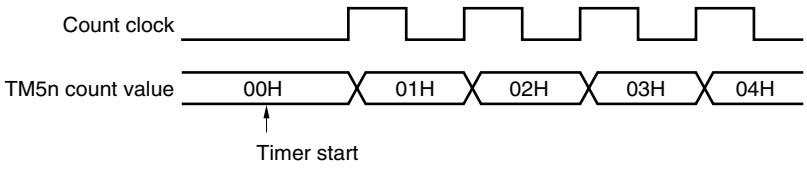
For details, see **8.4.3 Carrier generator operation (8-bit timer H1 only)**.

7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counters 50 and 51 (TM50, TM51) are started asynchronously to the count clock.

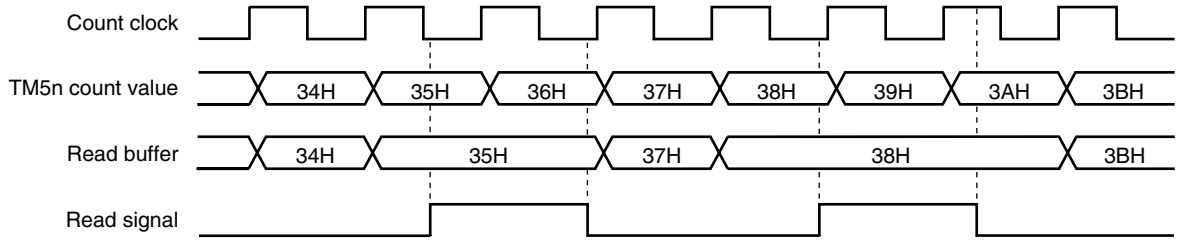
Figure 7-16. 8-Bit Timer Counter 5n (TM5n) Start Timing



(2) Reading of 8-bit timer counter 5n (TM5n)

TM5n can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 7-17. 8-bit Timer Counter 5n (TM5n) Read Timing



Remark n = 0, 1

CHAPTER 8 8-BIT TIMERS H0 AND H1

8.1 Functions of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 have the following functions.

- Interval timer
- Square-wave output
- PWM output
- Carrier generator (8-bit timer H1 only)[Note]

Note TM51 and TMH1 can be used in combination as a carrier generator mode.

8.2 Configuration of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 include the following hardware.

Table 8-1. Configuration of 8-Bit Timers H0 and H1

Item	Configuration
Timer register	8-bit timer counter Hn
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	TOHn, output controller
Control registers	8-bit timer H mode register n (TMHMDn) 8-bit timer H carrier control register 1 (TMCYC1)[Note] Port mode register 1 (PM1) Port register 1 (P1)

Note 8-bit timer H1 only

Remark n = 0, 1

Figures 8-1 and 8-2 show the block diagrams.

Figure 8-1. Block Diagram of 8-Bit Timer H0

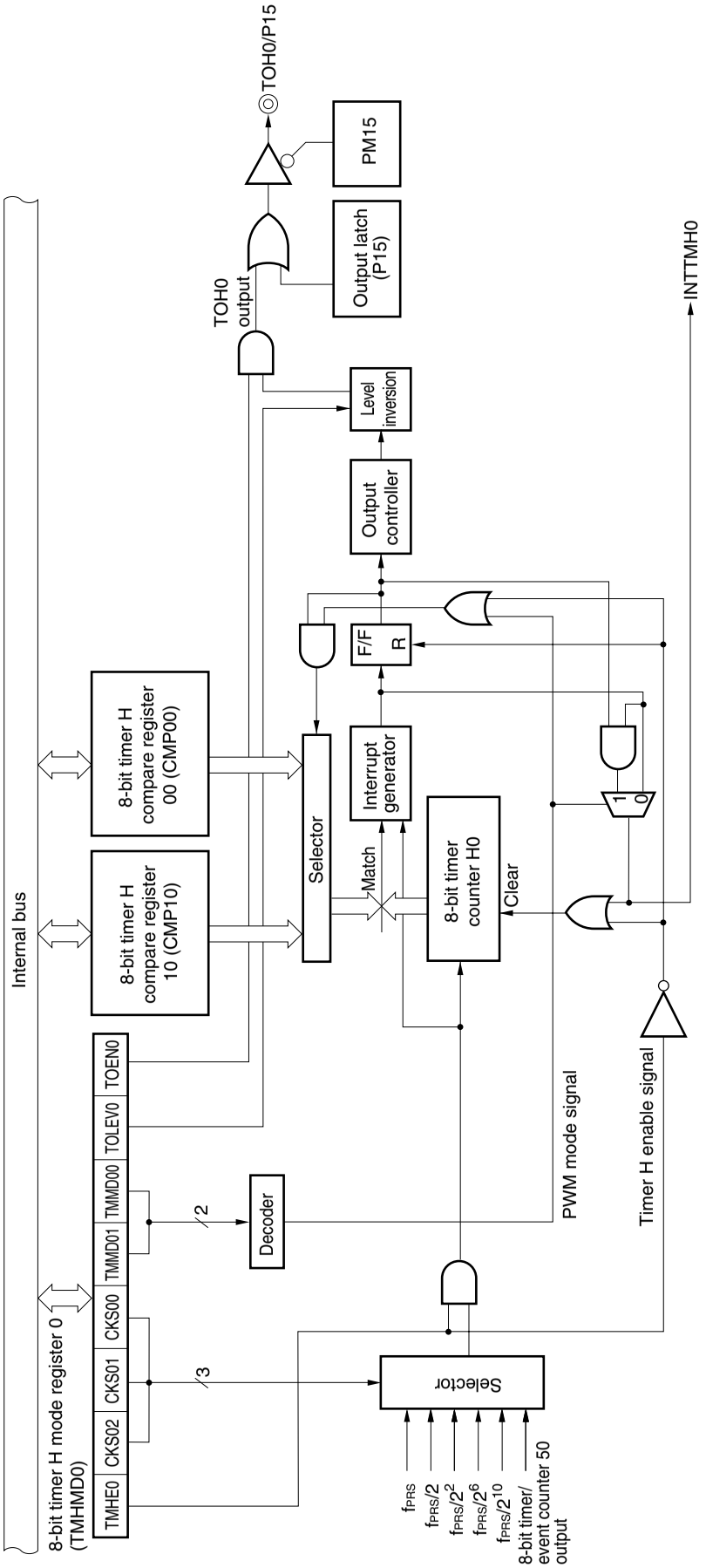
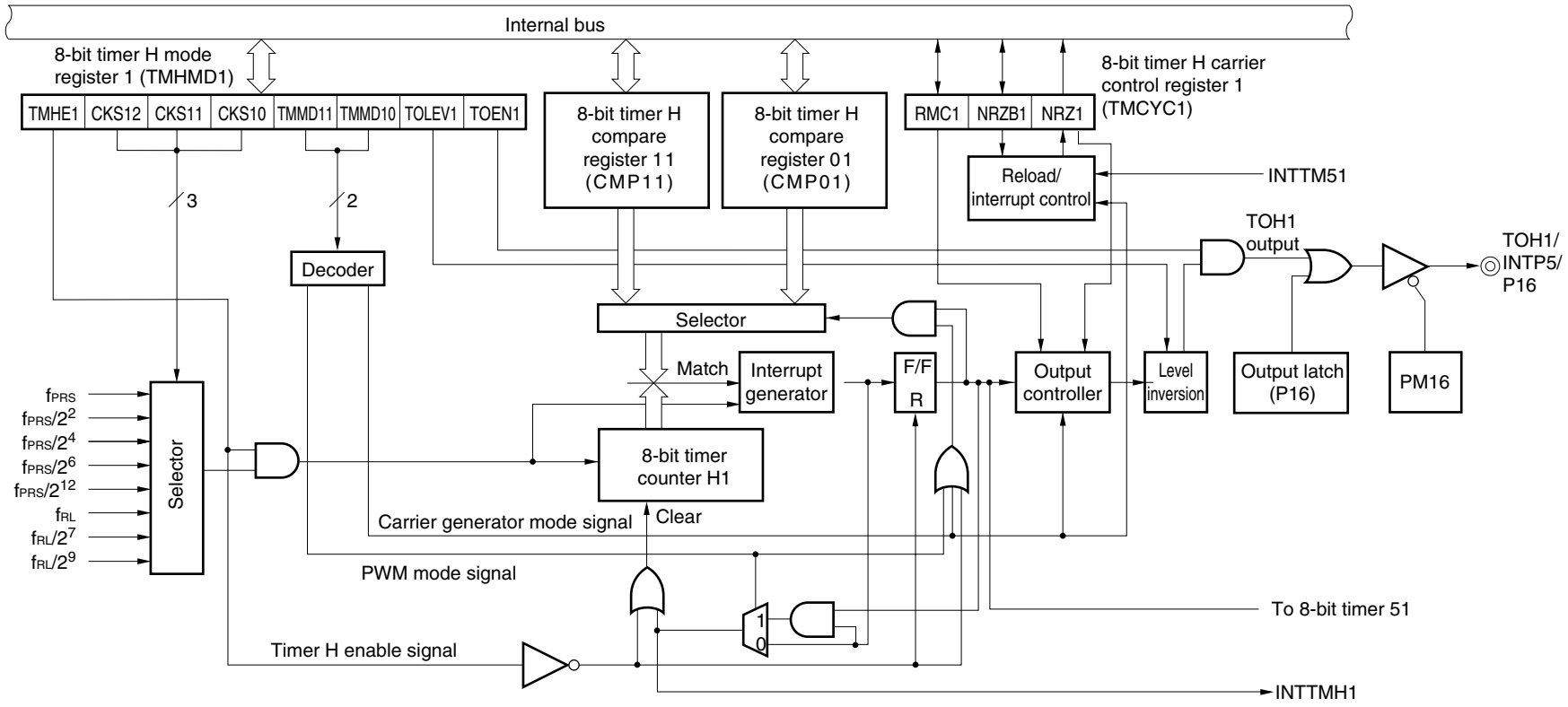


Figure 8-2. Block Diagram of 8-Bit Timer H1



(1) 8-bit timer H compare register 0n (CMP0n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in all of the timer operation modes.

This register constantly compares the value set to CMP0n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn) and inverts the output level of TOHn.

Rewrite the value of CMP0n while the timer is stopped (TMHEn = 0).

A reset signal generation clears this register to 00H.

Figure 8-3. Format of 8-Bit Timer H Compare Register 0n (CMP0n)

Address: FF18H (CMP00), FF1AH (CMP01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP0n (n = 0, 1)								

Caution CMP0n cannot be rewritten during timer count operation. CMP0n can be refreshed (the same value is written) during timer count operation.

(2) 8-bit timer H compare register 1n (CMP1n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in the PWM output mode and carrier generator mode.

In the PWM output mode, this register constantly compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, inverts the output level of TOHn. No interrupt request signal is generated.

In the carrier generator mode, the CMP1n register always compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn). At the same time, the count value is cleared.

CMP1n can be refreshed (the same value is written) and rewritten during timer count operation.

If the value of CMP1n is rewritten while the timer is operating, the new value is latched and transferred to CMP1n when the count value of the timer matches the old value of CMP1n, and then the value of CMP1n is changed to the new value. If matching of the count value and the CMP1n value and writing a value to CMP1n conflict, the value of CMP1n is not changed.

A reset signal generation clears this register to 00H.

Figure 8-4. Format of 8-Bit Timer H Compare Register 1n (CMP1n)

Address: FF19H (CMP10), FF1BH (CMP11) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP1n (n = 0, 1)								

Caution In the PWM output mode and carrier generator mode, be sure to set CMP1n when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to CMP1n).

Remark n = 0, 1

8.3 Registers Controlling 8-Bit Timers H0 and H1

The following four registers are used to control 8-bit timers H0 and H1.

- 8-bit timer H mode register n (TMHMDn)
- 8-bit timer H carrier control register 1 (TMCYC1)[Note]
- Port mode register 1 (PM1)
- Port register 1 (P1)

Note 8-bit timer H1 only

(1) 8-bit timer H mode register n (TMHMDn)

This register controls the mode of timer H.

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

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 8-5. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock selection				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	0	0	$f_{PRS}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.54 kHz
1	0	1	TM50 output[Note]				
Other than above			Setting prohibited				

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

Note Note the following points when selecting the TM50 output as the count clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Caution 1. When TMHE0 = 1, setting the other bits of TMHMD0 is prohibited. However, TMHMD0 can be refreshed (the same value is written).

Cautions 2. In the PWM output mode, be sure to set the 8-bit timer H compare register 10 (CMP10) when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP10).

3. The actual TOH0/P15 pin output is determined depending on PM15 and P15, besides TOH0 output.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)

3. TMC501: Bit 1 of TMC50

Figure 8-6. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock selection				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	0	0	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz	4.88 kHz
1	0	1	$f_{RL}/2^7$	1.88 kHz (TYP.)			
1	1	0	$f_{RL}/2^9$	0.47 kHz (TYP.)			
1	1	1	f_{RL}	240 kHz (TYP.)			

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
0	1	Carrier generator mode
1	0	PWM output mode
1	1	Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disables output
1	Enables output

Cautions 1. When TMHE1 = 1, setting the other bits of TMHMD1 is prohibited. However, TMHMD1 can be refreshed (the same value is written).

2. In the PWM output mode and carrier generator mode, be sure to set the 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).

3. When the carrier generator mode is used, set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

4. The actual TOH1/INTP5/P16 pin output is determined depending on PM16 and P16, besides TOH1 output.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{RL}: Internal low-speed oscillation clock frequency

(2) 8-bit timer H carrier control register 1 (TMCYC1)

This register controls the remote control output and carrier pulse output status of 8-bit timer H1.

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

Reset signal generation clears this register to 00H.

Figure 8-7. Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)

Address: FF6DH After reset: 00H R/W[Note]

	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	Remote control output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

NRZ1	Carrier pulse output status flag
0	Carrier output disabled status (low-level status)
1	Carrier output enabled status (RMC1 = 1: Carrier pulse output, RMC1 = 0: High-level status)

Note Bit 0 is read-only.

Caution Do not rewrite RMC1 when TMHE = 1. However, TMCYC1 can be refreshed (the same value is written).

(3) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using the P15/TOH0 and P16/TOH1/INTP5 pins for timer output, clear PM15 and PM16 and the output latches of P15 and P16 to 0.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 8-8. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operation of 8-Bit Timers H0 and H1

8.4.1 Operation as interval timer/square-wave output

When the 8-bit timer counter Hn and compare register 0n (CMP0n) match, an interrupt request signal (INTTMHn) is generated and the 8-bit timer counter Hn is cleared to 00H.

Compare register 1n (CMP1n) is not used in interval timer mode. Since a match of the 8-bit timer counter Hn and the CMP1n register is not detected even if the CMP1n register is set, timer output is not affected.

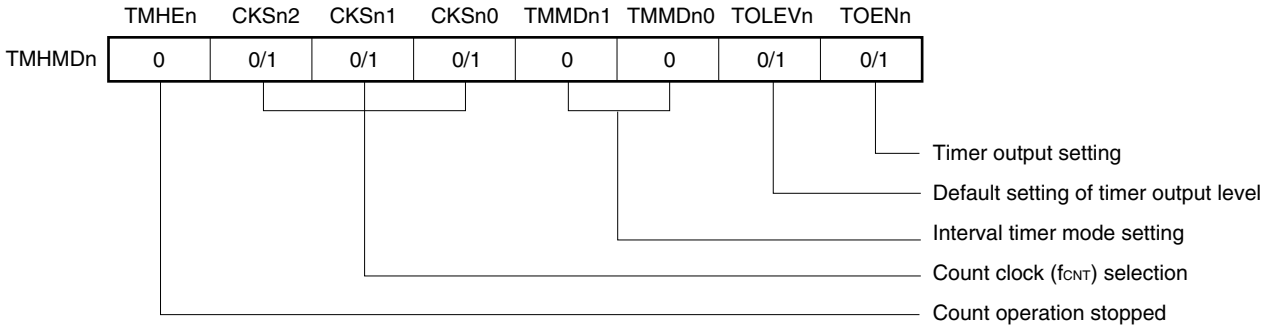
By setting bit 0 (TOENn) of timer H mode register n (TMHMDn) to 1, a square wave of any frequency (duty = 50%) is output from TOHn.

Setting

<1> Set each register.

Figure 8-9. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register n (TMHMDn)



(ii) CMP0n register setting

The interval time is as follows if N is set as a comparison value.

- Interval time = (N +1)/f_{CNT}

<2> Count operation starts when TMHEn = 1.

<3> When the values of the 8-bit timer counter Hn and the CMP0n register match, the INTTMHn signal is generated and the 8-bit timer counter Hn is cleared to 00H.

<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, clear TMHEn to 0.

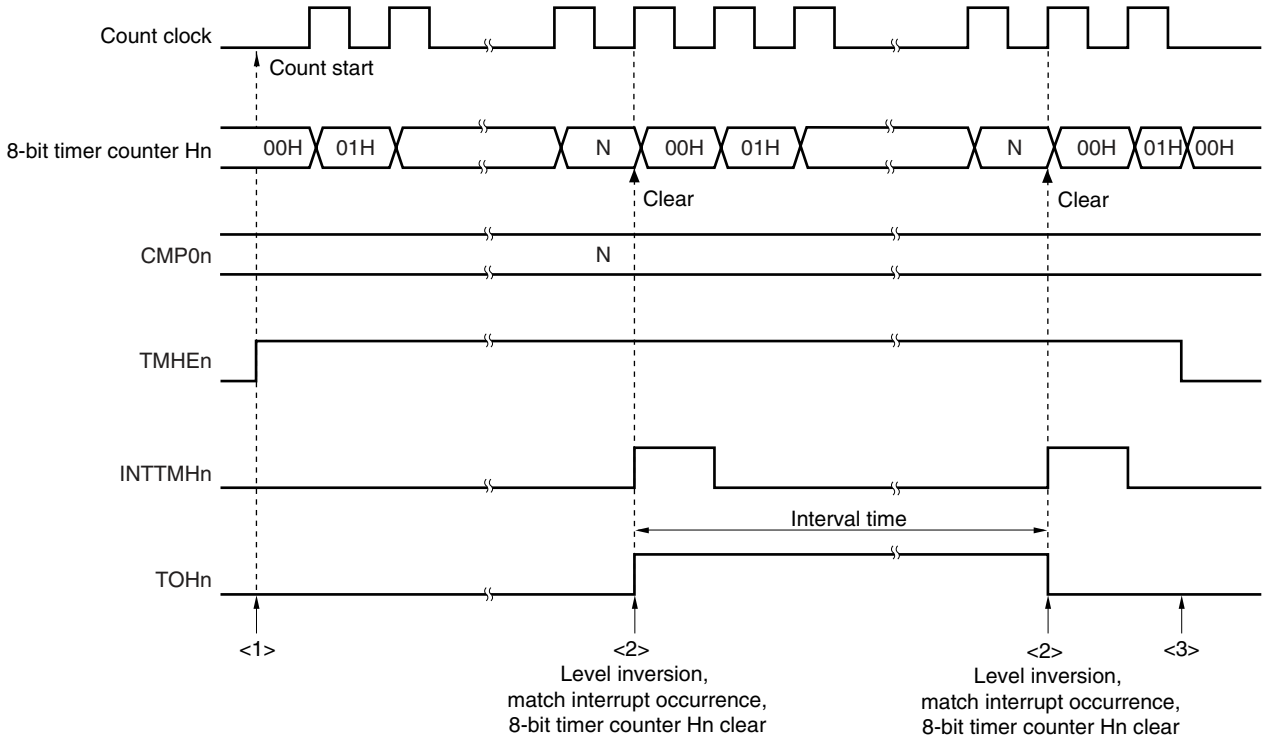
Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 1 (PM1)**.

2. For how to enable the INTTMHn signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

3. n = 0, 1

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation (Operation When 01H ≤ CMP0n ≤ FEH)



<1> The count operation is enabled by setting the TMHEn bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

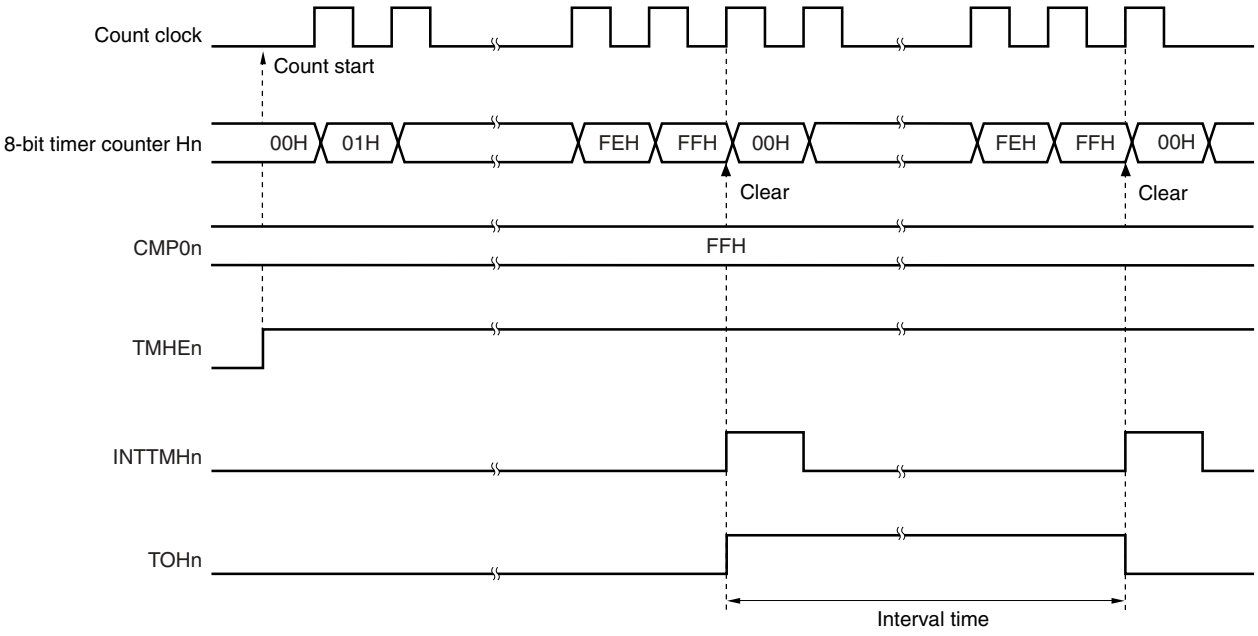
<2> When the value of the 8-bit timer counter Hn matches the value of the CMP0n register, the value of the timer counter is cleared, and the level of the TOHn output is inverted. In addition, the INTTMHn signal is output at the rising edge of the count clock.

<3> If the TMHEn bit is cleared to 0 while timer H is operating, the INTTMHn signal and TOHn output are set to the default level. If they are already at the default level before the TMHEn bit is cleared to 0, then that level is maintained.

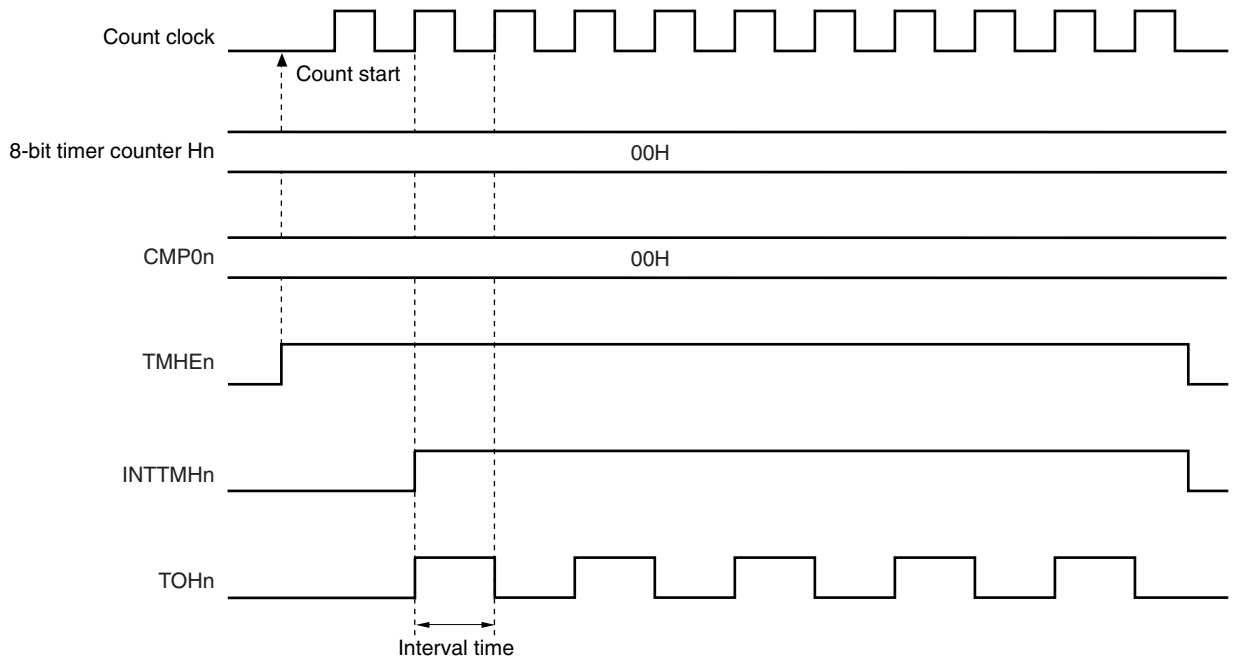
Remark n = 0, 1
01H ≤ N ≤ FEH

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



Remark n = 0, 1

8.4.2 Operation as PWM output

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

The 8-bit timer compare register 0n (CMP0n) controls the cycle of timer output (TOHn). Rewriting the CMP0n register during timer operation is prohibited.

The 8-bit timer compare register 1n (CMP1n) controls the duty of timer output (TOHn). Rewriting the CMP1n register during timer operation is possible.

The operation in PWM output mode is as follows.

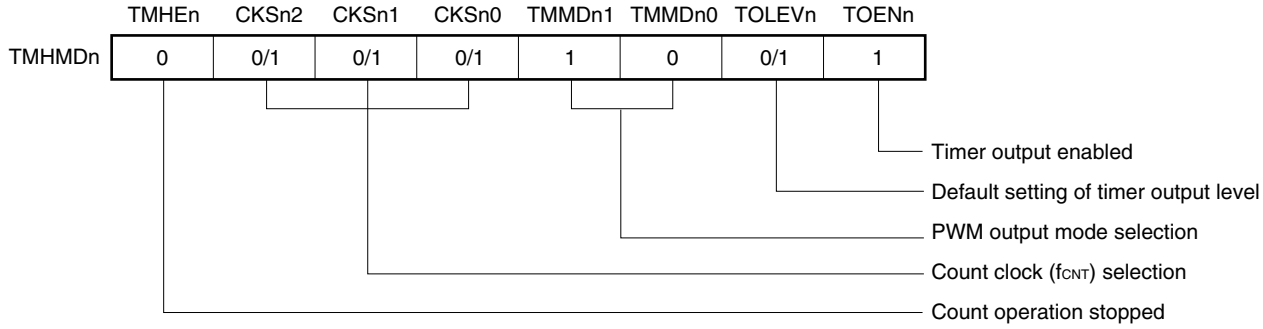
PWM output (TOHn output) outputs an active level and 8-bit timer counter Hn is cleared to 0 when 8-bit timer counter Hn and the CMP0n register match after the timer count is started. PWM output (TOHn output) outputs an inactive level when 8-bit timer counter Hn and the CMP1n register match.

Setting

<1> Set each register.

Figure 8-11. Register Setting in PWM Output Mode

(i) Setting timer H mode register n (TMHMDn)



(ii) Setting CMP0n register

- Compare value (N): Cycle setting

(iii) Setting CMP1n register

- Compare value (M): Duty setting

Remarks 1. n = 0, 1

2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of the 8-bit timer counter Hn and the CMP0n register match, the 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, and an active level is output. At the same time, the compare register to be compared with the 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

<4> When the 8-bit timer counter Hn and the CMP1n register match, an inactive level is output and the compare register to be compared with the 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, the 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHEn = 0.

If the setting value of the CMP0n register is N, the setting value of the CMP1n register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

- PWM pulse output cycle = (N + 1)/f_{CNT}
- Duty = (M + 1)/(N + 1)

Cautions 1. The set value of the CMP1n register can be changed while the timer counter is operating. However, this takes a duration of three operating clocks (signal selected by the CKSn2 to CKSn0 bits of the TMHMDn register) from when the value of the CMP1n register is changed until the value is transferred to the register.

2. Be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

3. Make sure that the CMP1n register setting value (M) and CMP0n register setting value (N) are within the following range.

00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

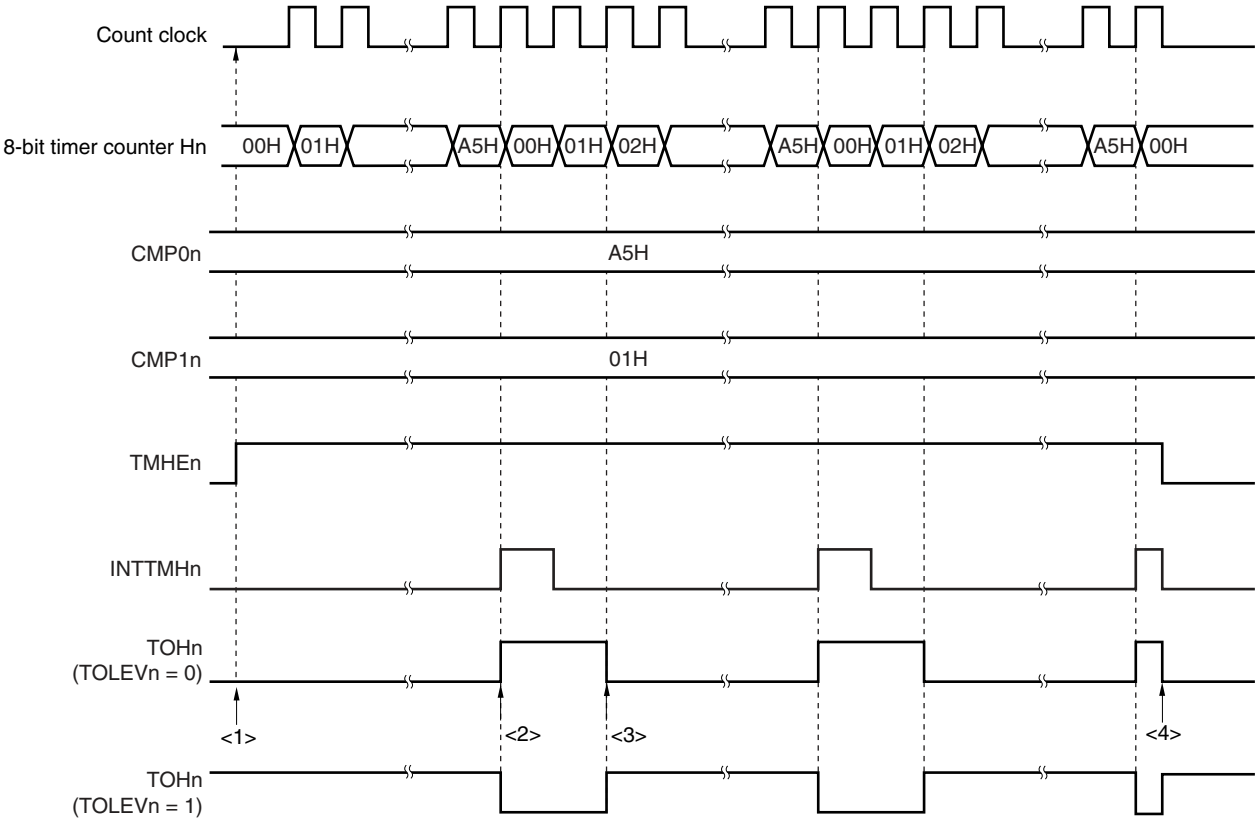
Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 1 (PM1)**.

2. For details on how to enable the INTTMHn signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

3. n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. Start the 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> When the values of the 8-bit timer counter Hn and the CMP0n register match, an active level is output. At this time, the value of the 8-bit timer counter Hn is cleared, and the INTTMHn signal is output.

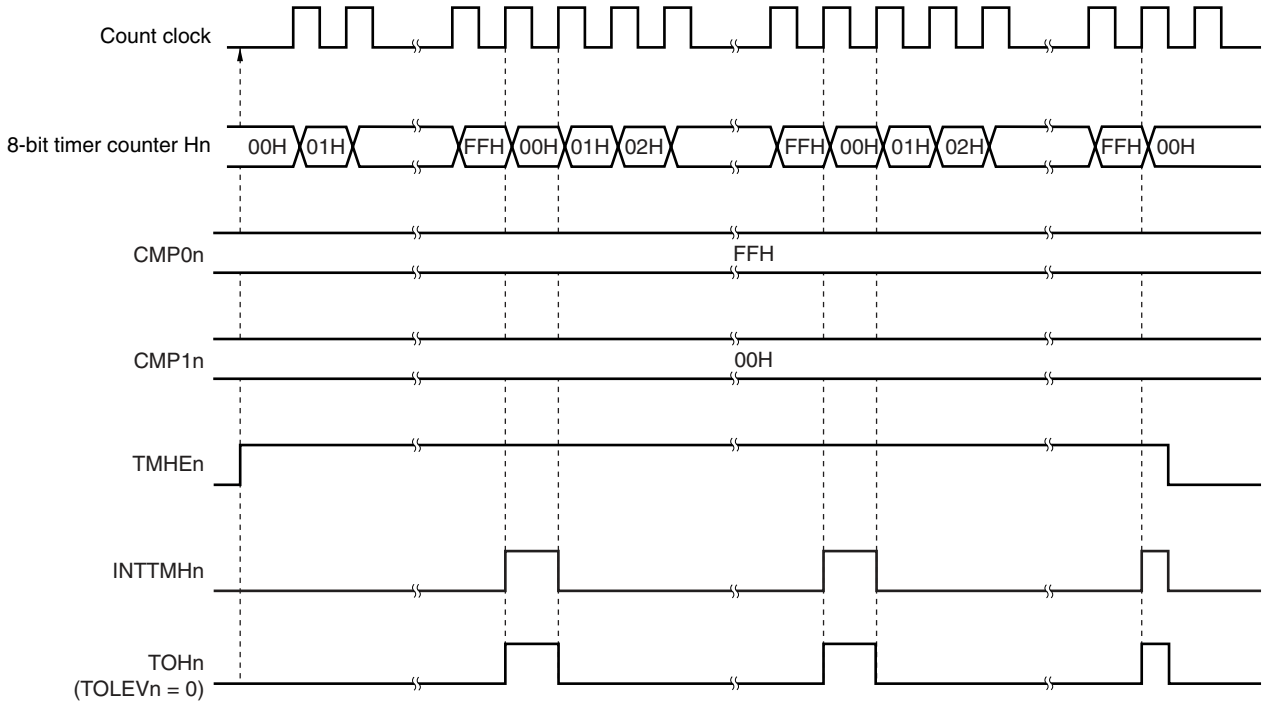
<3> When the values of the 8-bit timer counter Hn and the CMP1n register match, an inactive level is output. At this time, the 8-bit timer counter value is not cleared and the INTTMHn signal is not output.

<4> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

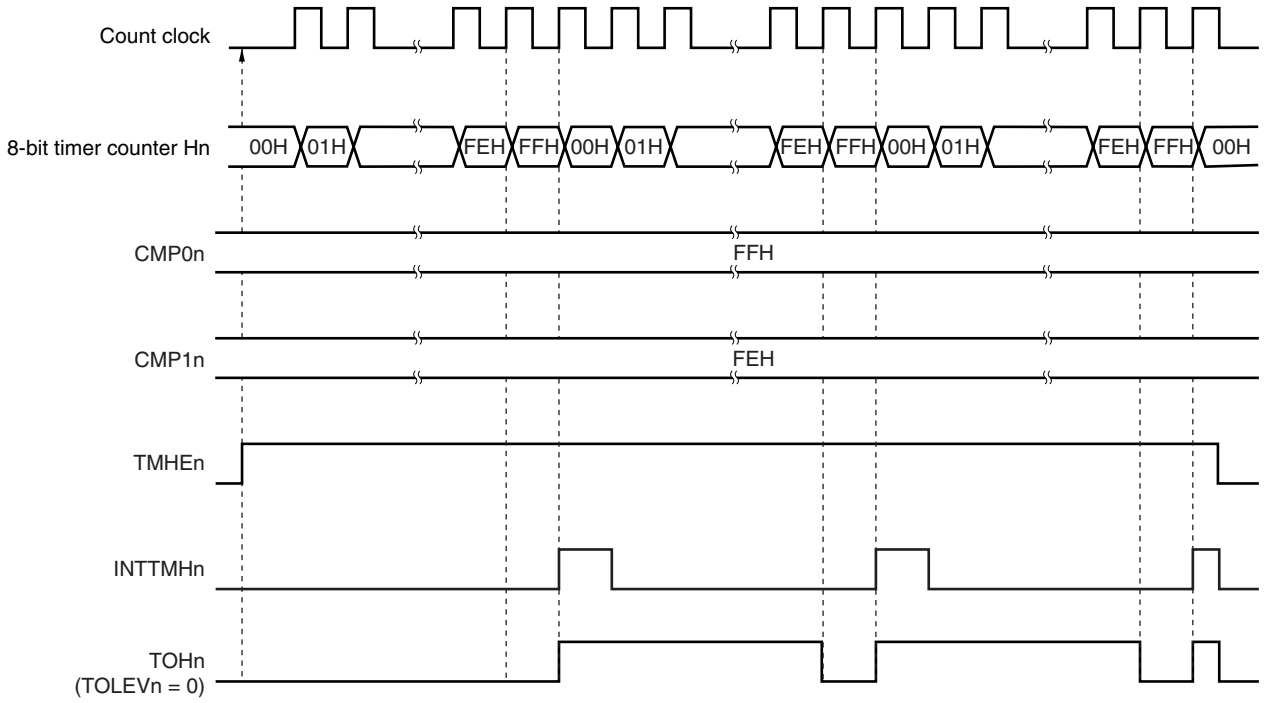
Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP0n = FFH, CMP1n = 00H



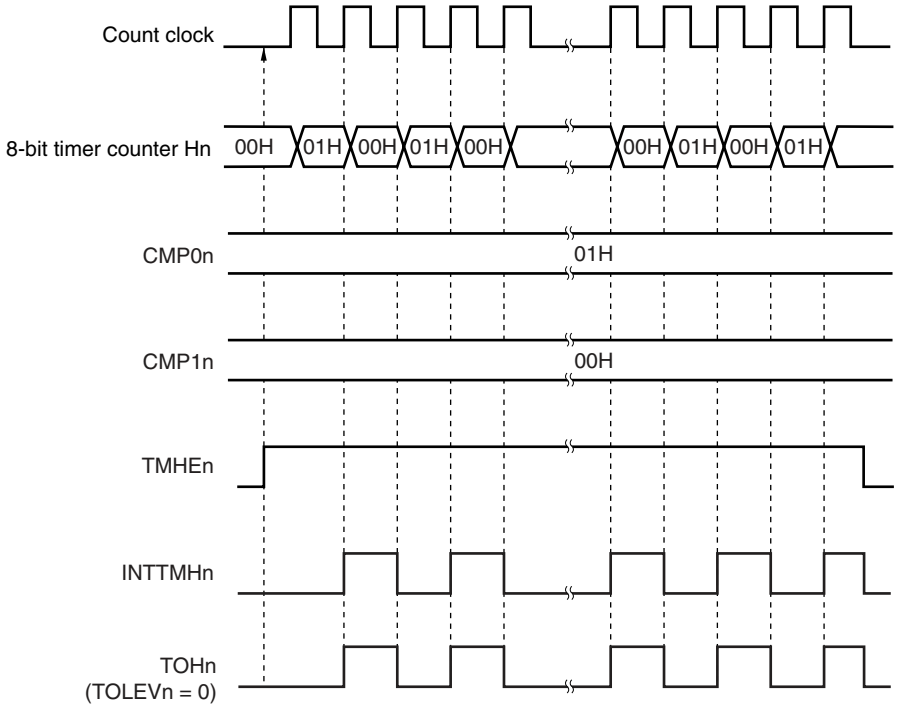
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (3/4)

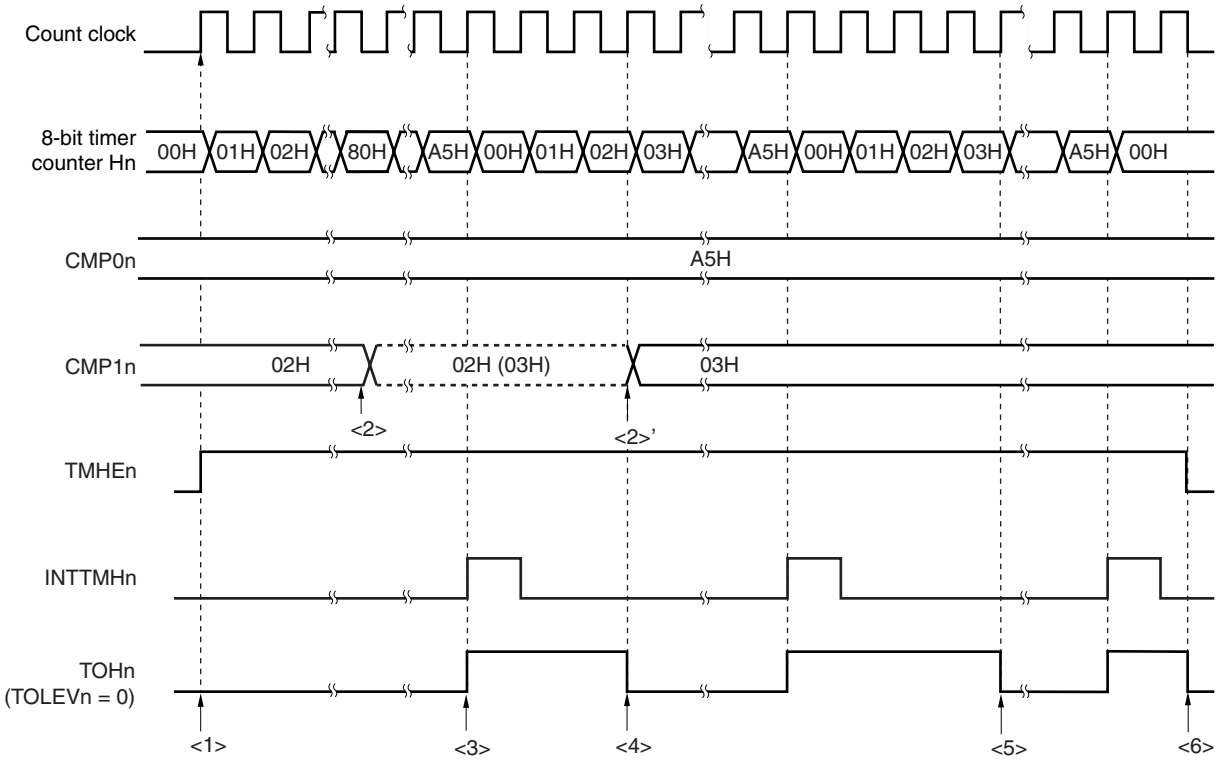
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP1n (CMP1n = 02H → 03H, CMP0n = A5H)



<1> The count operation is enabled by setting TMHEn = 1. Start the 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> The CMP1n register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of the 8-bit timer counter Hn and the CMP0n register match, the value of the 8-bit timer counter Hn is cleared, an active level is output, and the INTTMHn signal is output.

<4> If the CMP1n register value is changed, the value is latched and not transferred to the register. When the values of the 8-bit timer counter Hn and the CMP1n register before the change match, the value is transferred to the CMP1n register and the CMP1n register value is changed (<2>').

However, three count clocks or more are required from when the CMP1n register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of the 8-bit timer counter Hn and the CMP1n register after the change match, an inactive level is output. The 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<6> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

Remark n = 0, 1

8.4.3 Carrier generator operation (8-bit timer H1 only)

In the carrier generator mode, the 8-bit timer H1 is used to generate the carrier signal of an infrared remote controller, and the 8-bit timer/event counter 51 is used to generate an infrared remote control signal (time count).

The carrier clock generated by the 8-bit timer H1 is output in the cycle set by the 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by the 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

(1) Carrier generation

In carrier generator mode, the 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and the 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during the 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

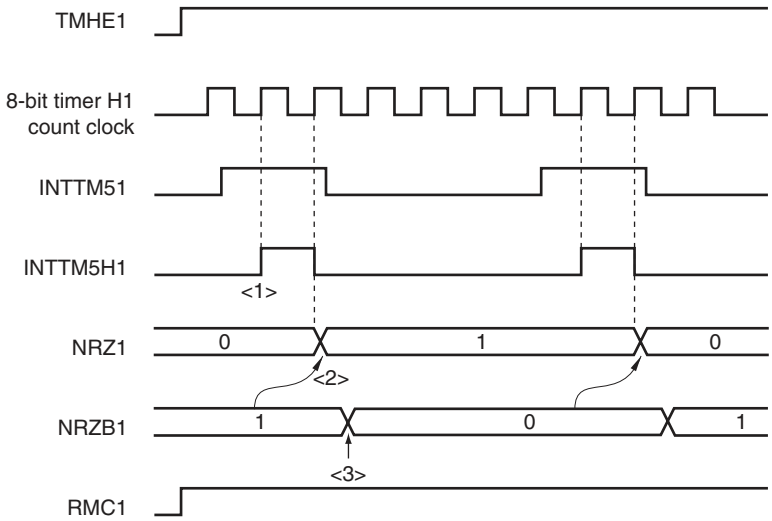
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of the 8-bit timer/event counter 51 and the NRZB1 and RMC1 bits of the 8-bit timer H carrier control register (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZB1 Bit	Output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

To control the carrier pulse output during a count operation, the NRZ1 and NRZB1 bits of the TMCYC1 register have a master and slave bit configuration. The NRZ1 bit is read-only but the NRZB1 bit can be read and written. The INTTM51 signal is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal of the NRZ1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit. The timing for transfer from the NRZB1 bit to the NRZ1 bit is as shown below.

Figure 8-13. Transfer Timing



<1> The INTTM51 signal is synchronized with the count clock of the 8-bit timer H1 and is output as the INTTM5H1 signal.

<2> The value of the NRZB1 bit is transferred to the NRZ1 bit at the second clock from the rising edge of the INTTM5H1 signal.

<3> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

Cautions 1. Do not rewrite the NRZB1 bit again until at least the second clock after it has been rewritten, or else the transfer from the NRZB1 bit to the NRZ1 bit is not guaranteed.

2. When the 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When the 8-bit timer/event counter 51 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

Remark INTTM5H1 is an internal signal and not an interrupt source.

Setting

<1> Set each register.

Figure 8-14. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)

(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... carrier output enable bit

(v) TCL51 and TMC51 register setting

- See **8.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51**.

<2> When TMHE1 = 1, the 8-bit timer H1 starts counting.

<3> When TCE51 of the 8-bit timer mode control register 51 (TMC51) is set to 1, the 8-bit timer/event counter 51 starts counting.

<4> After the count operation is enabled, the first compare register to be compared is the CMP01 register. When the count value of the 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of the 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register.

<6> By performing procedures <4> and <5> repeatedly, a carrier clock is generated.

<7> The INTTM51 signal is synchronized with count clock of the 8-bit timer H1 and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<8> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

<9> When the NRZ1 bit is high level, a carrier clock is output by TOH1 output.

<10> By performing the procedures above, an arbitrary carrier clock is obtained. To stop the count operation, clear TMHE1 to 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the carrier clock output cycle and duty are as follows.

- Carrier clock output cycle = (N + M + 2)/f_{CNT}
- Duty = High-level width/carrier clock output width = (M + 1)/(N + M + 2)

Cautions 1. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

2. Set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

3. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

4. The set value of the CMP11 register can be changed while the timer counter is operating. However, it takes the duration of three operating clocks (signal selected by the CKS12 to CKS10 bits of the TMHMD1 register) since the value of the CMP11 register has been changed until the value is transferred to the register.

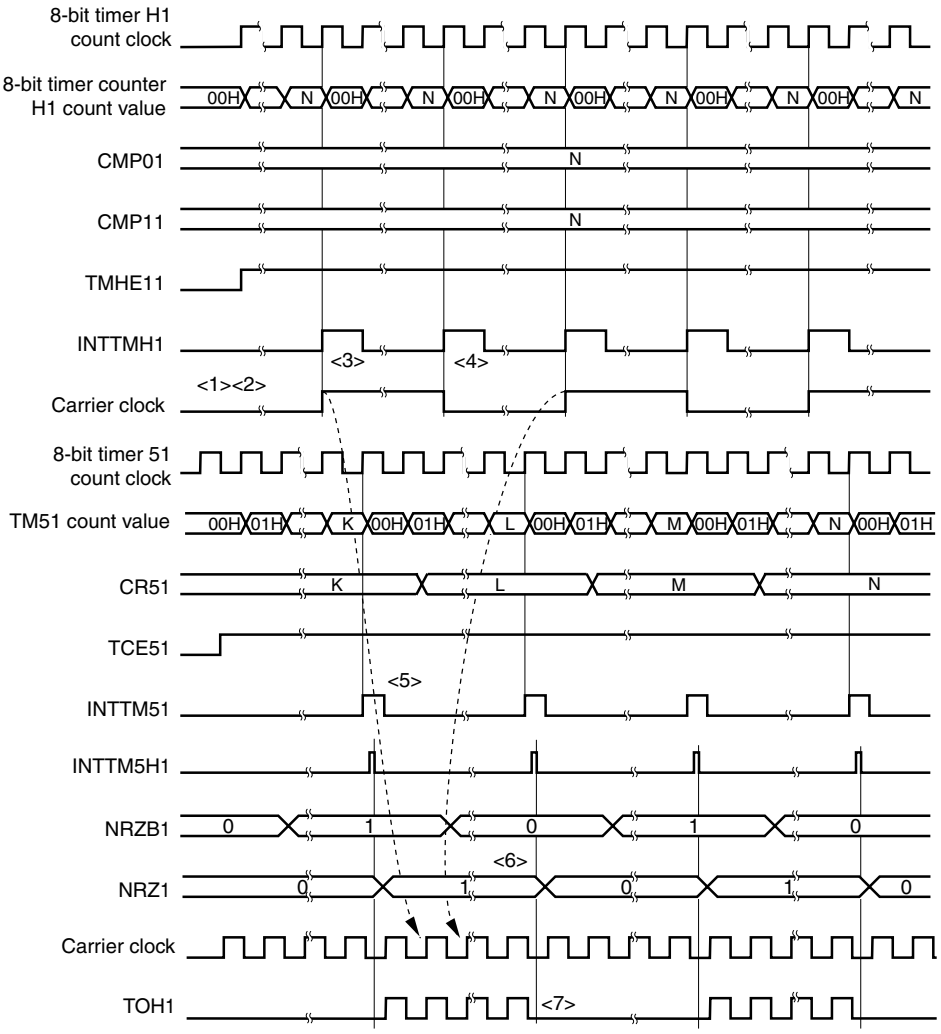
5. Be sure to set the RMC1 bit before the count operation is started.

Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode registers 1, 3 (PM1, PM3)**.

2. For how to enable the INTTMH1 signal interrupt, see **CHAPTER 20 INTERRUPT FUNCTIONS**.

Figure 8-15. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = N, CMP11 = N



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

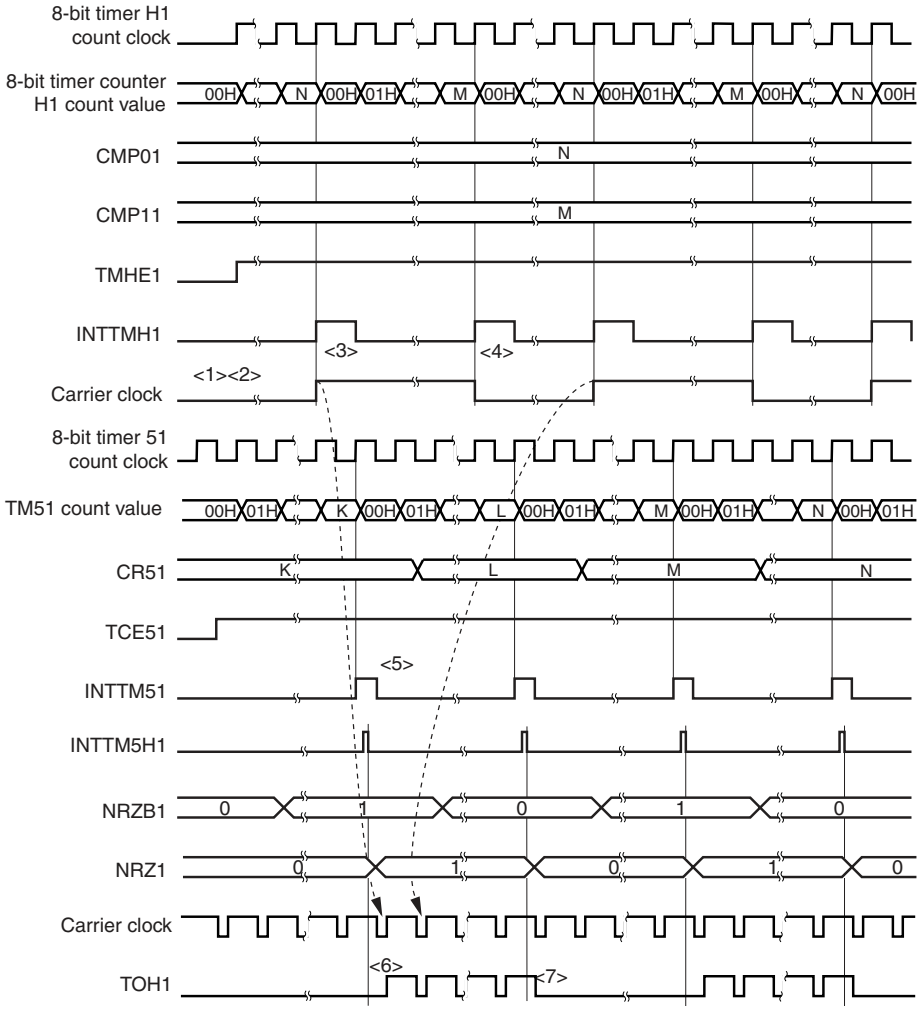
<6> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<7> When NRZ1 = 0 is set, the TOH1 output becomes low level.

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-15. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = N, CMP11 = M



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to other than 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

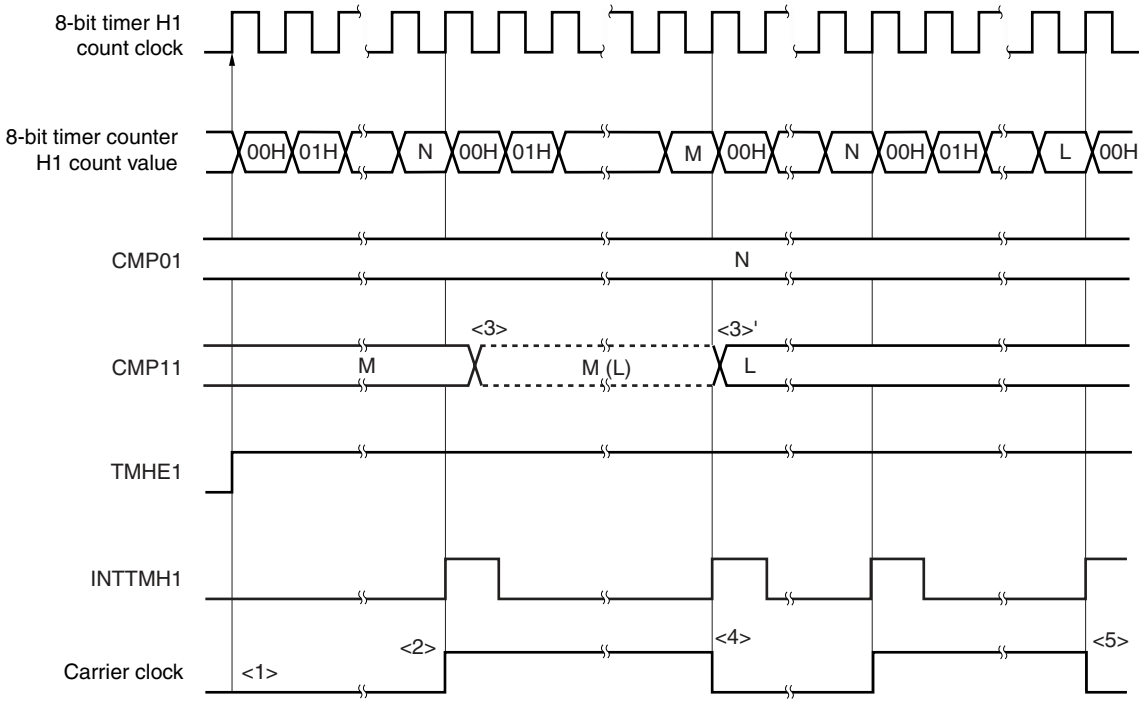
<6> A carrier signal is output at the first rising edge of the carrier clock if NRZ1 is set to 1.

<7> When NRZ1 = 0, the TOH1 output is held at the high level and is not changed to low level while the carrier clock is high level (from <6> and <7>, the high-level width of the carrier clock waveform is guaranteed).

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-15. Carrier Generator Mode Operation Timing (3/3)

(c) Operation when CMP11 is changed



<1> When TMHE1 = 1 is set, the 8-bit timer H1 starts a count operation. At that time, the carrier clock remains default.

<2> When the count value of the 8-bit timer counter H1 matches the value of the CMP01 register, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP01 register to the CMP11 register.

<3> The CMP11 register is asynchronous to the count clock, and its value can be changed while the 8-bit timer H1 is operating. The new value (L) to which the value of the register is to be changed is latched. When the count value of the 8-bit timer counter H1 matches the value (M) of the CMP11 register before the change, the CMP11 register is changed (<3>').

However, it takes three count clocks or more since the value of the CMP11 register has been changed until the value is transferred to the register. Even if a match signal is generated before the duration of three count clocks elapses, the new value is not transferred to the register.

<4> When the count value of 8-bit timer counter H1 matches the value (M) of the CMP1 register before the change, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register.

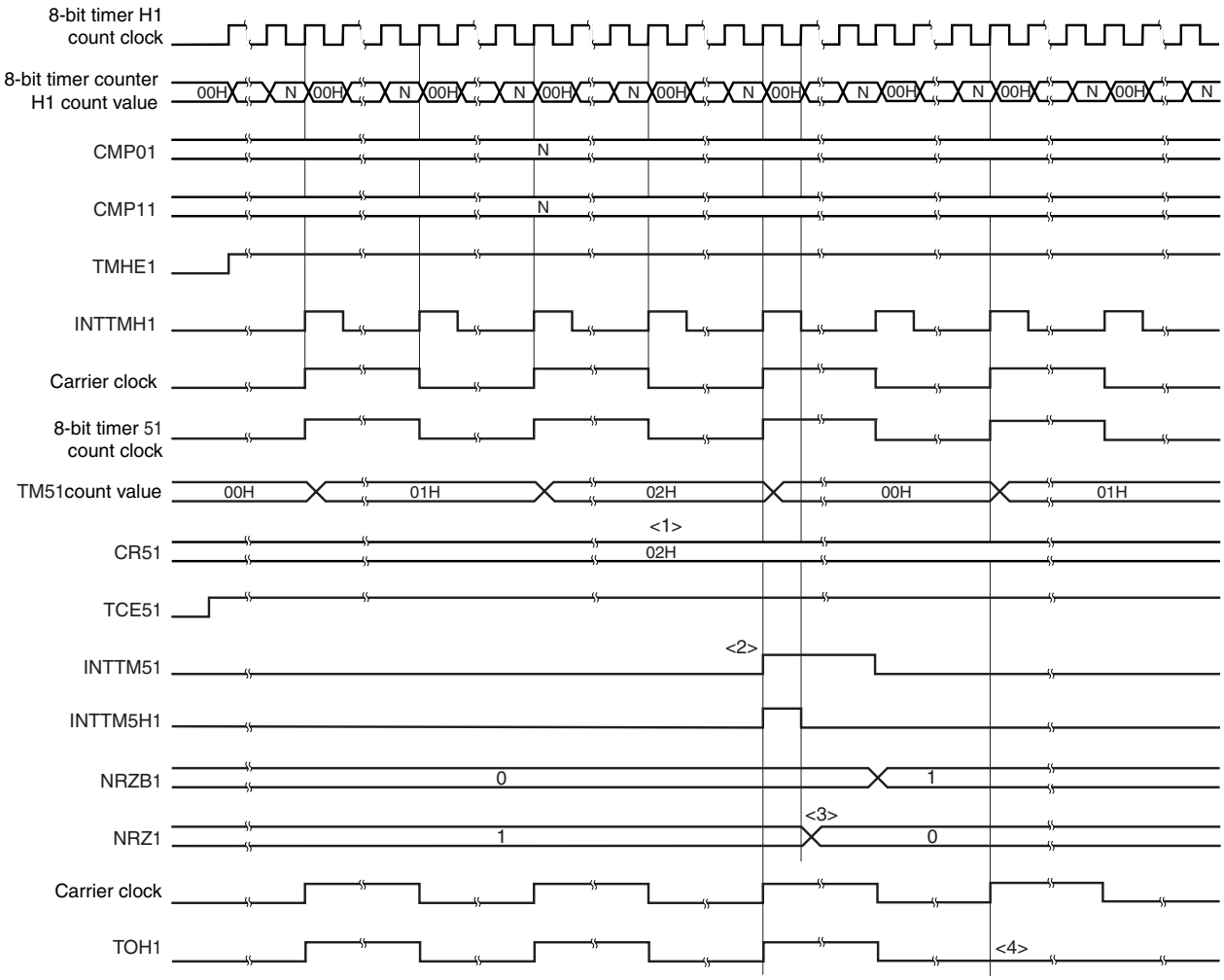
<5> The timing at which the count value of the 8-bit timer counter H1 and the CMP11 register value match again is indicated by the value after the change (L).

8.4.4 Control of number of carrier clocks by timer 51 counter

The number of carrier clocks to be output from the TOH1 pin can be controlled by selecting the timer H1 output signal for the 8-bit timer 51 count clock.

Figure 8-16 shows an example of control when three carrier clocks are to be output from the TOH1 pin

Figure 8-16. Example of Controlling Number of Carrier Clocks by Timer 51 Counter (Setting Timer H1 Output Signal for Timer 51 Count Clock (TCL51 = 07H))



<1> Set the CR51 register to 02H when three carrier clocks are to be output from the TOH1 pin.

<2> The INTTM51 signal is generated when the TM51 count value and the CR51 register value (02H) have matched. The signal is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

<3> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.
The transfer timing at this time is one timer H1 count clock after a rise of the INTTM5H1 signal.

<4> By setting NRZ1 to 0, the TOH1 output becomes low level after having output the third carrier clock.

Remark INTTM5H1 is an internal signal and not an interrupt source.

CHAPTER 9 REAL-TIME COUNTER

Caution Only one of fixed-cycle signal of real-time counter/alarm match detection and interval signal detection of real-time counter can be used.

9.1 Functions of Real-Time Counter

The real-time counter has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 1 month to 0.5 seconds)
- Alarm interrupt function (alarm: week, hour, minute)
- Interval interrupt function
- Pin output function of 1 Hz
- Pin output function of 512 Hz or 16.384 kHz or 32.768 kHz

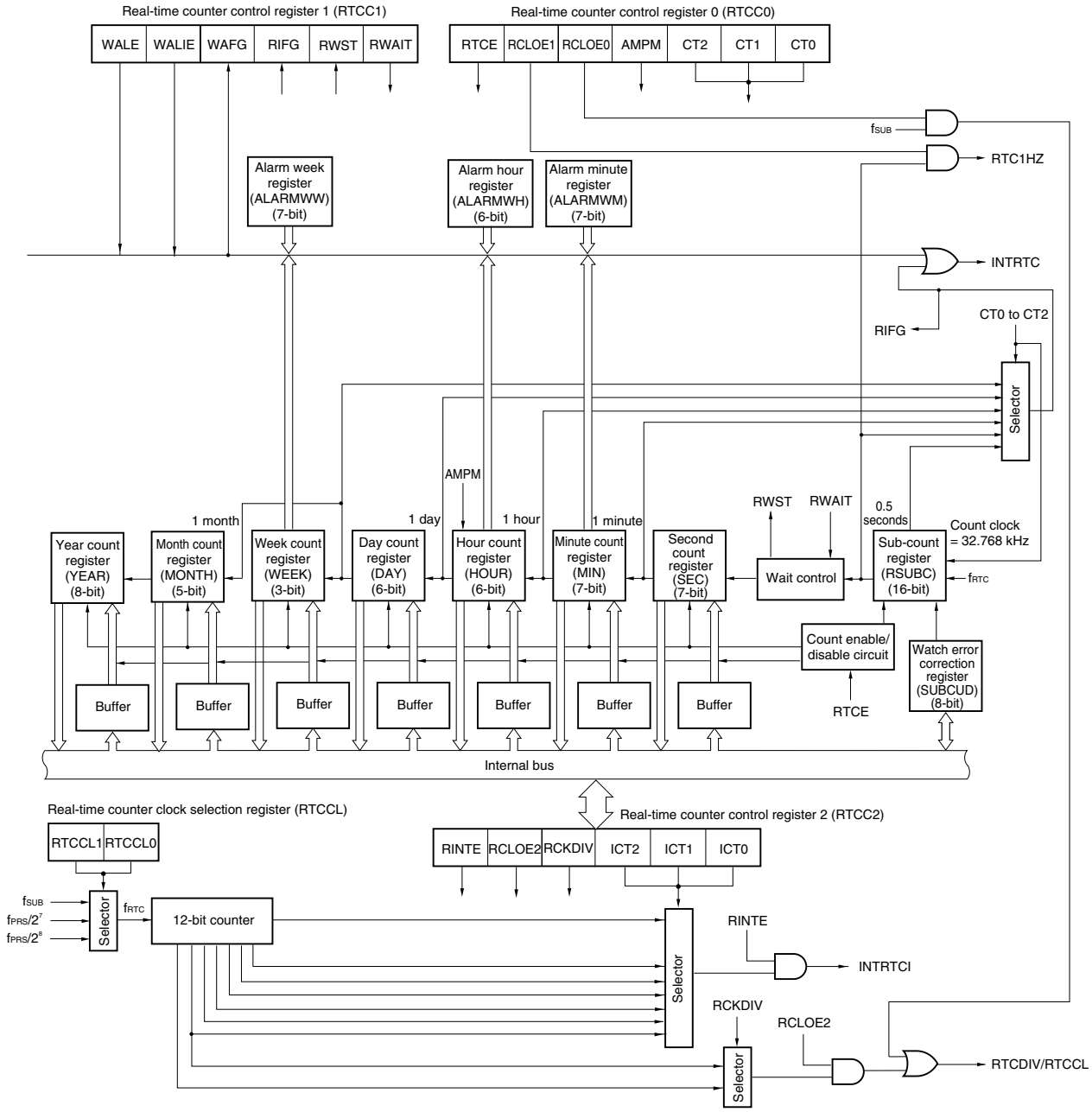
9.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

Table 9-1. Configuration of Real-Time Counter

Item	Configuration
Control registers	Real-time counter clock selection register (RTCCL) Real-time counter control register 0 (RTCC0) Real-time counter control register 1 (RTCC1) Real-time counter control register 2 (RTCC2) Sub-count register (RSUBC) Second count register (SEC) Minute count register (MIN) Hour count register (HOUR) Day count register (DAY) Week count register (WEEK) Month count register (MONTH) Year count register (YEAR) Watch error correction register (SUBCUD) Alarm minute register (ALARMWM) Alarm hour register (ALARMWH) Alarm week register (ALARMWW) Port mode register 4 (PM4) Port register 4 (P4)

Figure 9-1. Block Diagram of Real-Time Counter



9.3 Registers Controlling Real-Time Counter

Timer real-time counter is controlled by the following 18 registers.

- Real-time counter clock selection register (RTCCL)
- Real-time counter control register 0 (RTCC0)
- Real-time counter control register 1 (RTCC1)
- Real-time counter control register 2 (RTCC2)
- Sub-count register (RSUBC)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- Port mode register 4 (PM4)
- Port register 4 (P4)

(1) Real-time counter clock selection register (RTCCL)

This register controls the mode of real-time counter.

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

Reset signal generation clears this register to 00H.

Figure 9-2. Format of Real-time Counter Clock Selection Register (RTCCL)

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCCL	0	0	0	0	0	0	RTCCL1	RTCCL0

RTCCL1	RTCCL0	Control of real-time counter (RTC) input clock (f_{RTC})
0	0	f_{SUB}
0	1	$f_{PRS}/2^7$
1	0	$f_{PRS}/2^8$
1	1	Setting prohibited

Remark • When f_{PRS} = 4.19 MHz, $f_{RTC} = f_{PRS}/2^7$ = 32.768 kHz

• When f_{PRS} = 8.38 MHz, $f_{RTC} = f_{PRS}/2^8$ = 32.768 kHz

(2) Real-time counter control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time counter operation, control the RTCCL and RTC1HZ pins, and set a 12- or 24-hour system and the constant-period interrupt function.

RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-3. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FFADH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of RTC1HZ pin (1 Hz).
1	Enables output of RTC1HZ pin (1 Hz).

RCLOE0[Note]	RTCCL pin output control
0	Disables output of RTCCL pin (32.768 kHz).
1	Enables output of RTCCL pin (32.768 kHz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1, and re-set the hour count register (HOUR). • Table 9-2 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)
After changing the values of CT2 to CT0, clear the interrupt request flag.			

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, a pulse with a narrow width may be generated on the 32.768 kHz and 1 Hz output signals.

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 9-4. Format of Real-Time Counter Control Register 1 (RTCC1) (1/2)

Address: FFAEH 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.
To set the registers of alarm (WALIE flag of RTCC1, ALARMWM register, ALARMWH register, and ALARMWW register), disable WALE (clear it to “0”).	

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.	

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.	

Figure 9-4. Format of Real-Time Counter Control Register 1 (RTCC1) (2/2)

RWAIT	Wait control of real-time counter
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value

This bit controls the operation of the counter.
Be sure to write "1" to it to read or write the counter value.
Because RSUBC continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0.
When RWAIT = 1, it takes up to 1 clock (32.768 kHz) until the counter value can be read or written.
If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.

Caution 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 9-5. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: FFAFH 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	$2^6/f_{RTC}$ (1.953125 ms)
1	0	0	1	$2^7/f_{RTC}$ (3.90625 ms)
1	0	1	0	$2^8/f_{RTC}$ (7.8125 ms)
1	0	1	1	$2^9/f_{RTC}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{RTC}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{RTC}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{RTC}$ (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_{RTC} 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.

(5) Sub-count register (RSUBC)

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter. It takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

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

Reset signal generation clears this register to 0000H.

Cautions 1. When a correction is made by using the SUBCUD register, the value may become 8000H or more.

2. This register is also cleared by reset effected by writing the second count register.

3. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

Figure 9-6. Format of Sub-Count Register (RSUBC)

Address: F890H, F891H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

(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 this 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 9-7. Format of Second Count Register (SEC)

Address: F892H 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.

Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this 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 9-8. Format of Minute Count Register (MIN)

Address: F893H 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, 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.

Set a decimal value of 00 to 23 or 01 to 12, 21 to 32 to this register in BCD code. If a value outside this 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 9-9. Format of Hour Count Register (HOUR)

Address: F894H 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 9-2 shows the relationship between the setting value of the AMPM bit, the HOUR register value, and time.

Table 9-2. 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.
Set a decimal value of 01 to 31 to this register in BCD code. If a value outside this 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 9-10. Format of Day Count Register (DAY)

Address: F896H 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 this 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 9-11. Format of Week Count Register (WEEK)

Address: F895H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution The value corresponding to the month count register or the day count register is not stored in the week count register automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

(11) Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside this 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 9-12. Format of Month Count Register (MONTH)

Address: F897H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(12) Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month counter overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 99 to this register in BCD code. If a value outside this 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 9-13. Format of Year Count Register (YEAR)

Address: F898H 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.

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

Reset signal generation clears this register to 00H.

Figure 9-14. Format of Watch Error Correction Register (SUBCUD)

Address: FF4DH 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).

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 9-15. Format of Alarm Minute Register (ALARMWM)

Address: FF59H 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 9-16. Format of Alarm Hour Register (ALARMWH)

Address: FF5AH 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 9-17. Format of Alarm Week Register (ALARMWW)

Address: FF5BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

(17) Port mode register 4 (PM4)

This register sets port 4 input/output in 1-bit units.

When using the P40/RTC1HZ/CECOUT and P41/RTCDIV/RTCCL/CECIN pins for clock output of real-time counter, clear PM40 and PM41 and the output latches of P40 and P41 to 0.

PM4 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 9-18. Format of Port Mode Register 4 (PM4)

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	PM43 Note	PM42 Note	PM41	PM40

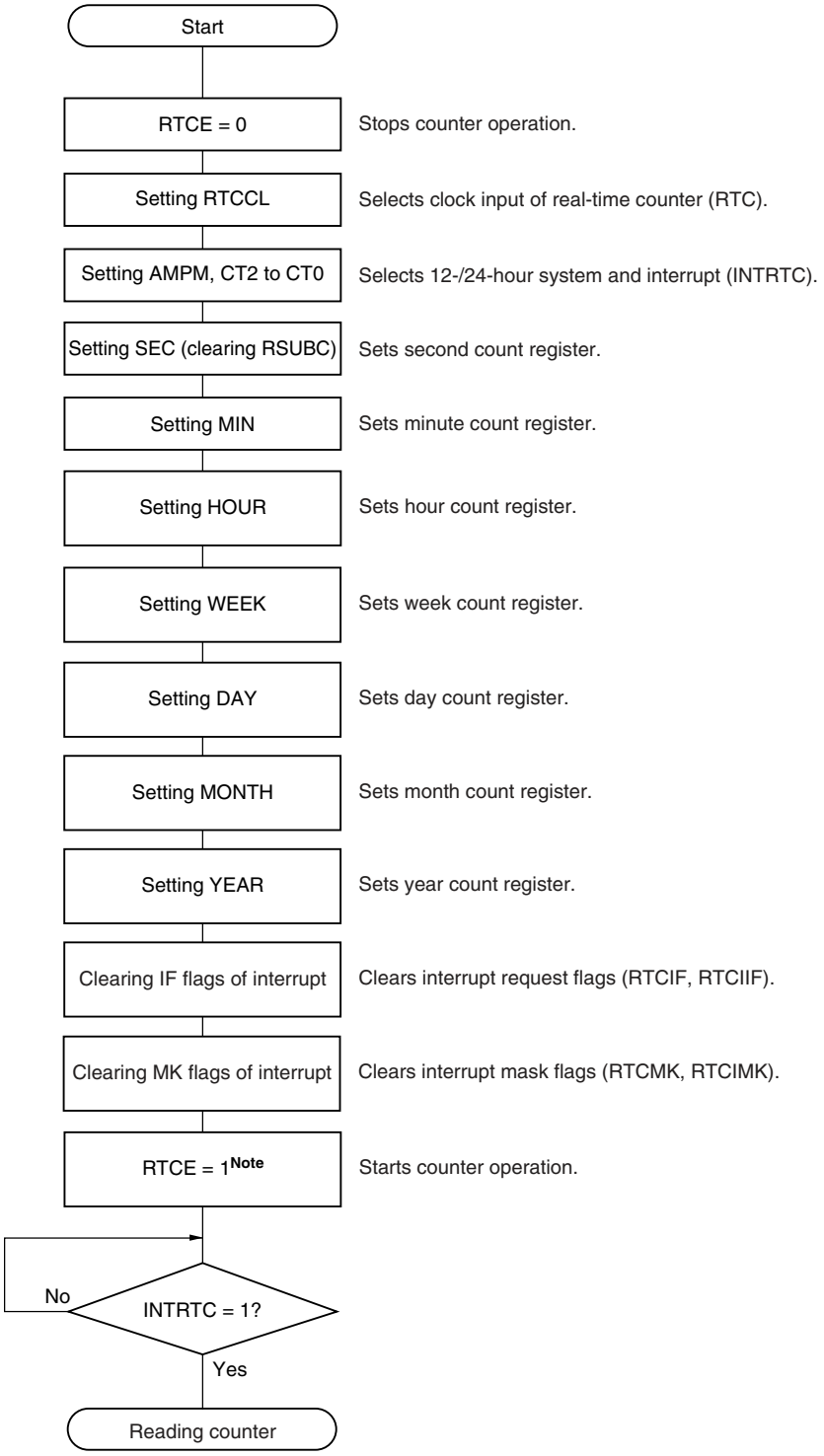
PM4n	P4n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 78K0/KE2-C only

9.4 Real-Time Counter Operation

9.4.1 Starting operation of real-time counter

Figure 9-19. Procedure for Starting Operation of Real-Time Counter



Note Confirm the procedure described in **9.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

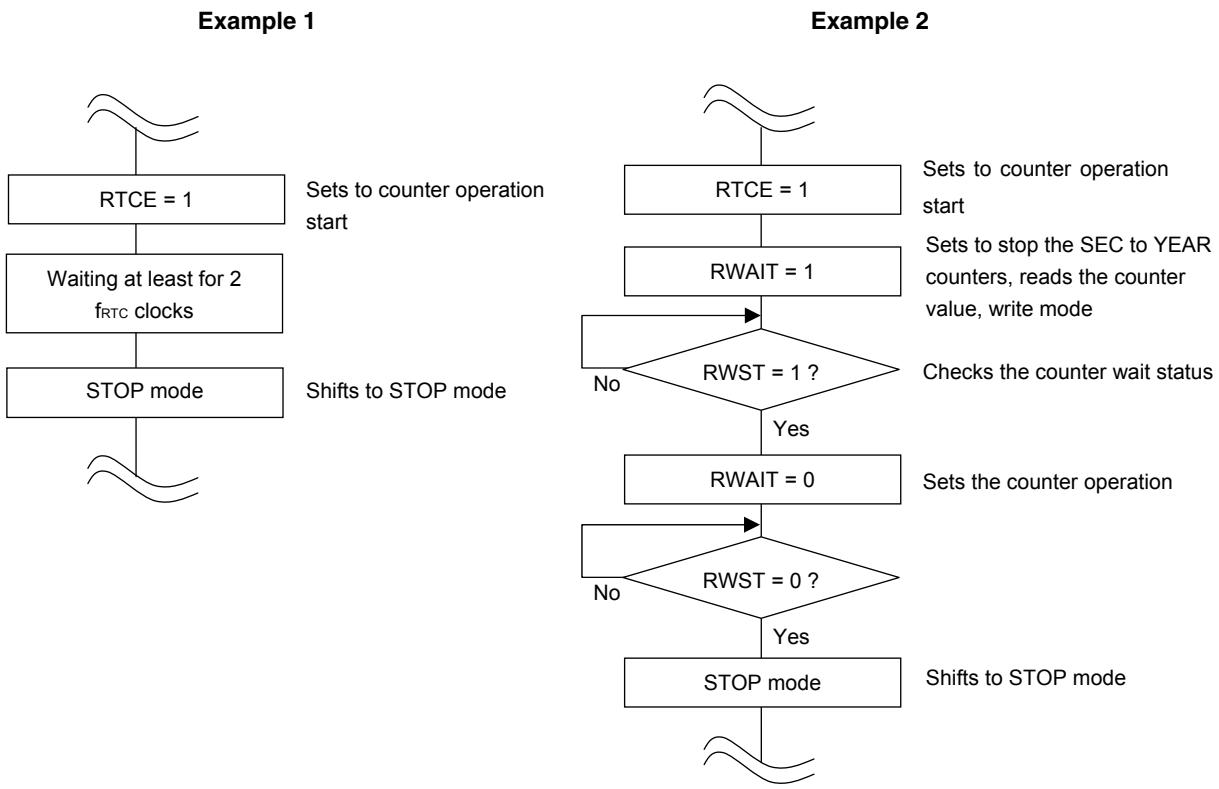
9.4.2 Shifting to STOP mode after starting operation

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 input clocks (f_{RTC}) have elapsed after setting RTCE to 1 (see **Figure 9-20**, **Example 1**).
- Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 9-20**, **Example 2**).

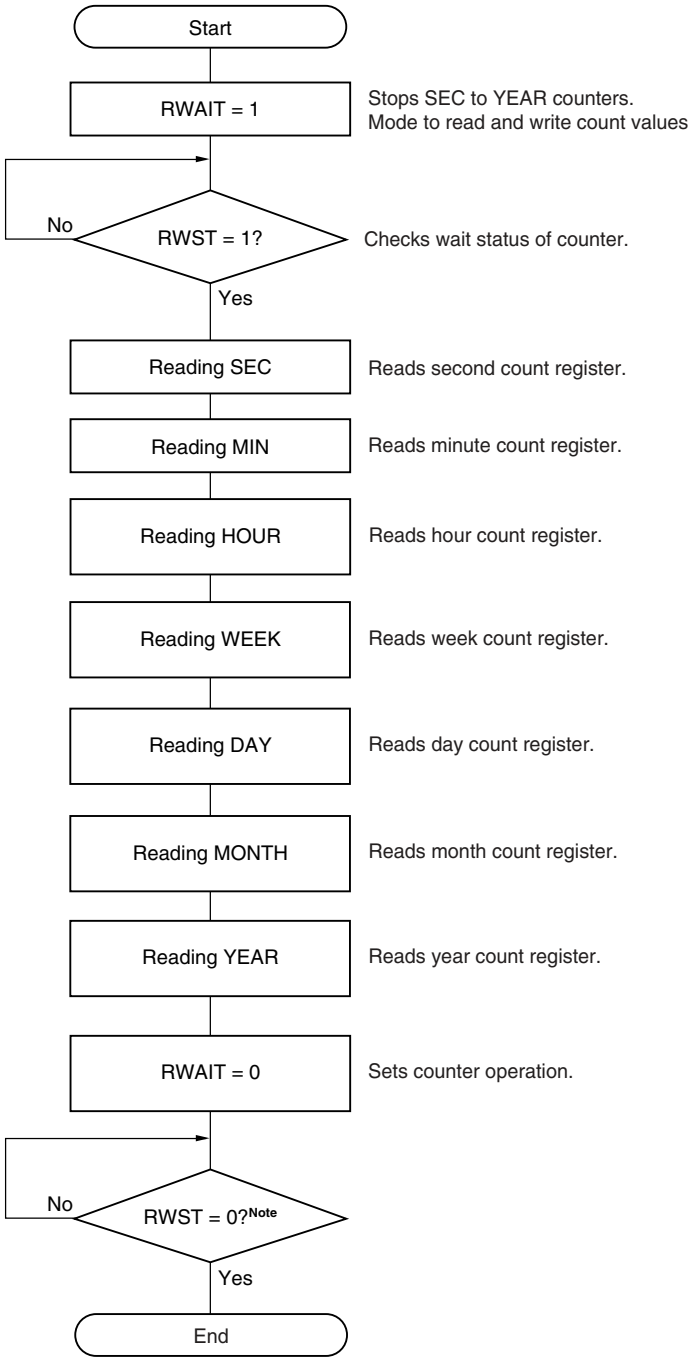
Figure 9-20. Procedure for Shifting to STOP Mode After Setting RTCE to 1



9.4.3 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 9-21. Procedure for Reading Real-Time Counter

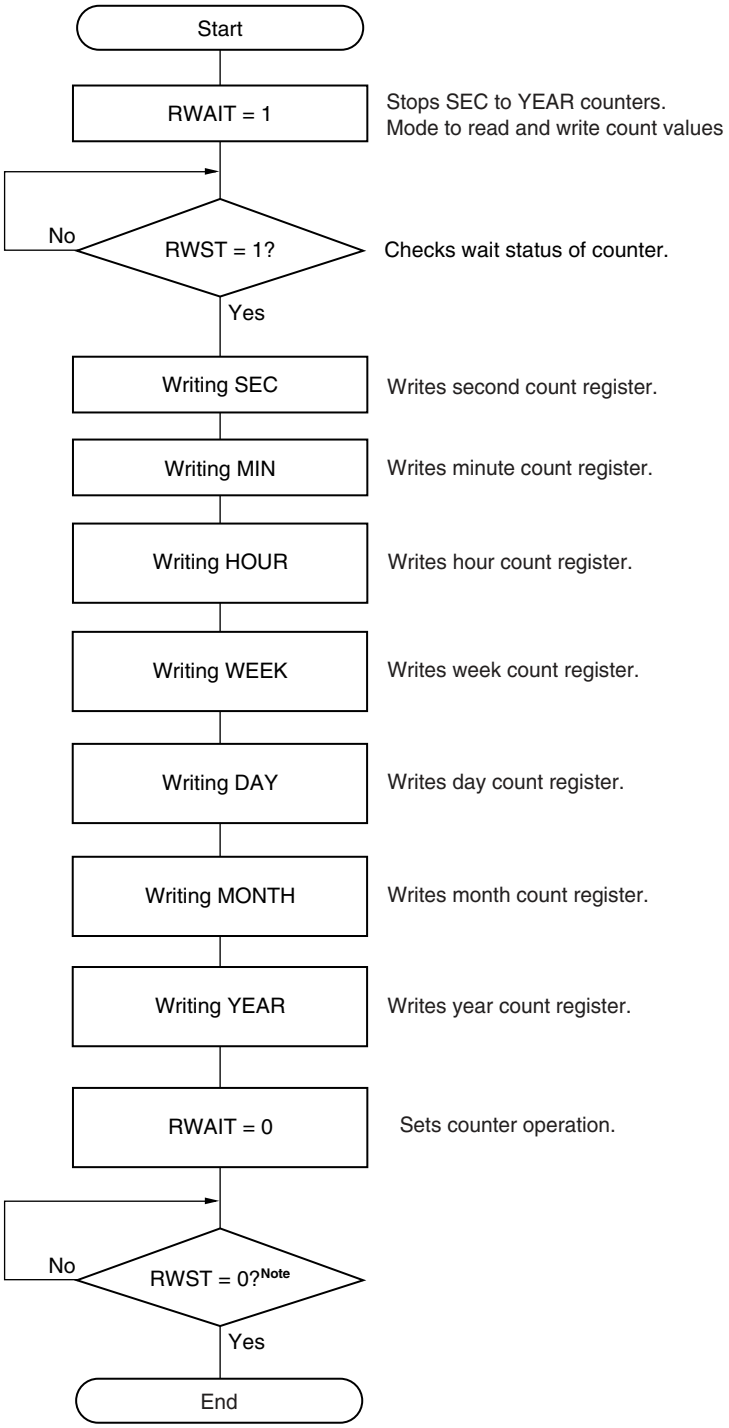


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence.
All the registers do not have to be set and only some registers may be read.

Figure 9-22. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

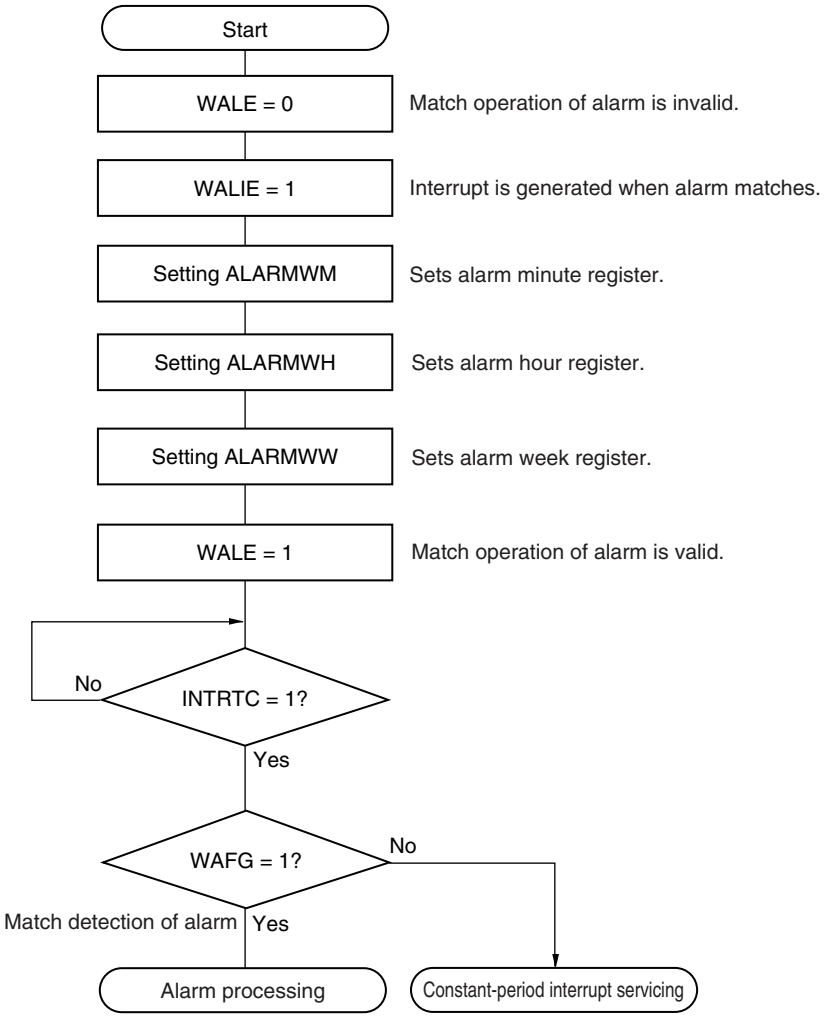
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence.
All the registers do not have to be set and only some registers may be written.

9.4.4 Setting alarm of real-time counter

Set time of alarm after setting 0 to WALE first.

Figure 9-23. Alarm Setting Procedure



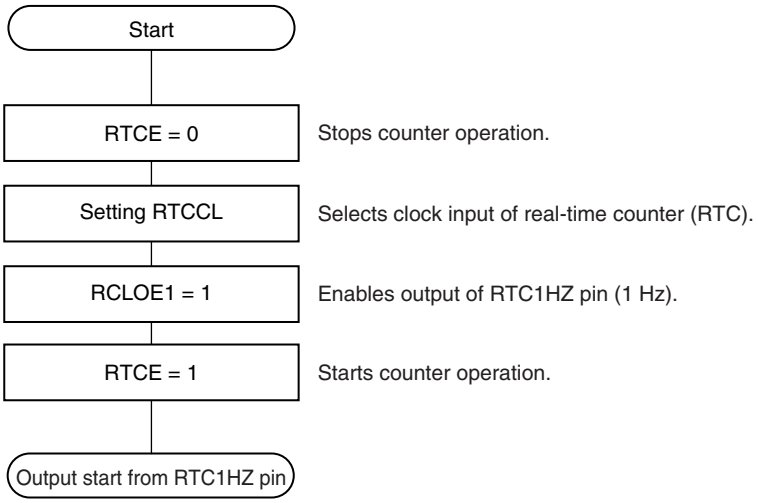
Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

9.4.5 1 Hz output of real-time counter

Set 1 Hz output after setting 0 to RTCE first.

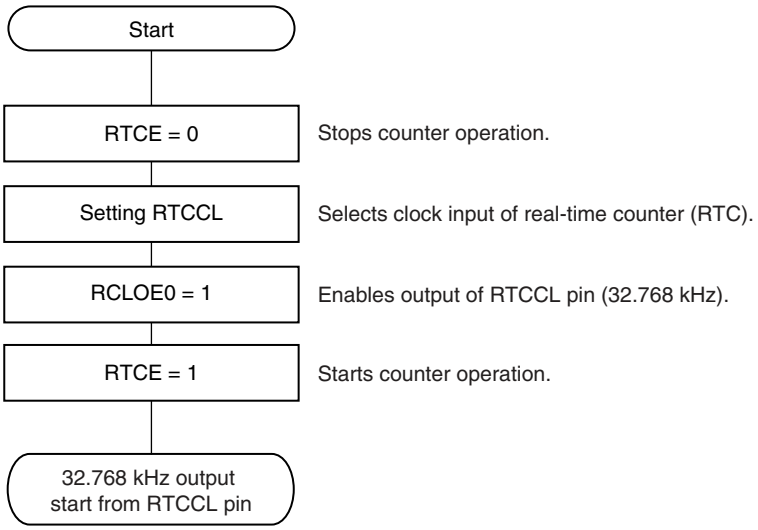
Figure 9-24. 1 Hz Output Setting Procedure



9.4.6 32.768 kHz output of real-time counter

Set 32.768 kHz output after setting 0 to RTCE first.

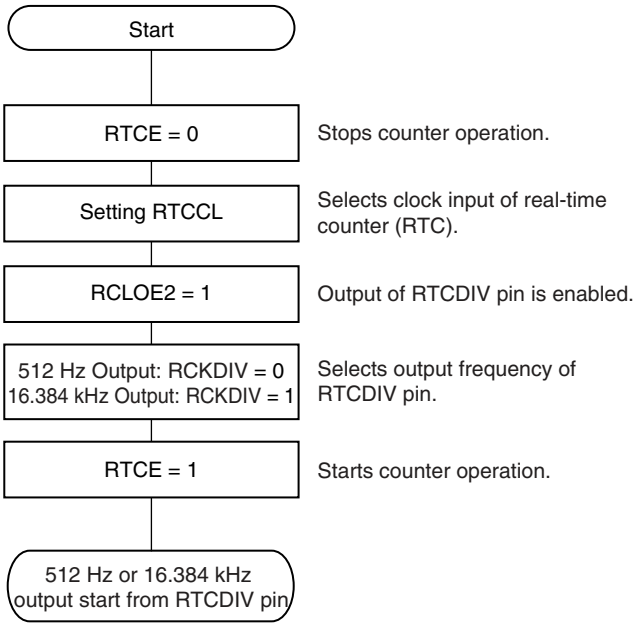
Figure 9-25. 32.768 kHz Output Setting Procedure



9.4.7 512 Hz, 16.384 kHz output of real-time counter

Set 512 Hz or 16.384 kHz output after setting 0 to RTCE first.

Figure 9-26. 512 Hz, 16.384 kHz output Setting Procedure



9.4.8 Example of watch error correction of real-time counter

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the sub-count register (RSUBC) is calculated by using the following expression.

Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1. /F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the input clock (f_{RTC}) value of the real-time counter (RTC).
It can be calculated from the 32 kHz output frequency of the RTCCL pin or the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example <1>

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **9.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **9.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 32772.3 Hz)

If the target frequency is assumed to be 32768 Hz (32772.3 Hz – 131.2 ppm), the correction range for –131.2 ppm is –63.1 ppm or less, so assume DEV to be 0.

The expression for calculating the correction value when DEV is 0 is applied.

Correction value = Number of correction counts in 1 minute ÷ 3
= (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 – 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or more (when delaying), assume F6 to be 0.

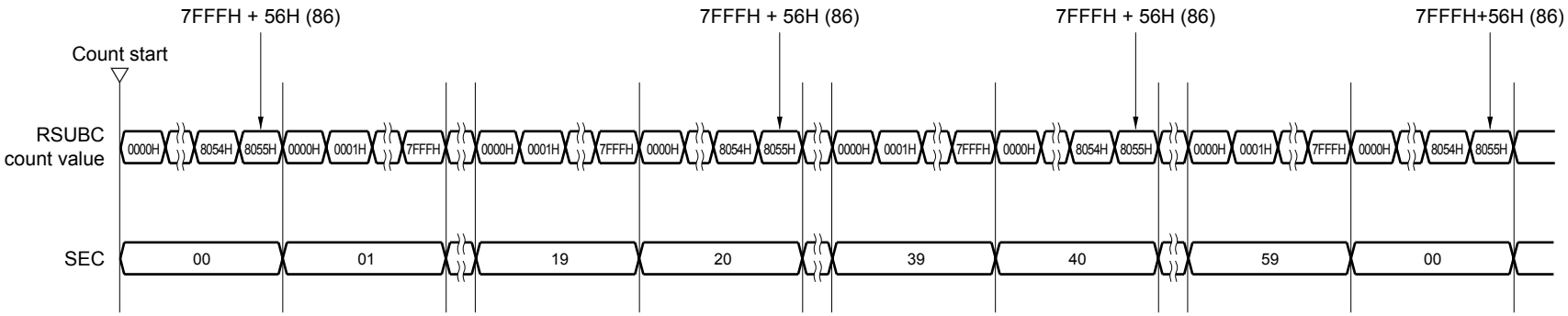
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{ (F5, F4, F3, F2, F1, F0) – 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of SUBCUD: 0101100) results in 32768 Hz (0 ppm).

Figure 9-27 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 9-27. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example <2>

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **9.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **9.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60
= (32767.4 ÷ 32768 − 1) × 32768 × 60
= −36

[Calculating the values to be set to (F6 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when speeding up), assume F6 to be 1.

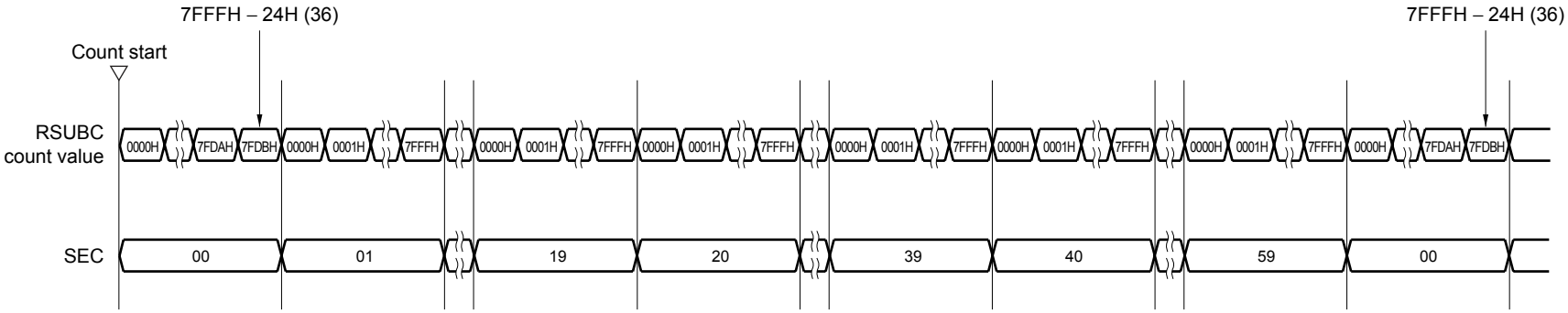
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

− {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2	= −36
(/F5, /F4, /F3, /F2, /F1, /F0)	= 17
(/F5, /F4, /F3, /F2, /F1, /F0)	= (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0)	= (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 6 to 0 of SUBCUD: 1101110) results in 32768 Hz (0 ppm).

Figure 9-28 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 9-28. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 10 WATCHDOG TIMER

10.1 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
- If the instruction is fetched from an area not set by the IMS and IXS registers[Note] (detection of an invalid check while the CPU hangs up)
- If the CPU accesses an area that is not set by the IMS and IXS registers[Note] (excluding FB00H to FFCFH and FFE0H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

Note A product that does not have an internal expansion RAM is not provided with the IXS register.

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

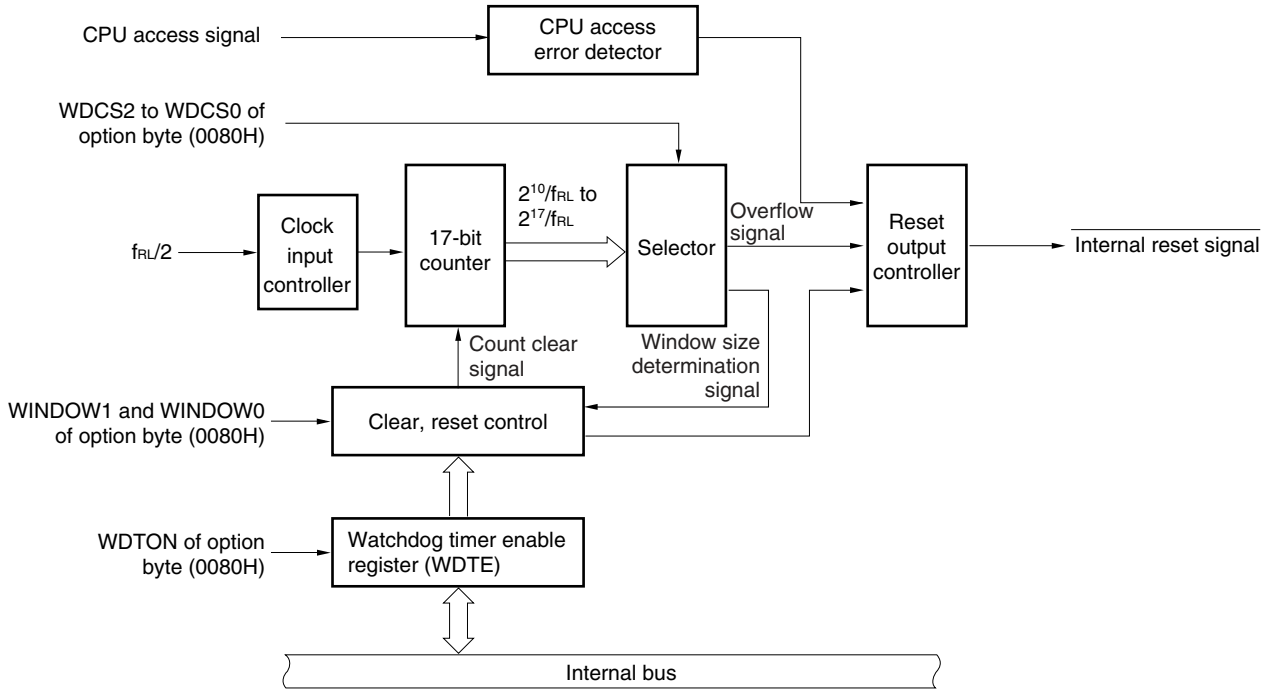
How the counter operation is controlled, overflow time, and window open period are set by the option byte.

Table 10-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (0080H)
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)

Remark For the option byte, see **CHAPTER 26 OPTION BYTE**.

Figure 10-1. Block Diagram of Watchdog Timer



10.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(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 10-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FF99H 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 (0080H). 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. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

10.4 Operation of Watchdog Timer

10.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (0080H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (0080H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 26**).

WDTON	Operation Control of Watchdog Timer Counter/Illegal Access Detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H) (for details, see **10.4.2** and **CHAPTER 26**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, see **10.4.3** and **CHAPTER 26**).

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.
   A 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
   - If the instruction is fetched from an area not set by the IMS and IXS registers[Note] (detection of an invalid check during a CPU program loop)
   - If the CPU accesses an area not set by the IMS and IXS registers[Note] (excluding FB00H to FFCFH and FFE0H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

Note A product that does not have an internal expansion RAM is not provided with the IXS register.

Cautions 1. The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, 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_{RL} seconds.

3. The watchdog timer can be cleared immediately before the count value overflows (FFFFH).

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 (LSROSC) of the option byte.

	LSROSC = 0 (Internal Low-Speed Oscillator Can Be Stopped by Software)	LSROSC = 1 (Internal Low-Speed Oscillator Cannot Be Stopped)
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If LSROSC = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is not cleared to 0 but starts counting from the value at which it was stopped.

If oscillation of the internal low-speed oscillator is stopped by setting LSRSTOP (bit 1 of the internal oscillation mode register (RCM) = 1) when LSROSC = 0, the watchdog timer stops operating. At this time, the counter is not cleared to 0.

5. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

10.4.2 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 (0080H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to WDTE during the window open period before the overflow time.

The following overflow time is set.

Table 10-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

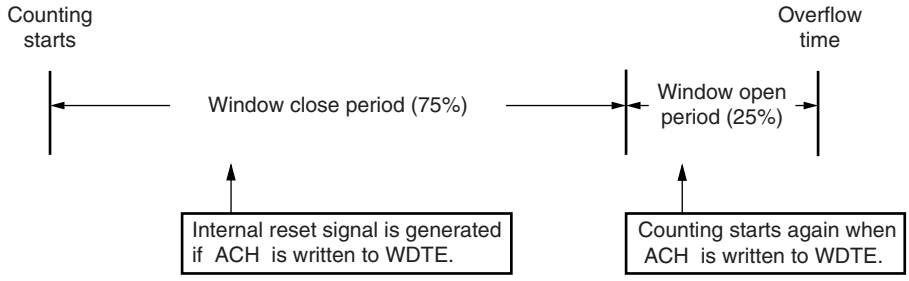
2. ( ): f_{RL} = 264 kHz (MAX.)

10.4.3 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 (0080H). The outline of the window is as follows.

- If “ACH” is written to WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to WDTE during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 25%



Caution The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, and the watchdog timer starts counting again.

The window open period to be set is as follows.

Table 10-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remark If the overflow time is set to $2^{11}/f_{RL}$, the window close time and open time are as follows.

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 7.11 ms	0 to 4.74 ms	0 to 2.37 ms	None
Window open time	7.11 to 7.76 ms	4.74 to 7.76 ms	2.37 to 7.76 ms	0 to 7.76 ms

<When window open period is 25%>

- Overflow time:
  $2^{11}/f_{RL}$ (MAX.) = 2^{11}/264 kHz (MAX.) = 7.76 ms
- Window close time:
  0 to $2^{11}/f_{RL}$ (MIN.) × (1 − 0.25) = 0 to 2^{11}/216 kHz (MIN.) × 0.75 = 0 to 7.11 ms
- Window open time:
  $2^{11}/f_{RL}$ (MIN.) × (1 − 0.25) to $2^{11}/f_{RL}$ (MAX.) = 2^{11}/216 kHz (MIN.) × 0.75 to 2^{11}/264 kHz (MAX.) = 7.11 to 7.76 ms

CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
Clock output	√	
Buzzer output	–	√

Caution If clock output (PCL) is used, remote controller receiver and INTP6 cannot be used.

Remark √: Mounted, –: Not mounted

11.1 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. The clock selected with the clock output selection register (CKS) is output.

In addition, the buzzer output is intended for square-wave output of buzzer frequency selected with CKS.

Figure 11-1 and 11-2 show the block diagram of clock output/buzzer output controller.

Figure 11-1. Block Diagram of Clock Output/Buzzer Output Controller (78K0/KC2-C)

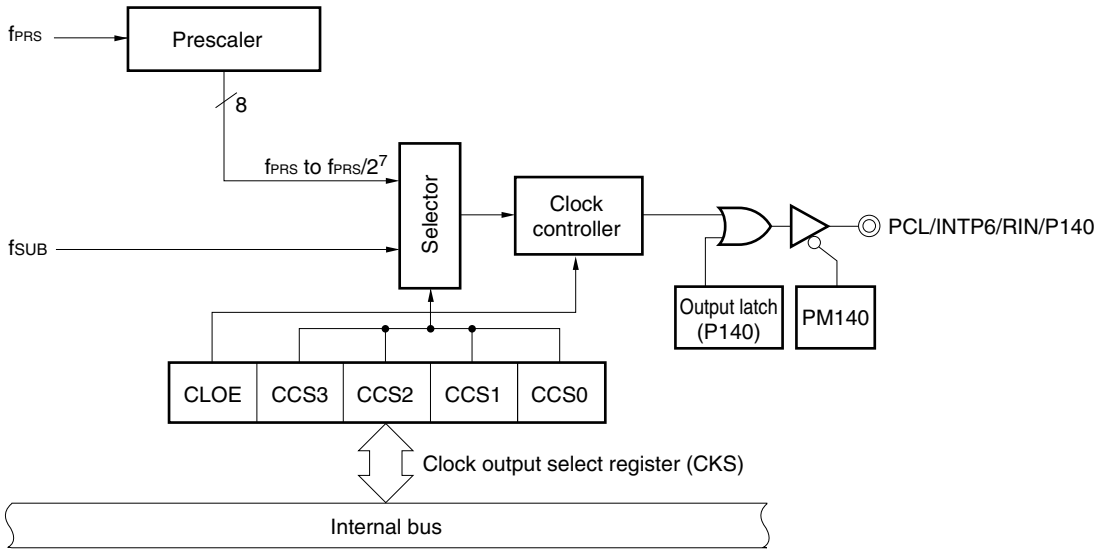
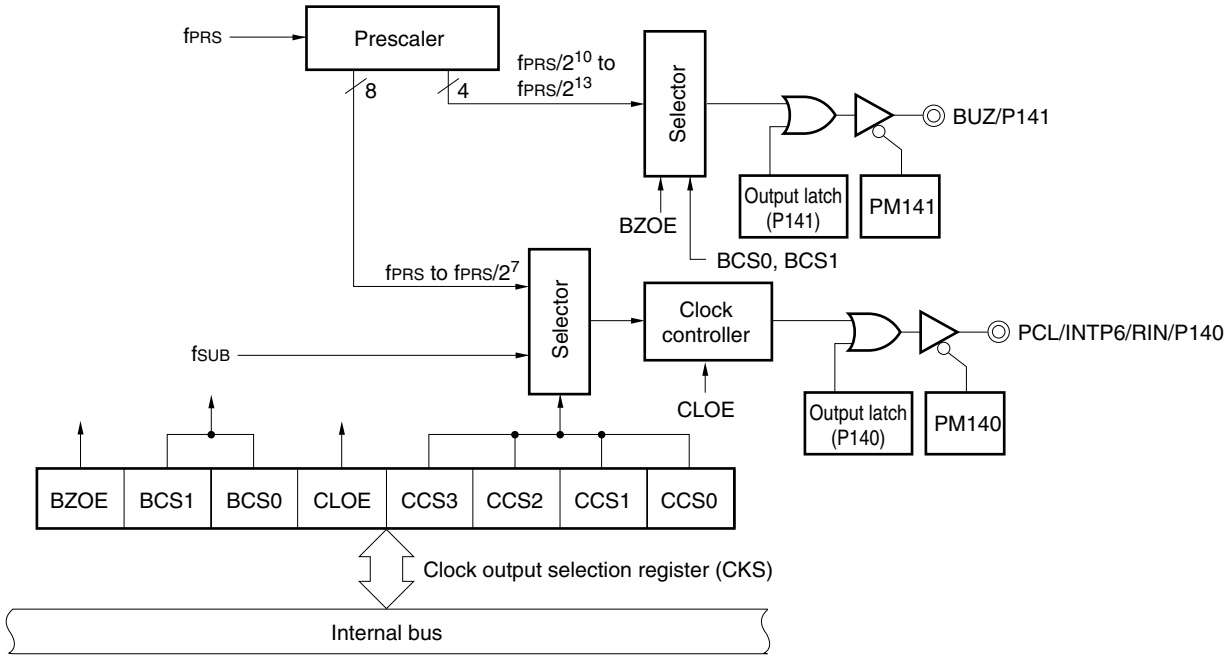


Figure 11-2. Block Diagram of Clock Output/Buzzer Output Controller (78K0/KE2-C)



11.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 11-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 14 (PM14) Port register 14 (P14)

11.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output selection register (CKS)
- Port mode register 14 (PM14)

(1) Clock output selection register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

CKS is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CKS to 00H.

Figure 11-3. Format of Clock Output Selection Register (CKS) (78K0/KC2-C)

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	0
CKS	0	0	0	CLOE	CCS3	CCS2	CCS1	CCS0

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection			
					f_{SUB} = 32.768 kHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	–	10 MHz	Setting prohibited[Note]
0	0	0	1	$f_{PRS}/2$		5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$		2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$		1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$		625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$		312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$		156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$		78.125 kHz	156.25 kHz
1	0	0	0	f_{SUB}	32.768 kHz	–	
Other than above				Setting prohibited			

Note The PCL output clock prohibits settings if they exceed 10 MHz.

Caution Set CCS3 to CCS0 while the clock output operation is stopped (CLOE = 0).

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{SUB}: Subsystem clock frequency

Figure 11-4. Format of Clock Output Selection Register (CKS) (78K0/KE2-C)

Address: FF40H After reset: 00H R/W

Symbol	<7>	6	5	<4>	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Clock division circuit operation stopped. BUZ fixed to low level.
1	Clock division circuit operation enabled. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection		
			f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	$f_{PRS}/2^{10}$	9.77 kHz	19.54 kHz
0	1	$f_{PRS}/2^{11}$	4.88 kHz	9.77 kHz
1	0	$f_{PRS}/2^{12}$	2.44 kHz	4.88 kHz
1	1	$f_{PRS}/2^{13}$	1.22 kHz	2.44 kHz

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection			
					f_{SUB} = 32.768 kHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	–	10 MHz	Setting prohibited[Note]
0	0	0	1	$f_{PRS}/2$		5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$		2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$		1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$		625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$		312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$		156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$		78.125 kHz	156.25 kHz
1	0	0	0	f_{SUB}	32.768 kHz	–	
Other than above				Setting prohibited			

Note The PCL output clock prohibits settings if they exceed 10 MHz.

Cautions 1. Set BCS1 and BCS0 when the buzzer output operation is stopped (BZOE = 0).

2. Set CCS3 to CCS0 while the clock output operation is stopped (CLOE = 0).

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{SUB}: Subsystem clock frequency

(2) Port mode register 14 (PM14)

This register sets port 14 input/output in 1-bit units.

When using the P140/RIN/INTP6/PCL pin for clock output and the P141/BUZ pin for buzzer output, clear PM140 and PM141 and the output latches of P140 and P141 to 0.

PM14 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM14 to FFH.

Figure 11-5. Format of Port Mode Register 14 (PM14)

Address: FF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	PM141[Note]	PM140

PM14n	P14n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 78K0/KE2-C only

11.4 Operations of Clock Output/Buzzer Output Controller

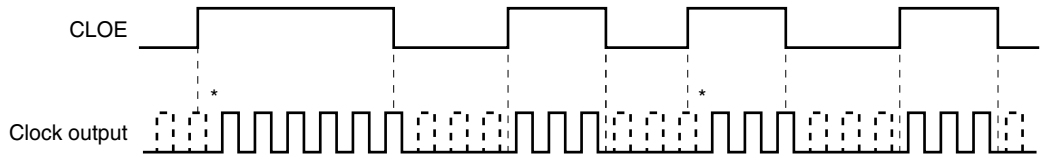
11.4.1 Operation as clock output

The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).
<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 11-6, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after the high-level period of the clock.

Figure 11-6. Remote Control Output Application Example



11.4.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output selection register (CKS) (buzzer output in disabled status).
<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

CHAPTER 12 A/D CONVERTER

12.1 Function of A/D Converter

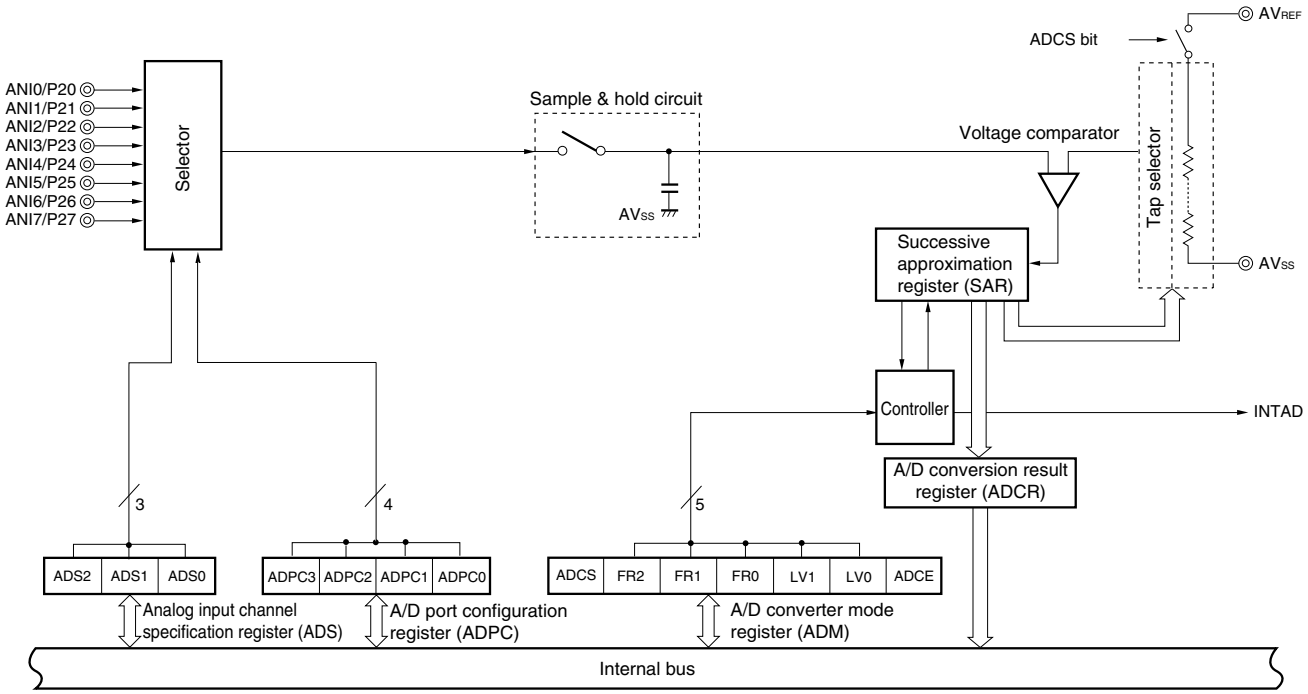
The A/D converter converts an analog input signal into a digital value, and consists of up to eight channels (ANI0 to ANI7) with a resolution of 10 bits.

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 to ANI7. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 12-1. Block Diagram of A/D Converter



12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI7 pins

These are the analog input pins of the 8-channel 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.

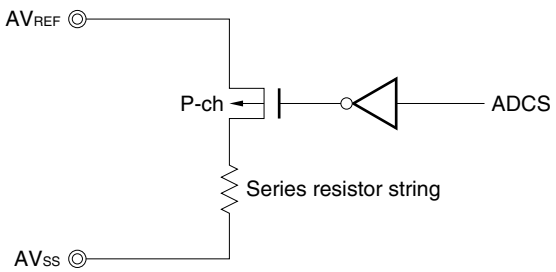
(2) Sample & hold circuit

The sample & hold circuit samples the input voltage of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled voltage value during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared with the sampled voltage value.

Figure 12-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled voltage value and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register converts the result of comparison by the voltage comparator, starting from the most significant bit (MSB). When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) 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.

Caution When data is read from ADCR and ADCRH, a wait cycle is generated. Do not read data from ADCR and ADCRH when the peripheral hardware clock (f_{PRS}) is stopped.

(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 an analog power/reference voltage to the A/D converter. Make this pin the same potential as the V_{DD} pin when port 2 is used as a digital port.

The signal input to ANI0 to ANI7 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.

(11) A/D converter mode register (ADM)

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

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

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

(13) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

(14) Port mode register 2 (PM2)

This register switches the ANI0/P20 to ANI7/P27 pins to input or output.

12.3 Registers Used in A/D Converter

The A/D converter uses the following six registers.

- A/D converter mode register (ADM)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- Port mode register 2 (PM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

(1) 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 12-3. Format of A/D Converter Mode Register (ADM)

Address: FF28H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	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

ADCE	Comparator operation control Note 2
0	Stops comparator operation
1	Enables comparator operation

Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 12-2 A/D Conversion Time Selection**.

2. The operation of the comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

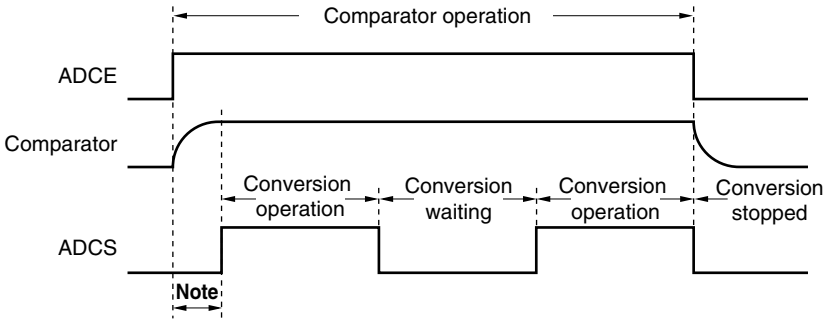
Caution Be sure to clear bit 6 to "0".

Table 12-1. 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 (comparator operation, only comparator consumes power)
1	0	Conversion mode (comparator operation stopped Note)
1	1	Conversion mode (comparator operation)

Note Ignore the first conversion data.

Figure 12-4. Timing Chart When Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR2, LV1, and LV0 to values other than the identical data.

2. If data is written to ADM, a wait cycle is generated. Do not write data to ADM when the peripheral hardware clock (f_{PRS}) is stopped.

Table 12-2. A/D Conversion Time Selection (1/2)

(1) 2.7 V ≤ AV_{REF} ≤ 5.5 V (LV0 = 0)

A/D Converter Mode Register (ADM)					Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	0	0	264/f_{PRS}	Setting prohibited	52.8 μs	26.4 μs	13.2 μs	f_{PRS}/12
0	0	1	0	0	176/f_{PRS}		35.2 μs	17.6 μs	8.8 μs[Note]	f_{PRS}/8
0	1	0	0	0	132/f_{PRS}	66.0 μs	26.4 μs	13.2 μs	6.6 μs[Note]	f_{PRS}/6
0	1	1	0	0	88/f_{PRS}	44.0 μs	17.6 μs	8.8 μs[Note]	Setting prohibited	f_{PRS}/4
1	0	0	0	0	66/f_{PRS}	33.0 μs	13.2 μs	6.6 μs[Note]		f_{PRS}/3
1	0	1	0	0	44/f_{PRS}	22.0 μs	8.8 μs[Note]	Setting prohibited		f_{PRS}/2
Other than above					Setting prohibited					

Note This can be set only when 4.0 V ≤ AV_{REF} ≤ 5.5 V.

Table 12-2. A/D Conversion Time Selection (2/2)

(2) 2.3 V ≤ AV_{REF} ≤ 5.5 V (LV0 = 1)

A/D Converter Mode Register (ADM)					Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	0	1	480/f_{PRS}	Setting prohibited	Setting prohibited	48.0 μs[Note 2]	24.0 μs[Note 2]	f_{PRS}/12
0	0	1	0	1	320/f_{PRS}		64.0 μs	32.0 μs[Note 2]	16.0 μs[Note 1]	f_{PRS}/8
0	1	0	0	1	240/f_{PRS}		48.0 μs	24.0 μs[Note 2]	12.0 μs[Note 1]	f_{PRS}/6
0	1	1	0	1	160/f_{PRS}		32.0 μs	16.0 μs[Note 1]	Setting prohibited	f_{PRS}/4
1	0	0	0	1	120/f_{PRS}	60.0 μs	24.0 μs[Note 2]	12.0 μs[Note 1]		f_{PRS}/3
1	0	1	0	1	80/f_{PRS}	40.0 μs	16.0 μs[Note 1]	Setting prohibited		f_{PRS}/2
Other than above					Setting prohibited					

Notes 1. This can be set only when 4.0 V ≤ AV_{REF} ≤ 5.5 V.

2. This can be set only when 2.7 V ≤ AV_{REF} ≤ 5.5 V.

Cautions 1. Set the conversion times with the following conditions.

(1) 2.7 V ≤ AV_{REF} ≤ 5.5 V (LV0 = 0)

- **4.0 V ≤ AV_{REF} ≤ 5.5 V: f_{AD} = 0.33 to 3.6 MHz**
- **2.7 V ≤ AV_{REF} < 4.0 V: f_{AD} = 0.33 to 1.8 MHz**

(2) 2.3 V ≤ AV_{REF} ≤ 5.5 V (LV0 = 1)

- **4.0 V ≤ AV_{REF} ≤ 5.5 V: f_{AD} = 0.6 to 3.6 MHz**
- **2.7 V ≤ AV_{REF} < 4.0 V: f_{AD} = 0.6 to 1.8 MHz**
- **2.3 V ≤ AV_{REF} < 2.7 V: f_{AD} = 0.6 to 1.48 MHz**

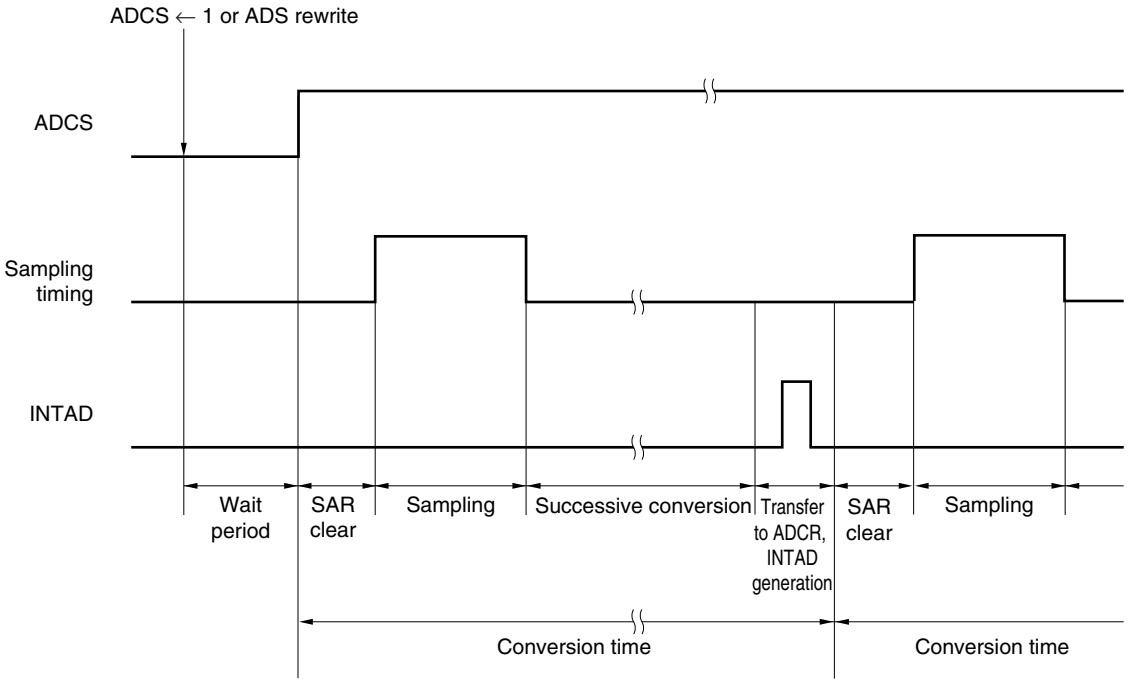
2. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

3. Change LV0 from the default value, when 2.3 V ≤ AV_{REF} < 2.7 V.

4. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 12-5. A/D Converter Sampling and A/D Conversion Timing



(2) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 8 bits of the conversion result are stored in FF09H and the lower 2 bits are stored in the higher 2 bits of FF08H.

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

Reset signal generation clears this register to 0000H.

Figure 12-6. Format of 10-Bit A/D Conversion Result Register (ADCR)

Address: FF08H, FF09H After reset: 0000H R

Symbol	FF09H								FF08H							
ADCR											0	0	0	0	0	0

Cautions 1. When writing to the 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.

2. If data is read from ADCR, a wait cycle is generated. Do not read data from ADCR when the peripheral hardware clock (f_{PRS}) is stopped.

(3) 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 12-7. Format of 8-Bit A/D Conversion Result Register (ADCRH)

Address: FF09H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Cautions 1. When writing to the 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.

2. If data is read from ADCRH, a wait cycle is generated. Do not read data from ADCRH when the peripheral hardware clock (f_{PRS}) is stopped.

(4) 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 12-8. Format of Analog Input Channel Specification Register (ADS)

Address: FF29H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

Cautions 1. Be sure to clear bits 3 to 7 to "0".

2 Set a channel to be used for A/D conversion in the input mode by using port mode register 2 (PM2).

3. If data is written to ADS, a wait cycle is generated. Do not write data to ADS when the peripheral hardware clock (f_{PRS}) is stopped.

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

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

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

Reset signal generation clears this register to 00H.

Figure 12-9. Format of A/D Port Configuration Register (ADPC)

Address: FF2FH After reset: 00H R/W

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

ADPC3	ADPC2	ADPC1	ADPC0	Digital I/O (D)/analog input (A) switching							
				P27/ ANI7	P26/ ANI6	P25/ ANI5	P24/ ANI4	P23/ ANI3	P22/ ANI2	P21/ ANI1	P20/ ANI0
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	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. If data is written to ADPC, a wait cycle is generated. Do not write data to ADPC when the peripheral hardware clock (f_{PRS}) is stopped.

(6) Port mode register 2 (PM2)

When using the ANI0/P20 to ANI7/P27 pins for analog input port, set PM20 to PM27 to 1. The output latches of P20 to P27 at this time may be 0 or 1.

If PM20 to PM27 are set to 0, they cannot be used as analog input port pins.

PM2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 12-10. Format of Port Mode Register 2 (PM2)

Address: FF22H 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 = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

ANI0/P20 to ANI7/P27 pins are as shown below depending on the settings of ADPC, ADS, and PM2.

Table 12-3. Setting Functions of ANI0/P20 to ANI7/P27 Pins

ADPC	PM2	ADS	ANI0/P20 to ANI7/P27 Pins
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.	
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output

12.4 A/D Converter Operations

12.4.1 Basic operations of A/D converter

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1 to start the operation of the comparator.

<2> Set 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 register 2 (PM2).

<3> Set A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM.

<4> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<5> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.
(<6> to <12> are operations performed by hardware.)

<6> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<7> 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.

<8> 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.

<9> 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.

<10> 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.

- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<11> Comparison is continued in this way up to bit 0 of SAR.

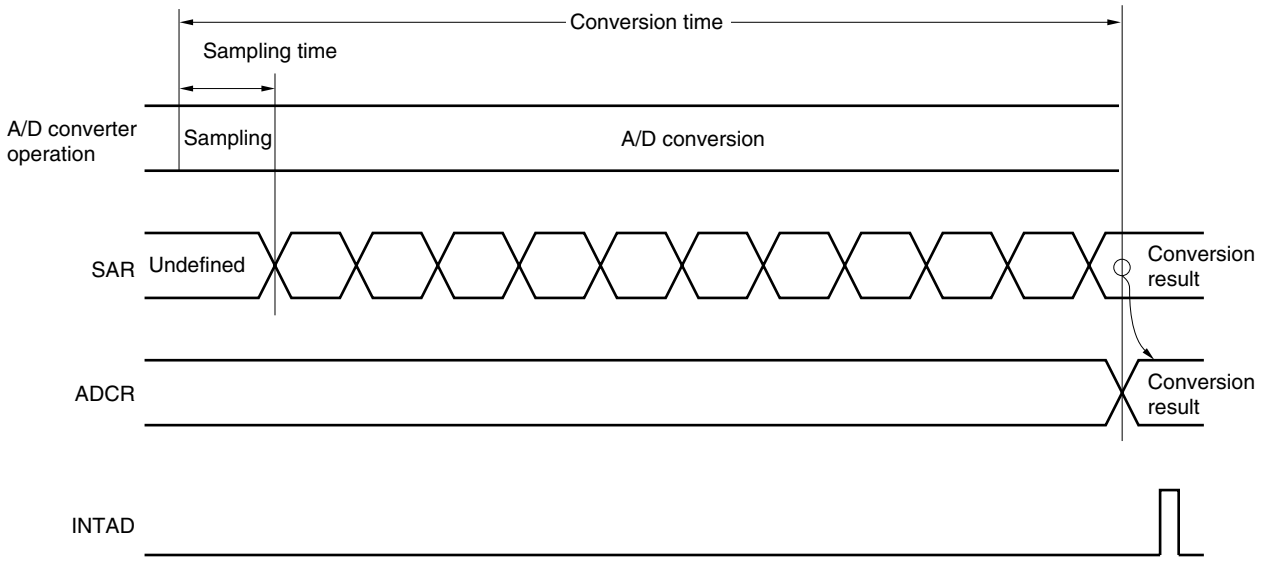
<12> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<13> Repeat steps <6> to <12>, 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 <5>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <5>. To change a channel of A/D conversion, start from <4>.

Caution Make sure the period of <1> to <5> 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

Figure 12-11. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the 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.

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) 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

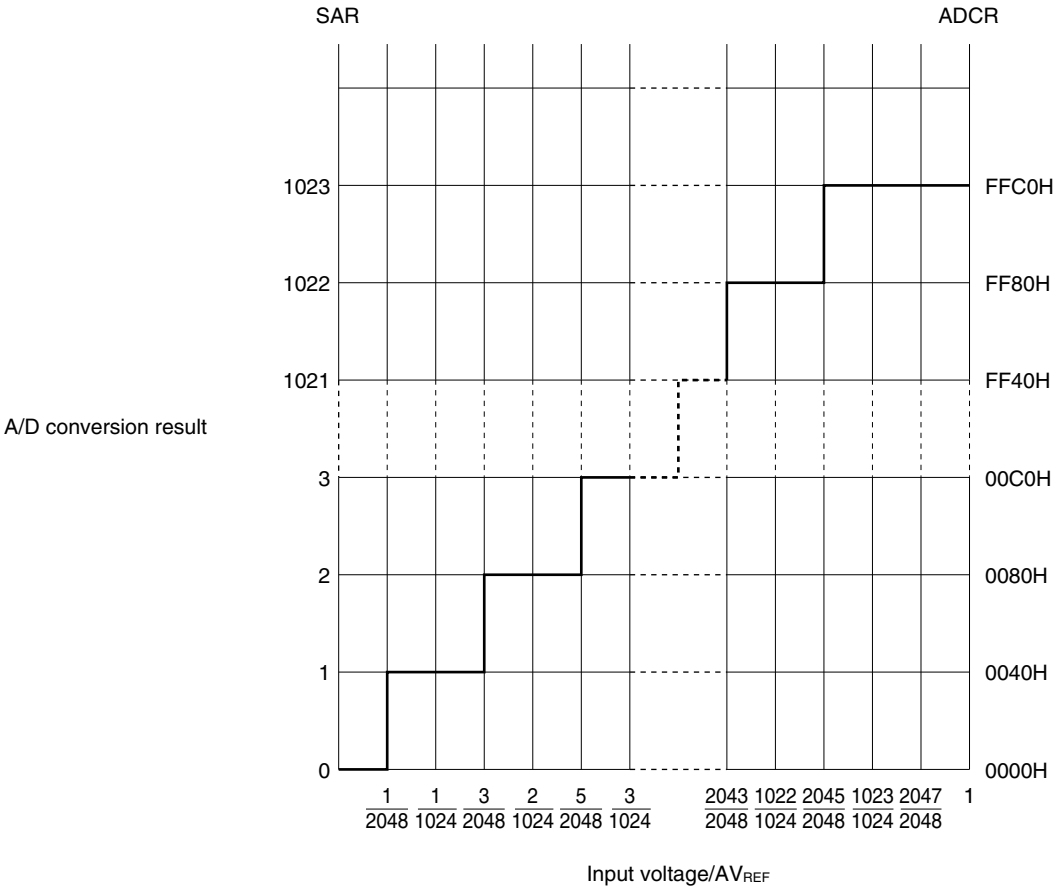
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-12. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI7 by the analog input channel specification register (ADS) and A/D conversion is executed.

(1) A/D conversion operation

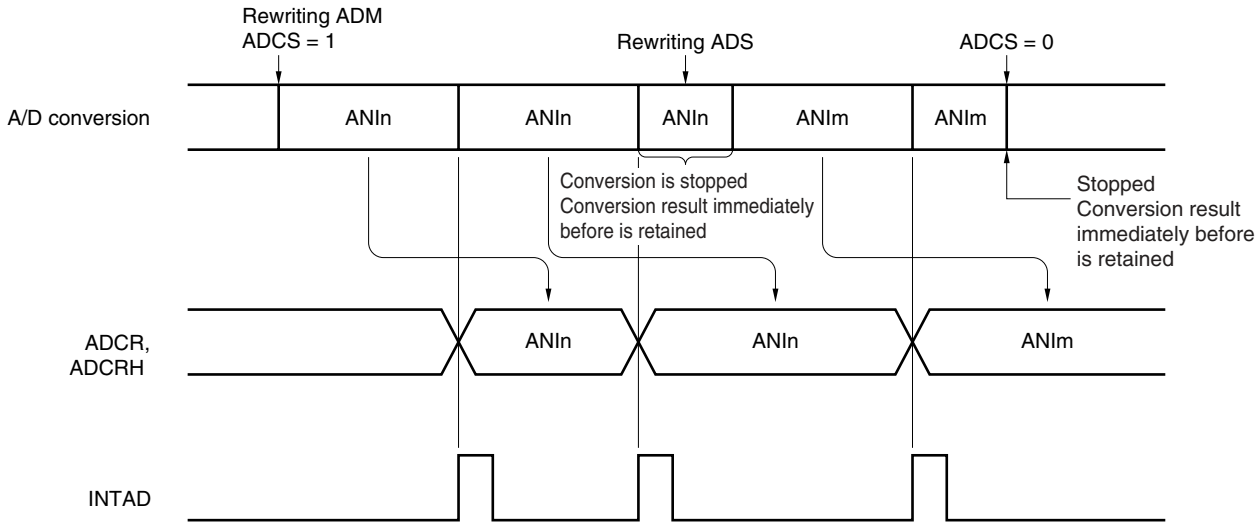
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 12-13. A/D Conversion Operation



Remarks 1. n = 0 to 7

2. m = 0 to 7

The setting methods are described below.

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
<2> Set the channel to be used in the analog input mode by using bits 3 to 0 (ADPC3 to ADPC0) of the A/D port configuration register (ADPC) and bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2).
<3> Select conversion time by using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM.
<4> Select a channel to be used by using bits 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS).
<5> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.
<6> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<7> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<8> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS to start A/D conversion.
<9> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<11> Clear ADCS to 0.
<12> Clear ADCE to 0.

Cautions 1. Make sure the period of <1> to <5> is 1 μs or more.
2. <1> may be done between <2> and <4>.
3. <1> can be omitted. However, ignore data of the first conversion after <5> in this case.
4. The period from <6> to <9> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <8> to <9> is the conversion time set using FR2 to FR0, LV1, and LV0.

12.5 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 12-14. Overall Error

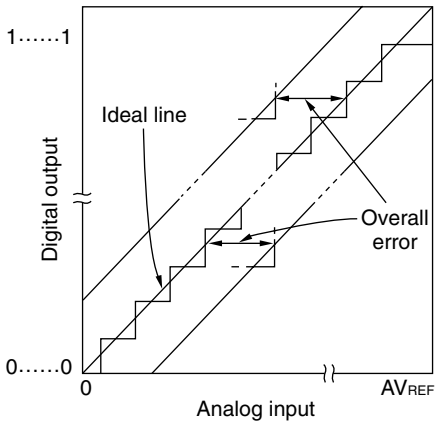
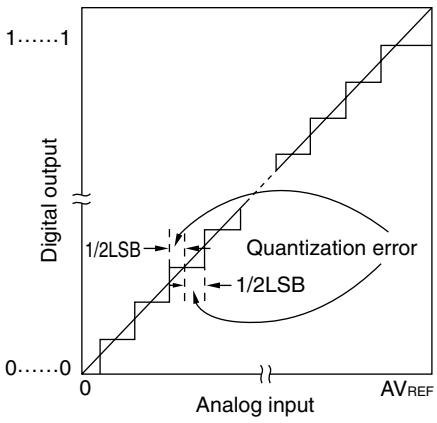


Figure 12-15. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

Ideally, the width to output a specific code is 1 LSB. This error indicates the difference between the actually measured value and its theoretical value when a specific code is output. This indicates the basic characteristics of the A/D conversion when the voltage applied to the analog input pins of the same channel is consistently increased bit by bit from AV_{SS} to AV_{REF}. When the input voltage is increased or decreased, or when two or more channels are used, see **12.5 (2) Overall error**.

Figure 12-16. Zero-Scale Error

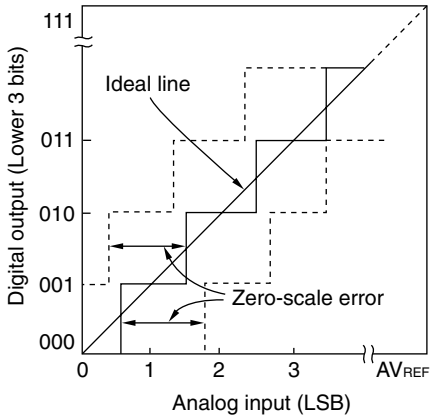


Figure 12-17. Full-Scale Error

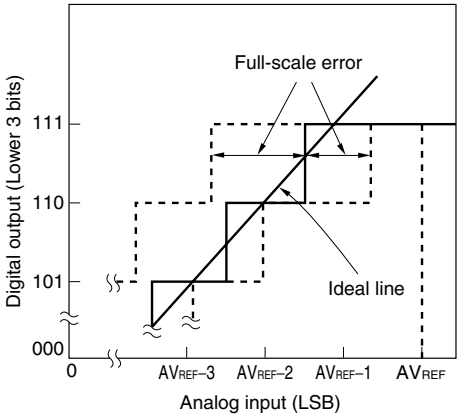


Figure 12-18. Integral Linearity Error

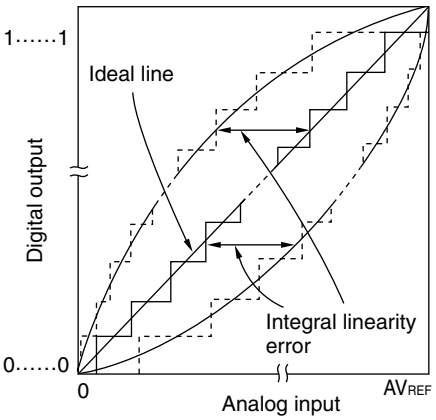
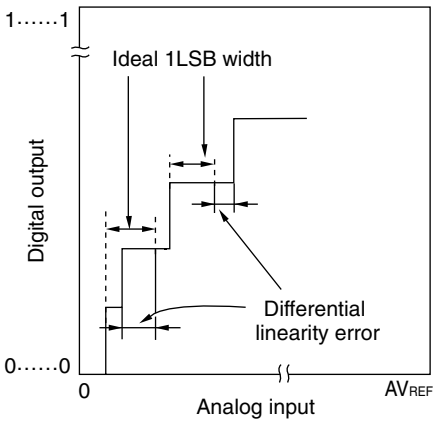


Figure 12-19. Differential Linearity Error

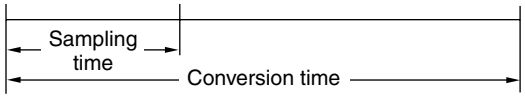


(8) Conversion time

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.



12.6 Cautions for A/D Converter

(1) Operating current in STOP mode

The A/D converter stops operating in the STOP mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of ANI0 to ANI7

Observe the rated range of the ANI0 to ANI7 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 to ANI7.

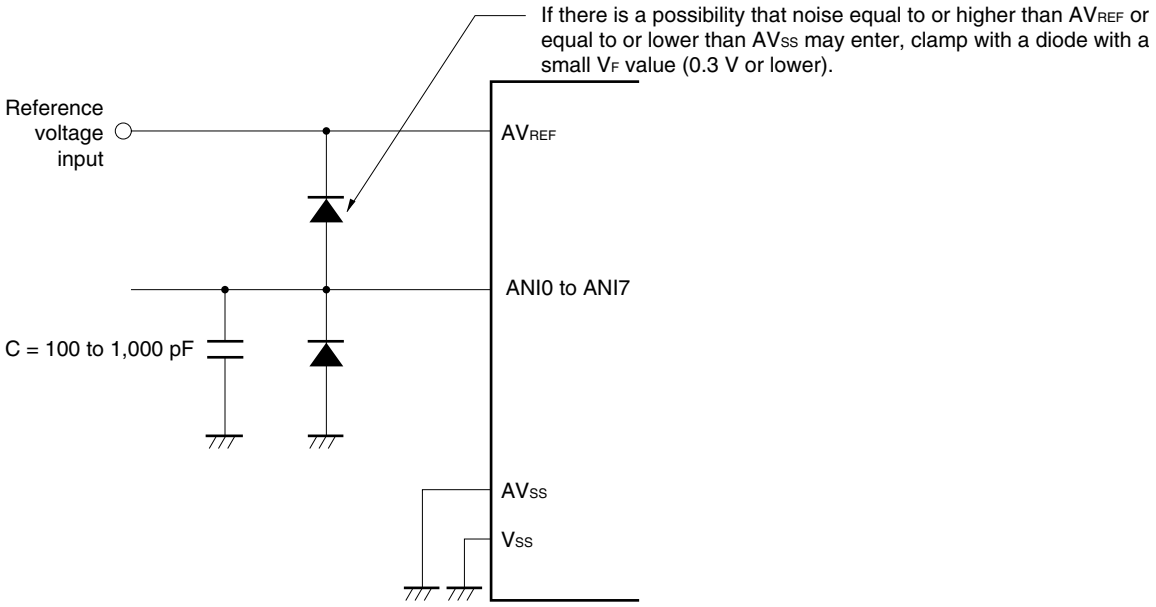
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 12-20 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 12-20. Analog Input Pin Connection



(5) ANI0/P20 to ANI7/P27

<1> The analog input pins (ANI0 to ANI7) are also used as I/O port pins (P20 to P27).
When A/D conversion is performed with any of ANI0 to ANI7 selected, do not access P20 to P27 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27 starting with the ANI0/P20 that is the furthest from AVREF.

<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 to ANI7 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 10 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI7 pins (see **Figure 12-20**).

(7) AVREF pin input impedance

A series resistor string of several tens of kΩ is connected between the AVREF and AVSS 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 AVREF and AVSS pins, resulting in a large reference voltage error.

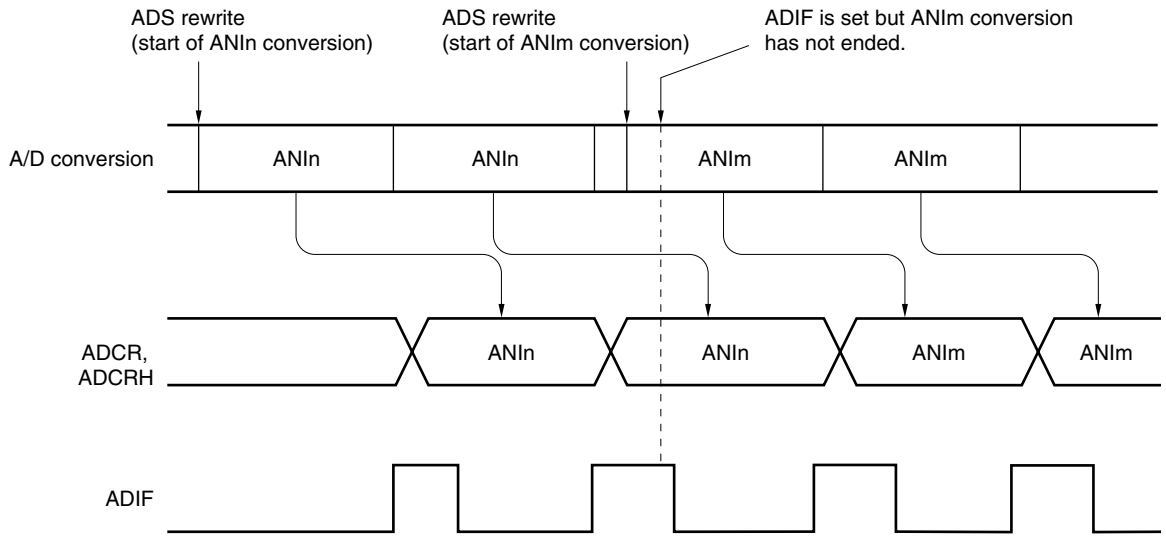
(8) Interrupt request flag (ADIF)

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 12-21. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 7

2. m = 0 to 7

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 12-22. Internal Equivalent Circuit of ANIn Pin

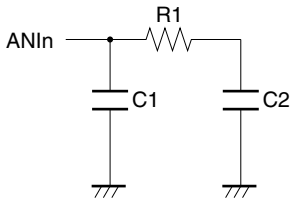


Table 12-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	C1	C2
4.0 V ≤ AV_{REF} ≤ 5.5 V	8.1 kΩ	8 pF	5 pF
2.7 V ≤ AV_{REF} < 4.0 V	31 kΩ	8 pF	5 pF
2.3 V ≤ AV_{REF} < 2.7 V	381 kΩ	8 pF	5 pF

Remarks 1. The resistance and capacitance values shown in Table 12-4 are not guaranteed values.

2. n = 0 to 7

CHAPTER 13 SERIAL INTERFACE UART0

Caution If UART0 is used, CSI10 cannot be used.

13.1 Functions of Serial Interface UART0

Serial interface UART0 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.
For details, see **13.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

The functions of this mode are outlined below.
For details, see **13.4.2 Asynchronous serial interface (UART) mode** and **13.4.3 Dedicated baud rate generator**.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD0: Transmit data output pin
  RxD0: Receive data input pin
- Length of communication data can be selected from 7 or 8 bits.
- Dedicated on-chip 5-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently (full-duplex operation).
- Fixed to LSB-first communication

Cautions 1. If clock supply to serial interface UART0 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART0 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD0 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER0 = 0, RXE0 = 0, and TXE0 = 0.

2. Set POWER0 = 1 and then set TXE0 = 1 (transmission) or RXE0 = 1 (reception) to start communication.

3. TXE0 and RXE0 are synchronized by the base clock (f_{XCLK0}) set by BRGC0. To enable transmission or reception again, set TXE0 or RXE0 to 1 at least two clocks of base clock after TXE0 or RXE0 has been cleared to 0. If TXE0 or RXE0 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.

4. Set transmit data to TXS0 at least one base clock (f_{XCLK0}) after setting TXE0 = 1.

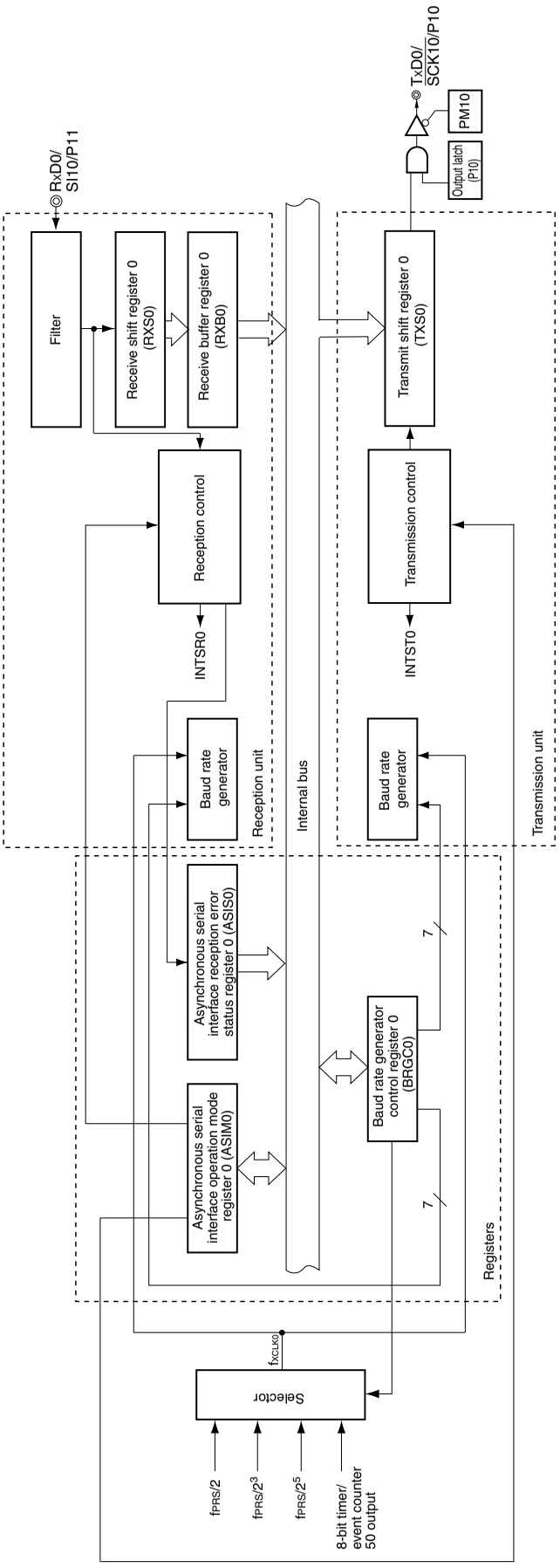
13.2 Configuration of Serial Interface UART0

Serial interface UART0 includes the following hardware.

Table 13-1. Configuration of Serial Interface UART0

Item	Configuration
Registers	Receive buffer register 0 (RXB0) Receive shift register 0 (RXS0) Transmit shift register 0 (TXS0)
Control registers	Asynchronous serial interface operation mode register 0 (ASIM0) Asynchronous serial interface reception error status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0) Port mode register 1 (PM1) Port register 1 (P1)

Figure 13-1. Block Diagram of Serial Interface UART0



(1) Receive buffer register 0 (RXB0)

This 8-bit register stores parallel data converted by receive shift register 0 (RXS0).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 0 (RXS0).

If the data length is set to 7 bits the receive data is transferred to bits 0 to 6 of RXB0 and the MSB of RXB0 is always 0.

If an overrun error (OVE0) occurs, the receive data is not transferred to RXB0.

RXB0 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

Reset signal generation and POWER0 = 0 set this register to FFH.

(2) Receive shift register 0 (RXS0)

This register converts the serial data input to the RxD0 pin into parallel data.

RXS0 cannot be directly manipulated by a program.

(3) Transmit shift register 0 (TXS0)

This register is used to set transmit data. Transmission is started when data is written to TXS0, and serial data is transmitted from the TxD0 pins.

TXS0 can be written by an 8-bit memory manipulation instruction. This register cannot be read.

Reset signal generation, POWER0 = 0, and TXE0 = 0 set this register to FFH.

Cautions 1. Set transmit data to TXS0 at least one base clock (f_{XCLK0}) after setting TXE0 = 1.

2. Do not write the next transmit data to TXS0 before the transmission completion interrupt signal (INTST0) is generated.

13.3 Registers Controlling Serial Interface UART0

Serial interface UART0 is controlled by the following five registers.

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Asynchronous serial interface operation mode register 0 (ASIM0)

This 8-bit register controls the serial communication operations of serial interface UART0.

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

Reset signal generation sets this register to 01H.

Figure 13-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (1/2)

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

Notes 1. The input from the RxD0 pin is fixed to high level when POWER0 = 0.

2. Asynchronous serial interface reception error status register 0 (ASIS0), transmit shift register 0 (TXS0), and receive buffer register 0 (RXB0) are reset.

Figure 13-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (2/2)

PS01	PS00	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL0	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL0	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE0) of asynchronous serial interface reception error status register 0 (ASIS0) is not set and the error interrupt does not occur.

Cautions 1. To start the transmission, set POWER0 to 1 and then set TXE0 to 1. To stop the transmission, clear TXE0 to 0, and then clear POWER0 to 0.

2. To start the reception, set POWER0 to 1 and then set RXE0 to 1. To stop the reception, clear RXE0 to 0, and then clear POWER0 to 0.

3. Set POWER0 to 1 and then set RXE0 to 1 while a high level is input to the RxD0 pin. If POWER0 is set to 1 and RXE0 is set to 1 while a low level is input, reception is started.

4. TXE0 and RXE0 are synchronized by the base clock (f_{XCLK0}) set by BRGC0. To enable transmission or reception again, set TXE0 or RXE0 to 1 at least two clocks of base clock after TXE0 or RXE0 has been cleared to 0. If TXE0 or RXE0 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXS0 at least one base clock (f_{XCLK0}) after setting TXE0 = 1.

6. Clear the TXE0 and RXE0 bits to 0 before rewriting the PS01, PS00, and CL0 bits.

7. Make sure that TXE0 = 0 when rewriting the SL0 bit. Reception is always performed with “number of stop bits = 1”, and therefore, is not affected by the set value of the SL0 bit.

8. Be sure to set bit 0 to 1.

(2) Asynchronous serial interface reception error status register 0 (ASIS0)

This register indicates an error status on completion of reception by serial interface UART0. It includes three error flag bits (PE0, FE0, OVE0).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER0) or bit 5 (RXE0) of ASIM0 to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS0 and then read receive buffer register 0 (RXB0) to clear the error flag.

Figure 13-3. Format of Asynchronous Serial Interface Reception Error Status Register 0 (ASIS0)

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Status flag indicating parity error
0	If POWER0 = 0 or RXE0 = 0, or if ASIS0 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE0	Status flag indicating framing error
0	If POWER0 = 0 or RXE0 = 0, or if ASIS0 register is read.
1	If the stop bit is not detected on completion of reception.

OVE0	Status flag indicating overrun error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If receive data is set to the RXB0 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE0 bit differs depending on the set values of the PS01 and PS00 bits of asynchronous serial interface operation mode register 0 (ASIM0).

2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 0 (RXB0) but discarded.

4. If data is read from ASIS0, a wait cycle is generated. Do not read data from ASIS0 when the peripheral hardware clock (f_{PRS}) is stopped.

(3) Baud rate generator control register 0 (BRGC0)

This register selects the base clock of serial interface UART0 and the division value of the 5-bit counter.

BRGC0 can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 1FH.

Figure 13-4. Format of Baud Rate Generator Control Register 0 (BRGC0)

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	Base clock (f_{XCLK0}) selection				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	TM50 output[Note]				
0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
1	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz

MDL04	MDL03	MDL02	MDL01	MDL00	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	$f_{XCLK0}/8$
0	1	0	0	1	9	$f_{XCLK0}/9$
0	1	0	1	0	10	$f_{XCLK0}/10$
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	0	26	$f_{XCLK0}/26$
1	1	0	1	1	27	$f_{XCLK0}/27$
1	1	1	0	0	28	$f_{XCLK0}/28$
1	1	1	0	1	29	$f_{XCLK0}/29$
1	1	1	1	0	30	$f_{XCLK0}/30$
1	1	1	1	1	31	$f_{XCLK0}/31$

Note Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)

  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)

  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Caution 1. Make sure that bit 6 (TXE0) and bit 5 (RXE0) of the ASIM0 register = 0 when rewriting the MDL04 to MDL00 bits.

Cautions 2. Make sure that bit 7 (POWER0) of the ASIM0 register = 0 when rewriting the TPS01 and TPS00 bits.

3. The baud rate value is the output clock of the 5-bit counter divided by 2.

Remarks 1. f_{XCLK0}: Frequency of base clock selected by the TPS01 and TPS00 bits

2. f_{PRS}: Peripheral hardware clock frequency

3. k: Value set by the MDL04 to MDL00 bits (k = 8, 9, 10, ..., 31)

4. ×: Don't care

5. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)

TMC501: Bit 1 of TMC50

(4) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using the P10/TxD0/$\overline{SCK10}$ pin for serial interface data output, clear PM10 to 0 and set the output latch of P10 to 1.

When using the P11/RxD0/SI10 pin for serial interface data input, set PM11 to 1. The output latch of P11 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 13-5. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

13.4 Operation of Serial Interface UART0

Serial interface UART0 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

13.4.1 Operation stop mode

In this mode, serial communication cannot be executed, thus reducing the power consumption. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER0, TXE0, and RXE0) of ASIM0 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 0 (ASIM0).

ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The input from the RxD0 pin is fixed to high level when POWER0 = 0.

2. Asynchronous serial interface reception error status register 0 (ASIS0), transmit shift register 0 (TXS0), and receive buffer register 0 (RXB0) are reset.

Caution Clear POWER0 to 0 after clearing TXE0 and RXE0 to 0 to set the operation stop mode. To start the communication, set POWER0 to 1, and then set TXE0 or RXE0 to 1.

Remark To use the RxD0/SI10/P11 and TxD0/$\overline{\text{SCK10}}$/P10 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

13.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the BRGC0 register (see **Figure 13-4**).
<2> Set bits 1 to 4 (SL0, CL0, PS00, and PS01) of the ASIM0 register (see **Figure 13-2**).
<3> Set bit 7 (POWER0) of the ASIM0 register to 1.
<4> Set bit 6 (TXE0) of the ASIM0 register to 1. → Transmission is enabled.
Set bit 5 (RXE0) of the ASIM0 register to 1. → Reception is enabled.
<5> Write data to the TXS0 register. → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 13-2. Relationship Between Register Settings and Pins

POWER0	TXE0	RXE0	PM10	P10	PM11	P11	UART0 Operation	Pin Function	
								TxD0/$\overline{\text{SCK10}}$/P10	RxD0/SI10/P11
0	0	0	×Note	×Note	×Note	×Note	Stop	$\overline{\text{SCK10}}$/P10	SI10/P11
1	0	1	×Note	×Note	1	×	Reception	$\overline{\text{SCK10}}$/P10	RxD0
	1	0	0	1	×Note	×Note	Transmission	TxD0	SI10/P11
	1	1	0	1	1	×	Transmission/ reception	TxD0	RxD0

Note Can be set as port function or serial interface CSI10.

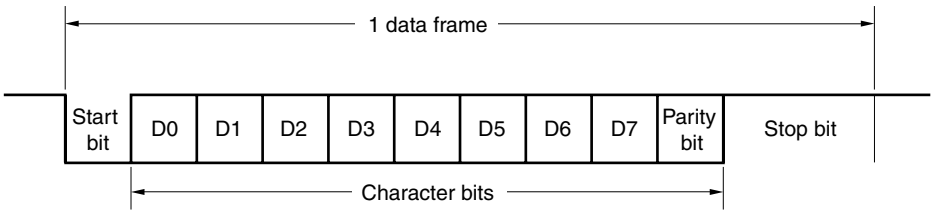
Remark ×: don't care
POWER0: Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
PM1×: Port mode register
P1×: Port output latch

(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

Figures 13-6 and 13-7 show the format and waveform example of the normal transmit/receive data.

Figure 13-6. Format of Normal UART Transmit/Receive Data



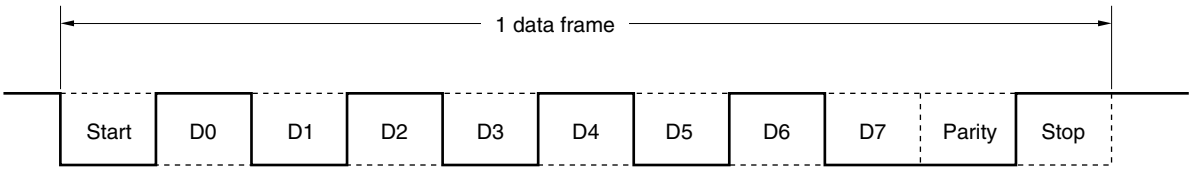
One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits (LSB first)
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

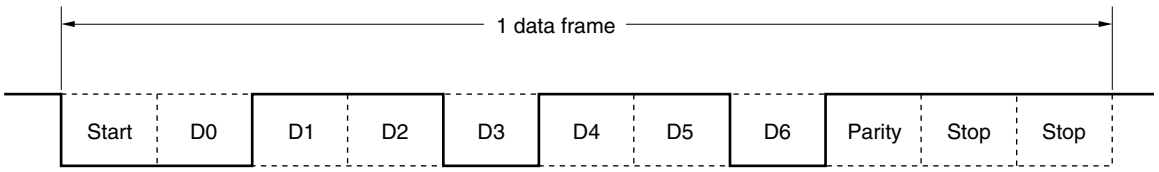
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 0 (ASIM0).

Figure 13-7. Example of Normal UART Transmit/Receive Data Waveform

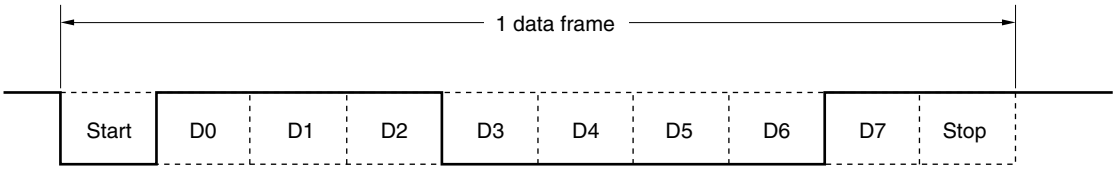
1. Data length: 8 bits, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



2. Data length: 7 bits, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H



3. Data length: 8 bits, Parity: None, Stop bit: 1 bit, Communication data: 87H



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are "1" is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are "1": 1
  If transmit data has an even number of bits that are "1": 0

- Reception

  The number of bits that are "1" in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are "1" is odd.

  If transmit data has an odd number of bits that are "1": 0
  If transmit data has an even number of bits that are "1": 1

- Reception

  The number of bits that are "1" in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is "0" or "1".

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Transmission

If bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is set to 1 and bit 6 (TXE0) of ASIM0 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit shift register 0 (TXS0). The start bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the start bit is output from the TxD0 pin, and the transmit data is output followed by the rest of the data in order starting from the LSB. When transmission is completed, the parity and stop bits set by ASIM0 are appended and a transmission completion interrupt request (INTST0) is generated.

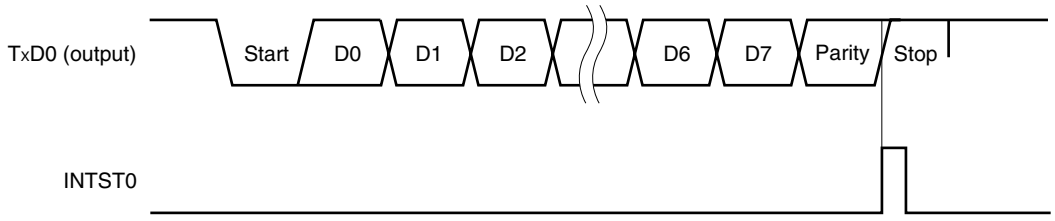
Transmission is stopped until the data to be transmitted next is written to TXS0.

Figure 13-8 shows the timing of the transmission completion interrupt request (INTST0). This interrupt occurs as soon as the last stop bit has been output.

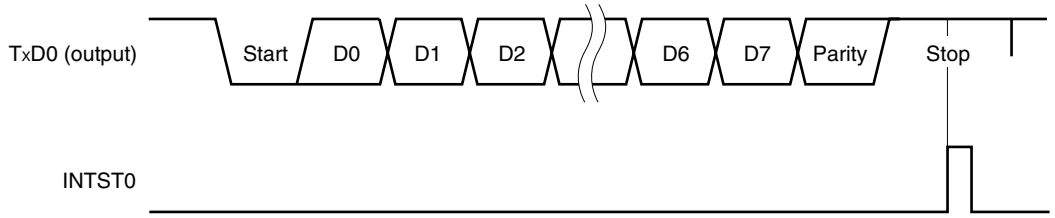
Caution After transmit data is written to TXS0, do not write the next transmit data before the transmission completion interrupt signal (INTST0) is generated.

Figure 13-8. Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Reception

Reception is enabled and the RxD0 pin input is sampled when bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is set to 1 and then bit 5 (RXE0) of ASIM0 is set to 1.

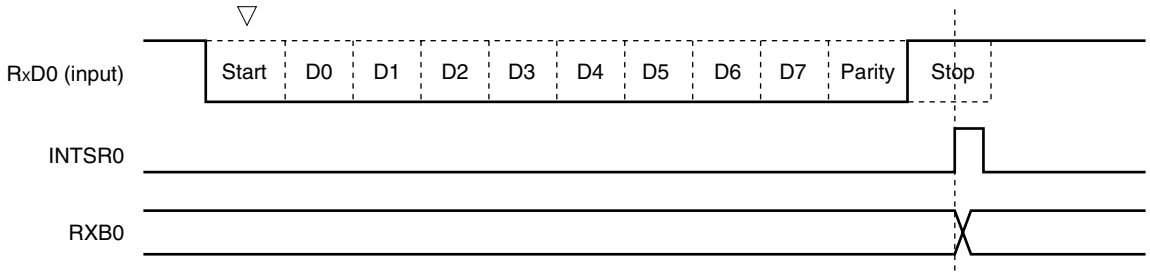
The 5-bit counter of the baud rate generator starts counting when the falling edge of the RxD0 pin input is detected. When the set value of baud rate generator control register 0 (BRGC0) has been counted, the RxD0 pin input is sampled again (in Figure 13-9). If the RxD0 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in receive shift register 0 (RXS0) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR0) is generated and the data of RXS0 is written to receive buffer register 0 (RXB0). If an overrun error (OVE0) occurs, however, the receive data is not written to RXB0.

Even if a parity error (PE0) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an reception error interrupt (INTSR0) is generated after completion of reception.

INTSR0 occurs upon completion of reception and in case of a reception error.

Figure 13-9. Reception Completion Interrupt Request Timing



Cautions 1. If a reception error occurs, read asynchronous serial interface reception error status register 0 (ASIS0) and then read receive buffer register 0 (RXB0) to clear the error flag. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

(e) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 0 (ASIS0) is set as a result of data reception, a reception error interrupt (INTSR0) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS0 in the reception error interrupt (INTSR0) servicing (see **Figure 13-3**).

The contents of ASIS0 are cleared to 0 when ASIS0 is read.

Table 13-3. Cause of Reception Error

Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 0 (RXB0).

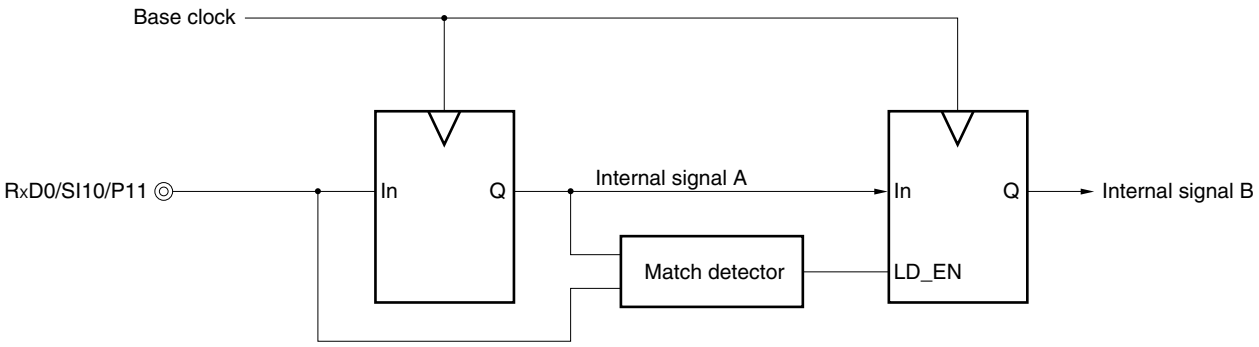
(f) Noise filter of receive data

The RxD0 signal is sampled using the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 13-10, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 13-10. Noise Filter Circuit



13.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and a 5-bit programmable counter, and generates a serial clock for transmission/reception of UART0.

Separate 5-bit counters are provided for transmission and reception.

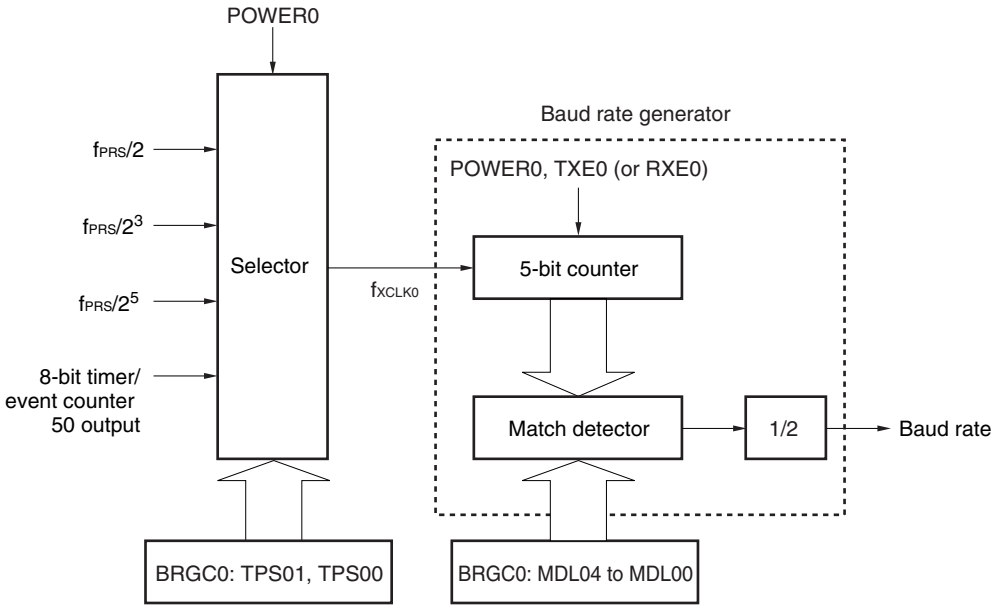
(1) Configuration of baud rate generator

- Base clock
  The clock selected by bits 7 and 6 (TPS01 and TPS00) of baud rate generator control register 0 (BRGC0) is supplied to each module when bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is 1. This clock is called the base clock and its frequency is called f_{XCLK0}. The base clock is fixed to low level when POWER0 = 0.

- Transmission counter
  This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 6 (TXE0) of asynchronous serial interface operation mode register 0 (ASIM0) is 0.
  It starts counting when POWER0 = 1 and TXE0 = 1.
  The counter is cleared to 0 when the first data transmitted is written to transmit shift register 0 (TXS0).

- Reception counter
  This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 5 (RXE0) of asynchronous serial interface operation mode register 0 (ASIM0) is 0.
  It starts counting when the start bit has been detected.
  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 13-11. Configuration of Baud Rate Generator



Remark POWER0: Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
BRGC0: Baud rate generator control register 0

(2) Generation of serial clock

A serial clock to be generated can be specified by using baud rate generator control register 0 (BRGC0).

Select the clock to be input to the 5-bit counter by using bits 7 and 6 (TPS01 and TPS00) of BRGC0.

Bits 4 to 0 (MDL04 to MDL00) of BRGC0 can be used to select the division value ($f_{XCLK0}/8$ to $f_{XCLK0}/31$) of the 5-bit counter.

13.4.4 Calculation of baud rate

(1) Baud rate calculation expression

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK0}}{2 \times k}$ [bps]

f_{XCLK0}: Frequency of base clock selected by the TPS01 and TPS00 bits of the BRGC0 register

k: Value set by the MDL04 to MDL00 bits of the BRGC0 register (k = 8, 9, 10, ..., 31)

Table 13-4. Set Value of TPS01 and TPS00

TPS01	TPS00	Base clock (f_{XCLK0}) selection				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	TM50 output[Note]				
0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
1	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz

Note Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

(2) Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz
Set value of MDL04 to MDL00 bits of BRGC0 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)
= 2,500,000/(2 × 16) = 78,125 [bps]

Error = (78,125/76,800 − 1) × 100
= 1.725 [%]

(3) Example of setting baud rate

Table 13-5. Set Data of Baud Rate Generator

Baud Rate [bps]	f_{PRS} = 2.0 MHz				f_{PRS} = 5.0 MHz				f_{PRS} = 10.0 MHz				f_{PRS} = 20.0 MHz			
	TPS01, TPS00	k	Calculated Value	ERR [%]	TPS01, TPS00	k	Calculated Value	ERR [%]	TPS01, TPS00	k	Calculated Value	ERR [%]	TPS01, TPS00	k	Calculated Value	ERR [%]
4800	2H	26	4808	0.16	3H	16	4883	1.73	–	–	–	–	–	–	–	–
9600	2H	13	9615	0.16	3H	8	9766	1.73	3H	16	9766	1.73	–	–	–	–
10400	2H	12	10417	0.16	2H	30	10417	0.16	3H	15	10417	0.16	3H	30	10417	0.16
19200	1H	26	19231	0.16	2H	16	19531	1.73	3H	8	19531	1.73	3H	16	19531	1.73
24000	1H	21	23810	–0.79	2H	13	24038	0.16	2H	26	24038	0.16	3H	13	24038	0.16
31250	1H	16	31250	0	2H	10	31250	0	2H	20	31250	0	3H	10	31250	0
33600	1H	15	33333	–0.79	2H	9	34722	3.34	2H	19	32895	–2.1	3H	9	34722	3.34
38400	1H	13	38462	0.16	2H	8	39063	1.73	2H	16	39063	1.73	3H	8	39063	1.73
56000	1H	9	55556	–0.79	1H	22	56818	1.46	2H	11	56818	1.46	2H	22	56818	1.46
62500	1H	8	62500	0	1H	20	62500	0	2H	10	62500	0	2H	20	62500	0
76800	–	–	–	–	1H	16	78125	1.73	2H	8	78125	1.73	2H	16	78125	1.73
115200	–	–	–	–	1H	11	113636	–1.36	1H	22	113636	–1.36	2H	11	113636	–1.36
153600	–	–	–	–	1H	8	156250	1.73	1H	16	156250	1.73	2H	8	156250	1.73
312500	–	–	–	–	–	–	–	–	1H	8	312500	0	1H	16	312500	0
625000	–	–	–	–	–	–	–	–	–	–	–	–	1H	8	625000	0

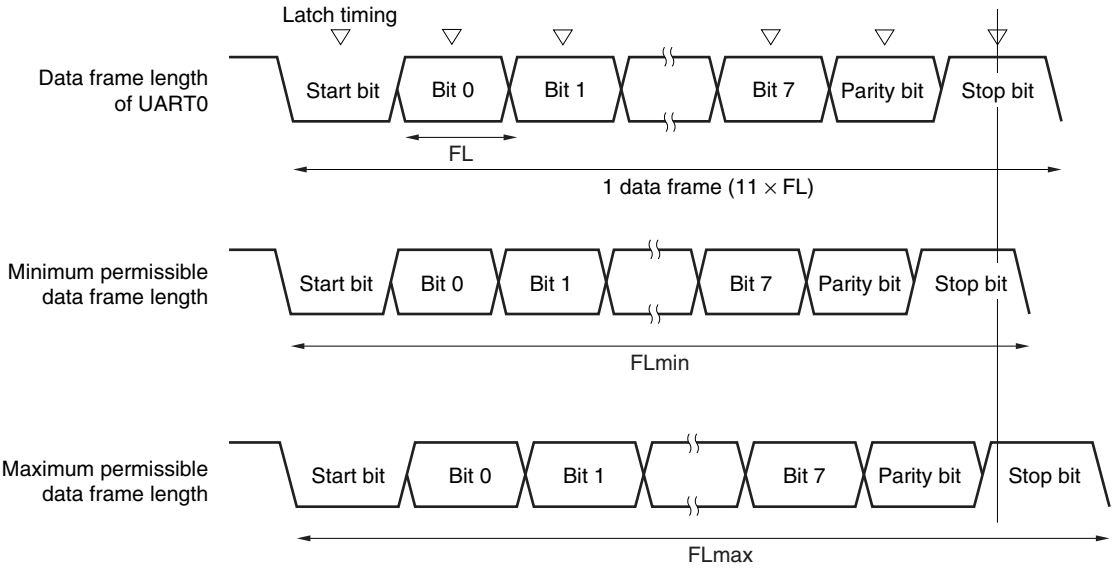
Remark TPS01, TPS00: Bits 7 and 6 of baud rate generator control register 0 (BRGC0) (setting of base clock (f_{XCLK0}))
k: Value set by the MDL04 to MDL00 bits of BRGC0 (k = 8, 9, 10, ..., 31)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 13-12. Permissible Baud Rate Range During Reception



As shown in Figure 13-12, the latch timing of the receive data is determined by the counter set by baud rate generator control register 0 (BRGC0) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART0
k: Set value of BRGC0
FL: 1-bit data length
Margin of latch timing: 2 clocks

Minimum permissible data frame length: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} Brate$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} Brate$$

The permissible baud rate error between UART0 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 13-6. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	–3.61%
16	+4.14%	–4.19%
24	+4.34%	–4.38%
31	+4.44%	–4.47%

Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC0

CHAPTER 14 SERIAL INTERFACES UART60 AND UART61

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
Serial interface UART60	√	
Serial interface UART61	–	√ Note

Note LIN (Local Interconnect Network)-bus is not supported in UART61.

Caution If receiving data using UART61, INTTM001 of TM01 cannot be used (only 64-pin products).
If transmitting data using UART61, INTTM011 of TM01 cannot be used (only 64-pin products).

Remark √: Mounted, –: Not mounted

14.1 Functions of Serial Interfaces UART60 and UART61

Serial interfaces UART60 and UART61 have the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.
For details, see **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

The functions of this mode are outlined below.
For details, see **14.4.2 Asynchronous serial interface (UART) mode** and **14.4.3 Dedicated baud rate generator**.
Only for UART60, the LIN (Local Interconnect Network)-bus.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD6n: Transmit data output pin
  RxD6n: Receive data input pin
- Data length of communication data can be selected from 7 or 8 bits.
- Dedicated internal 8-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently (full duplex operation).
- MSB- or LSB-first communication selectable
- Inverted transmission operation
- Sync break field transmission from 13 to 20 bits
- More than 11 bits can be identified for sync break field reception (only for UART60, SBF reception flag provided).

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

Cautions 1. The TxD6n output inversion function inverts only the transmission side and not the reception side. To use this function, the reception side must be ready for reception of inverted data.

2. If clock supply to serial interface UART6n is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART6n is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6n pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER6n = 0, RXE6n = 0, and TXE6n = 0.

3. Set POWER6n = 1 and then set TXE6n = 1 (transmission) or RXE6n = 1 (reception) to start communication.

4. TXE6n and RXE6n are synchronized by the base clock (f_{XCLK6n}) set by CKSR6n. To enable transmission or reception again, set TXE6n or RXE6n to 1 at least two clocks of the base clock after TXE6n or RXE6n has been cleared to 0. If TXE6n or RXE6n is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXB6n at least one base clock (f_{XCLK6n}) after setting TXE6n = 1.

6. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use continuous transmission when UART60 is used in LIN communication.

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

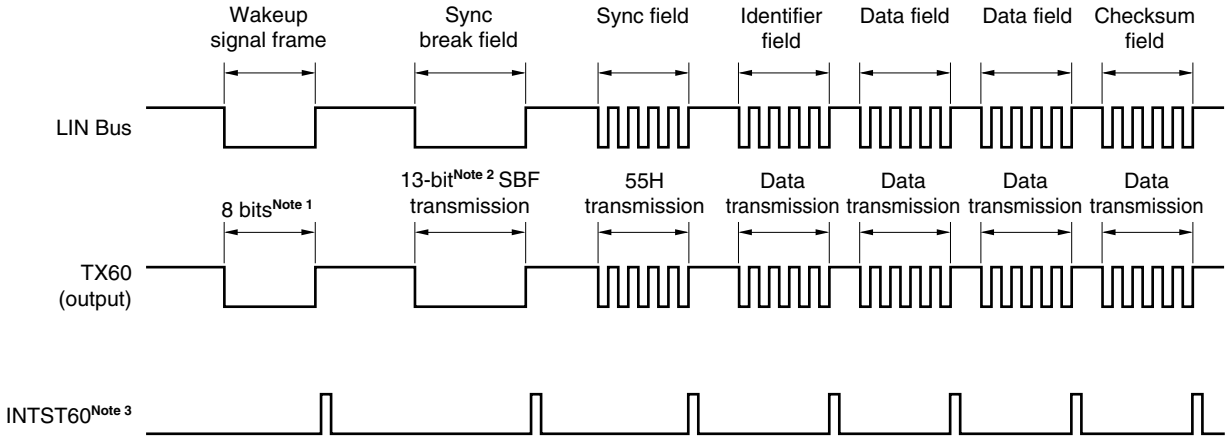
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 14-1 and 14-2 outline the transmission and reception operations of LIN.

Figure 14-1. LIN Transmission Operation



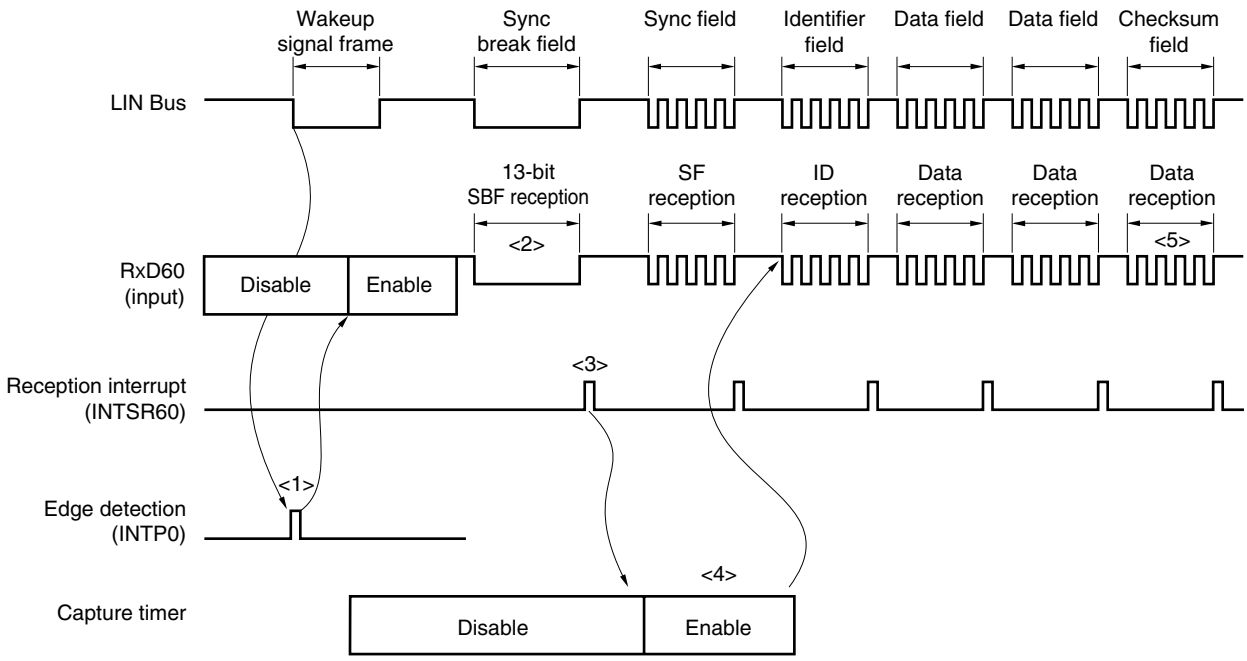
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL602 to SBL600) of asynchronous serial interface control register 60 (ASICL60) (see **14.4.2 (2) (h) SBF transmission**).

3. INTST60 is output on completion of each transmission. It is also output when SBF is transmitted.

Remark The interval between each field is controlled by software.

Figure 14-2. LIN Reception Operation



Reception processing is as follows.

<1> The wakeup signal is detected at the edge of the pin, and enables UART60 and sets the SBF reception mode.

<2> Reception continues until the STOP bit is detected. When an SBF with low-level data of 11 bits or more has been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If an SBF with low-level data of less than 11 bits has been detected, it is assumed that an SBF reception error has occurred. The interrupt signal is not output and the SBF reception mode is restored.

<3> If SBF reception has been completed correctly, an interrupt signal is output. Start 16-bit timer/event counter 00 by the SBF reception end interrupt servicing and measure the bit interval (pulse width) of the sync field (see **6.4.8 Pulse width measurement operation**). Detection of errors OVE60, PE60, and FE60 is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB60 is not performed. The shift register holds the reset value FFH.

<4> Calculate the baud rate error from the bit interval of the sync field, disable UART60 after SF reception, and then re-set baud rate generator control register 60 (BRGC60).

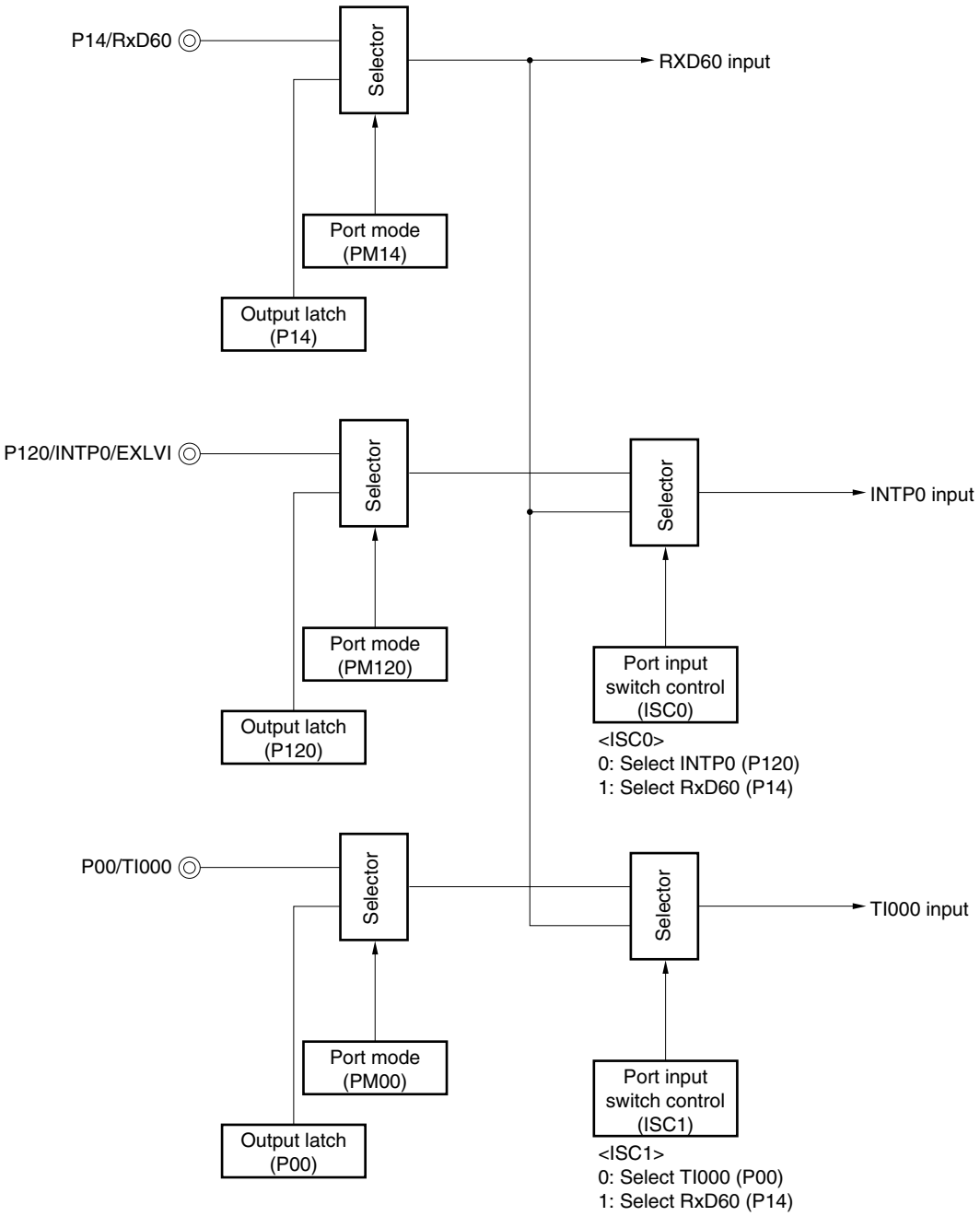
<5> Distinguish the checksum field by software. Also perform processing by software to initialize UART60 after reception of the checksum field and to set the SBF reception mode again.

Figure 14-3 shows the port configuration for LIN reception operation.

The wakeup signal transmitted from the LIN master is received by detecting the edge of the external interrupt (INTP0). The length of the sync field transmitted from the LIN master can be measured using the external event capture operation of 16-bit timer/event counter 00, and the baud rate error can be calculated.

The input source of the reception port input (RxD60) can be input to the external interrupt (INTP0) and 16-bit timer/event counter 00 by port input switch control (ISC0/ISC1), without connecting RxD60 and INTP0/TI000 externally.

Figure 14-3. Port Configuration for LIN Reception Operation



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (see **Figure 14-11**)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTP0); wakeup signal detection
  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter 00 (TI000); baud rate error detection
  Use: Detects the baud rate error (measures the TI000 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART60

14.2 Configuration of Serial Interfaces UART60 and UART61

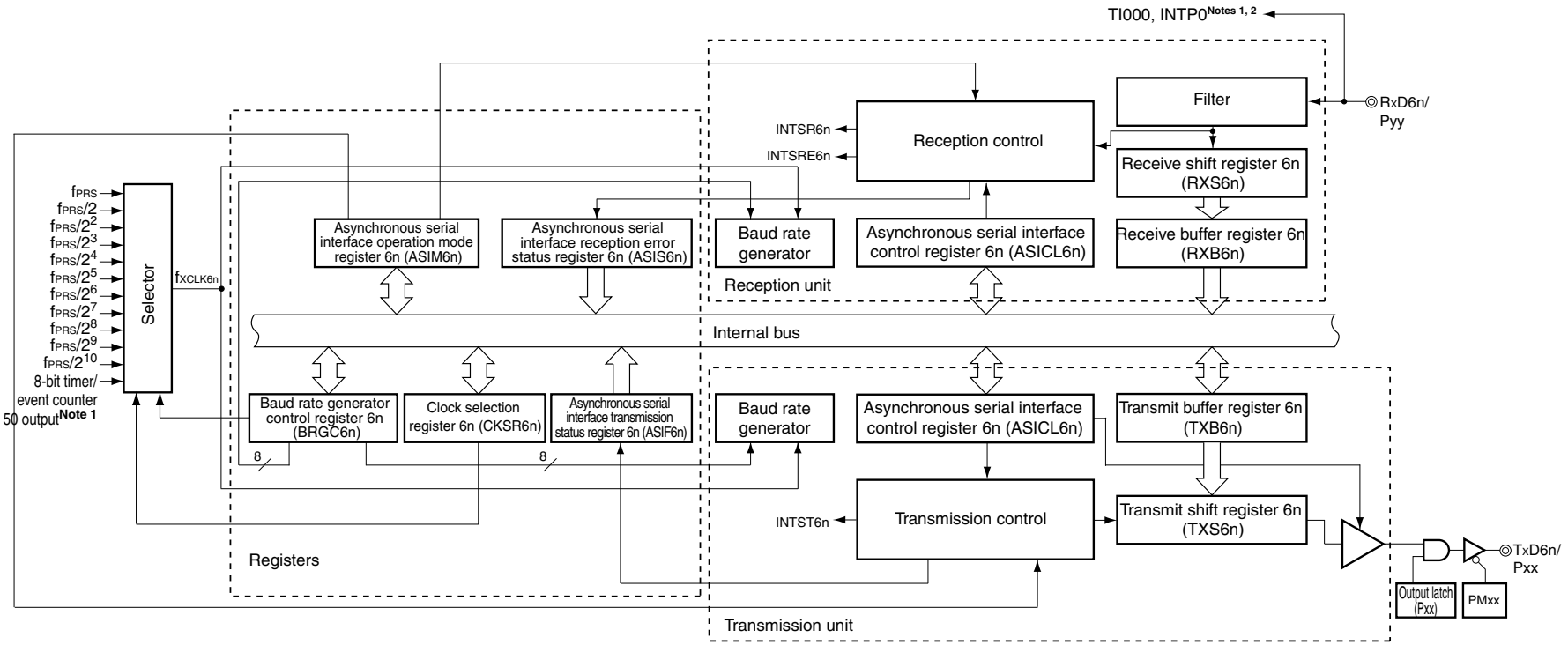
Serial interfaces UART60 and UART61 includ the following hardware.

Table 14-1. Configuration of Serial Interface UART6n

Item	Configuration
Registers	Receive buffer register 6n (RXB6n) Receive shift register 6n (RXS6n) Transmit buffer register 6n (TXB6n) Transmit shift register 6n (TXS6n)
Control registers	Asynchronous serial interface operation mode register 6n (ASIM6n) Asynchronous serial interface reception error status register 6n (ASIS6n) Asynchronous serial interface transmission status register 6n (ASIF6n) Clock selection register 6n (CKSR6n) Baud rate generator control register 6n (BRGC6n) Asynchronous serial interface control register 6n (ASICL6n) Input switch control register (ISC) Port mode registers 1, 4 (PM1, PM4) Port registers 1, 4 (P1, P4)

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

Figure 14-4. Block Diagram of Serial Interfaces UART60 and UART61



Notes 1. UART60 only

2. Selectable with input switch control register (ISC).

Remark n = 0, xx = 13, yy = 14

n = 1, xx = 42, yy = 43

(n = 0: 78K0/KC2-C, n = 0, 1: 78K0/KE2-C)

(1) Receive buffer register 6n (RXB6n)

This 8-bit register stores parallel data converted by receive shift register 6n (RXS6n).

Each time 1 byte of data has been received, new receive data is transferred to this register from RXS6n. If the data length is set to 7 bits, data is transferred as follows.

- In LSB-first reception, the receive data is transferred to bits 0 to 6 of RXB6n and the MSB of RXB6n is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6n and the LSB of RXB6n is always 0.

If an overrun error (OVE6n) occurs, the receive data is not transferred to RXB6n.

RXB6n can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

Reset signal generation sets this register to FFH.

(2) Receive shift register 6n (RXS6n)

This register converts the serial data input to the RxD6n pin into parallel data.

RXS6n cannot be directly manipulated by a program.

(3) Transmit buffer register 6n (TXB6n)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6n.

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

Reset signal generation sets this register to FFH.

Cautions 1. Do not write data to TXB6n when bit 1 (TXBF6n) of asynchronous serial interface transmission status register 6n (ASIF6n) is 1.

2. Do not refresh (write the same value to) TXB6n by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) are 1 or when bits 7 and 5 (POWER6n, RXE6n) of ASIM6n are 1).

3. Set transmit data to TXB6n at least one base clock (f_{XCLK6n}) after setting TXE6n = 1.

(4) Transmit shift register 6n (TXS6n)

This register transmits the data transferred from TXB6n from the TxD6n pin as serial data. Data is transferred from TXB6n immediately after TXB6n is written for the first transmission, or immediately before INTST6n occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6n and transmitted from the TxD6n pin at the falling edge of the base clock.

TXS6n cannot be directly manipulated by a program.

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

14.3 Registers Controlling Serial Interfaces UART60 and UART61

Serial interfaces UART60 and UART61 are controlled by the following nine registers.

- Asynchronous serial interface operation mode register 6n (ASIM6n)
- Asynchronous serial interface reception error status register 6n (ASIS6n)
- Asynchronous serial interface transmission status register 6n (ASIF6n)
- Clock selection register 6n (CKSR6n)
- Baud rate generator control register 6n (BRGC6n)
- Asynchronous serial interface control register 6n (ASICL6n)
- Input switch control register (ISC)
- Port mode registers 1, 4 (PM1, PM4)
- Port registers 1, 4 (P1, P4)

(1) Asynchronous serial interface operation mode register 6n (ASIM6n)

This 8-bit register controls the serial communication operations of serial interface UART6n.

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

Reset signal generation sets this register to 01H.

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. ASIM6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

Figure 14-5. Format of Asynchronous Serial Interface Operation Mode Register 6n (ASIM6n) (1/2)

Address: FF50H (UART60), FF90H (UART61) After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6n	POWER6n	TXE6n	RXE6n	PS6n1	PS6n0	CL6n	SL6n	ISRM6n

POWER6n	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1	Enables operation of the internal operation clock

TXE6n	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE6n	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. If POWER6n = 0 is set while transmitting data, the output of the TxD6n pin will be fixed to high level (if TXDLV6n = 0). Furthermore, the input from the RxD6n pin will be fixed to high level.

2. Asynchronous serial interface reception error status register 6n (ASIS6n), asynchronous serial interface transmission status register 6n (ASIF6n), bit 7 (SBRF6n) and bit 6 (SBRT6n) of asynchronous serial interface control register 6n (ASICL6n), and receive buffer register 6n (RXB6n) are reset.

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

Figure 14-5. Format of Asynchronous Serial Interface Operation Mode Register 6n (ASIM6n) (2/2)

PS6n1	PS6n0	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL6n	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6n	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6n	Enables/disables occurrence of reception completion interrupt in case of error
0	“INTSRE6n” occurs in case of error (at this time, INTSR6n does not occur).
1	“INTSR6n” occurs in case of error (at this time, INTSRE6n does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE6n) of asynchronous serial interface reception error status register 6n (ASIS6n) is not set and the error interrupt does not occur.

Cautions
1. **To start the transmission, set POWER6n to 1 and then set TXE6n to 1. To stop the transmission, clear TXE6n to 0, and then clear POWER6n to 0.**
2. **To start the reception, set POWER6n to 1 and then set RXE6n to 1. To stop the reception, clear RXE6n to 0, and then clear POWER6n to 0.**
3. **Set POWER6n to 1 and then set RXE6n to 1 while a high level is input to the RxD6n pin. If POWER6n is set to 1 and RXE6n is set to 1 while a low level is input, reception is started.**
4. **TXE6n and RXE6n are synchronized by the base clock (f_{XCLK6n}) set by CKSR6n. To enable transmission or reception again, set TXE6n or RXE6n to 1 at least two clocks of the base clock after TXE6n or RXE6n has been cleared to 0. If TXE6n or RXE6n is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.**
5. **Set transmit data to TXB6n at least one base clock (f_{XCLK6n}) after setting TXE6n = 1.**
6. **Clear the TXE6n and RXE6n bits to 0 before rewriting the PS6n1, PS6n0, and CL6n bits.**
7. **Fix the PS6n1 and PS6n0 bits to 0 when UART60 is used in LIN communication operation.**
8. **Clear TXE6n to 0 before rewriting the SL6n bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL6n bit.**
9. **Make sure that RXE6n = 0 when rewriting the ISRM6n bit.**

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

(2) Asynchronous serial interface reception error status register 6n (ASIS6n)

This register indicates an error status on completion of reception by serial interface UART6n. It includes three error flag bits (PE6n, FE6n, OVE6n).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6n) or bit 5 (RXE6n) of ASIM6n to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS6n and then read receive buffer register 6n (RXB6n) to clear the error flag.

Figure 14-6. Format of Asynchronous Serial Interface Reception Error Status Register 6n (ASIS6n)

Address: FF53H (UART60), FF93H (UART61) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6n	0	0	0	0	0	PE6n	FE6n	OVE6n

PE6n	Status flag indicating parity error
0	If POWER6n = 0 or RXE6n = 0, or if ASIS6n register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE6n	Status flag indicating framing error
0	If POWER6n = 0 or RXE6n = 0, or if ASIS6n register is read
1	If the stop bit is not detected on completion of reception

OVE6n	Status flag indicating overrun error
0	If POWER6n = 0 or RXE6n = 0, or if ASIS6n register is read
1	If receive data is set to the RXB6n register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE6n bit differs depending on the set values of the PS6n1 and PS6n0 bits of asynchronous serial interface operation mode register 6n (ASIM6n).

2. For the stop bit of the receive data, only the first stop bit is checked regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 6n (RXB6n) but discarded.

4. If data is read from ASIS6n, a wait cycle is generated. Do not read data from ASIS6n when the peripheral hardware clock (f_{PRS}) is stopped.

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

(3) Asynchronous serial interface transmission status register 6n (ASIF6n)

This register indicates the status of transmission by serial interface UART6n. It includes two status flag bits (TXBF6n and TXSF6n).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6n register after data has been transferred from the TXB6n register to the TXS6n register.

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6n) or bit 6 (TXE6n) of ASIM6n to 0 clears this register to 00H.

Figure 14-7. Format of Asynchronous Serial Interface Transmission Status Register 6n (ASIF6n)

Address: FF55H (UART60), FF95H (UART61) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6n	0	0	0	0	0	0	TXBF6n	TXSF6n

TXBF6n	Transmit buffer data flag
0	If POWER6n = 0 or TXE6n = 0, or if data is transferred to transmit shift register 6n (TXS6n)
1	If data is written to transmit buffer register 6n (TXB6n) (if data exists in TXB6n)

TXSF6n	Transmit shift register data flag
0	If POWER6n = 0 or TXE6n = 0, or if the next data is not transferred from transmit buffer register 6n (TXB6n) after completion of transfer
1	If data is transferred from transmit buffer register 6n (TXB6n) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB6n register. Be sure to check that the TXBF6n flag is "0". If so, write the next transmit data (second byte) to the TXB6n register. If data is written to the TXB6n register while the TXBF6n flag is "1", the transmit data cannot be guaranteed.

2. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6n flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6n flag is "1", the transmit data cannot be guaranteed.

(4) Clock selection register 6n (CKSR6n)

This register selects the base clock of serial interface UART6n.

CKSR6n can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. CKSR6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

Figure 14-8. Format of Clock Selection Register 6n (CKSR6n)

Address: FF56H (UART60), FF96H (UART61) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6n	0	0	0	0	TPS6n3	TPS6n2	TPS6n1	TPS6n0

TPS6n3	TPS6n2	TPS6n1	TPS6n0	Base clock (f_{XCLK6n}) selection				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output Notes 1, 2				
Other than above				Setting prohibited				

Notes 1. UART60 only

2. Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Caution Make sure POWER6n = 0 when rewriting TPS6n3 to TPS6n0.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

3. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(5) Baud rate generator control register 6n (BRGC6n)

This register sets the division value of the 8-bit counter of serial interface UART6n.

BRGC6n can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Remark BRGC6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

Figure 14-9. Format of Baud Rate Generator Control Register 6n (BRGC6n)

Address: FF57H (UART60), FF97H (UART61) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6n	MDL6n7	MDL6n6	MDL6n5	MDL6n4	MDL6n3	MDL6n2	MDL6n1	MDL6n0

MDL6n7	MDL6n6	MDL6n5	MDL6n4	MDL6n3	MDL6n2	MDL6n1	MDL6n0	k	Output clock selection of 8-bit counter
0	0	0	0	0	0	×	×	×	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{XCLK6n}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK6n}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK6n}/6$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6n}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6n}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6n}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6n}/255$

Cautions 1. Make sure that bit 6n (TXE6n) and bit 5 (RXE6n) of the ASIM6n register = 0 when rewriting the MDL6n7 to MDL6n0 bits.

2. The baud rate is the output clock of the 8-bit counter divided by 2.

Remarks 1. f_{XCLK6n}: Frequency of base clock selected by the TPS6n3 to TPS6n0 bits of CKSR6n register

2. n = 0: 78K0/KC2-C

n = 0, 1: 78K0/KE2-C

3. k: Value set by MDL6n7 to MDL6n0 bits (k = 4, 5, 6, ..., 255)

4. ×: Don't care

(6) Asynchronous serial interface control register 6n (ASICL6n)

This register controls the serial communication operations of serial interface UART6n.

ASICL6n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 16H.

Caution ASICL6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1). However, do not set both SBRT6n and SBTT6n to 1 by a refresh operation during SBF reception (SBRT6n = 1) or SBF transmission (until INTST6n occurs since SBTT6n has been set (1)) for UART60, because it may re-trigger SBF reception or SBF transmission.

Figure 14-10. Format of Asynchronous Serial Interface Control Register 6n (ASICL6n) (1/2)

Address: FF58H (UART60), FFB5H (UART61) After reset: 16H R/W[Note 1]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL6n	SBRF60[Note 2]	SBRT60[Note 2]	SBTT60[Note 2]	SBL602[Note 2]	SBL601[Note 2]	SBL600[Note 2]	DIR6n	TXDLV6n

SBRF60[Note 2]	SBF reception status flag
0	If POWER6n = 0 and RXE6n = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT60[Note 2]	SBF reception trigger
0	–
1	SBF reception trigger

SBTT60[Note 2]	SBF transmission trigger
0	–
1	SBF transmission trigger

Notes 1. Bit 7 is read-only.

2. UART60 only

Remark n = 0: 78K0/KC2-C

n = 0, 1: 78K0/KE2-C

Figure 14-10. Format of Asynchronous Serial Interface Control Register 6n (ASICL6n) (2/2)

SBL602[Note]	SBL601[Note]	SBL600[Note]	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6n	First-bit specification
0	MSB
1	LSB

TXDLV6n	Enables/disables inverting TxD6n output
0	Normal output of TxD6n
1	Inverted output of TxD6n

Note UART60 only

Cautions
1. **In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF60 flag is held (1).**
2. **Before setting the SBRT60 bit, make sure that bit 7 (POWER60) and bit 5 (RXE60) of ASIM60 = 1. After setting the SBRT60 bit to 1, do not clear it to 0 before SBF reception is completed (before an interrupt request signal is generated).**
3. **The read value of the SBRT60 bit is always 0. SBRT60 is automatically cleared to 0 after SBF reception has been correctly completed.**
4. **Before setting the SBTT60 bit to 1, make sure that bit 7 (POWER60) and bit 6 (TXE60) of ASIM60 = 1. After setting the SBTT60 bit to 1, do not clear it to 0 before SBF transmission is completed (before an interrupt request signal is generated).**
5. **The read value of the SBTT60 bit is always 0. SBTT60 is automatically cleared to 0 at the end of SBF transmission.**
6. **Do not set the SBRT60 bit to 1 during reception, and do not set the SBTT60 bit to 1 during transmission.**
7. **Before rewriting the DIR60 and TXDLV60 bits, clear the TXE60 and RXE60 bits to 0.**
8. **When the TXDLV60 bit is set to 1 (inverted TxD60 output), the TxD60/P13 pin cannot be used as a general-purpose port, regardless of the settings of POWER60 and TXE60. When using the TxD60/P13 pin as a general-purpose port, clear the TXDLV60 bit to 0 (normal TxD60 output).**

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

(7) Input switch control register (ISC)

The input switch control register (ISC) is used to receive a status signal transmitted from the master when UART60 is used in LIN (Local Interconnect Network) reception.

The signal input from the P14/RxD60 pin is selected as the input source of INTP0 and TI000 when ISC0 and ISC1 are set to 1.

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

Reset signal generation sets this register to 00H.

Figure 14-11. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

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

ISC1	TI000 input source selection
0	TI000 (P00)
1	RxD60 (P14)

ISC0	INTP0 input source selection
0	INTP0 (P120)
1	RxD60 (P14)

(8) Port mode registers 1, 4 (PM1, PM4)

This register sets ports 1, 4 input/output in 1-bit units.

When using the P13/TxD60 and P42/TxD61 pins for serial interface data output, clear PM13 and PM42 to 0 and set the output latch of P13 and P42 to 1.

When using the P14/RxD60 and P43/RxD61 pins for serial interface data input, set PM14 and PM43 to 1. The output latch of P14 and P43 at this time may be 0 or 1.

PM1 and PM4 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 14-12. Format of Port Mode Registers 1, 4 (PM1, PM4)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	PM43 Note	PM42 Note	PM41	PM40

PMmn	Pmn pin I/O mode selection (m = 1, 4; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 78K0/KE2-C only

14.4 Operation of Serial Interfaces UART60 and UART61

Serial interfaces UART60 and UART61 have the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

14.4.1 Operation stop mode

In this mode, serial communication cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER6n, TXE6n, and RXE6n) of ASIM6n to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6n (ASIM6n).

ASIM6n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

Address: FF50H (UART60), FF90H (UART61) After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6n	POWER6n	TXE6n	RXE6n	PS61n	PS60n	CL6n	SL6n	ISRM6n

POWER6n	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE6n	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).

RXE6n	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. If POWER6n = 0 is set while transmitting data, the output of the TxD6n pin will be fixed to high level (if TXDLV6n = 0). Furthermore, the input from the RxD6n pin will be fixed to high level.

2. Asynchronous serial interface reception error status register 6n (ASIS6n), asynchronous serial interface transmission status register 6n (ASIF6n), bit 7 (SBRF6n) and bit 6 (SBRT6n) of asynchronous serial interface control register 6n (ASICL6n), and receive buffer register 6n (RXB6n) are reset.

Caution Clear POWER6n to 0 after clearing TXE6n and RXE6n to 0 to stop the operation. To start the communication, set POWER6n to 1, and then set TXE6n or RXE6n to 1.

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. To use the RxD60/P14, TxD60/P13, RxD61/P43, and TxD61/P42 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

14.4.2 Asynchronous serial interface (UART) mode

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 6n (ASIM6n)
- Asynchronous serial interface reception error status register 6n (ASIS6n)
- Asynchronous serial interface transmission status register 6n (ASIF6n)
- Clock selection register 6n (CKSR6n)
- Baud rate generator control register 6n (BRGC6n)
- Asynchronous serial interface control register 6n (ASICL6n)
- Input switch control register (ISC)
- Port mode registers 1, 4 (PM1, PM4)
- Port registers 1, 4 (P1, P4)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6n register (see **Figure 14-8**).
<2> Set the BRGC6n register (see **Figure 14-9**).
<3> Set bits 0 to 4 (ISRM6n, SL6n, CL6n, PS6n0, PS6n1) of the ASIM6n register (see **Figure 14-5**).
<4> Set bits 0 and 1 (TXDLV6n, DIR6n) of the ASICL6n register (see **Figure 14-10**).
<5> Set bit 7 (POWER6n) of the ASIM6n register to 1.
<6> Set bit 6 (TXE6n) of the ASIM6n register to 1. → Transmission is enabled.
Set bit 5 (RXE6n) of the ASIM6n register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6n (TXB6n). → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

The relationship between the register settings and pins is shown below.

Table 14-2. Relationship Between Register Settings and Pins

(a) Serial interface UART60

POWER60	TXE60	RXE60	PM13	P13	PM14	P14	UART60 Operation	Pin Function	
								TxD60/P13	RxD60/P14
0	0	0	×Note	×Note	×Note	×Note	Stop	P13	P14
1	0	1	×Note	×Note	1	×	Reception	P13	RxD60
	1	0	0	1	×Note	×Note	Transmission	TxD60	P14
	1	1	0	1	1	×	Transmission/ reception	TxD60	RxD60

(b) Serial interface UART61

POWER61	TXE61	RXE61	PM42	P42	PM43	P43	UART61 Operation	Pin Function	
								TxD61/P42	RxD61/P43
0	0	0	×Note	×Note	×Note	×Note	Stop	P42	P43
1	0	1	×Note	×Note	1	×	Reception	P42	RxD61
	1	0	0	1	×Note	×Note	Transmission	TxD61	P43
	1	1	0	1	1	×	Transmission/ reception	TxD61	RxD61

Note Can be set as port function.

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. ×: don't care
POWER6n: Bit 7 of asynchronous serial interface operation mode register 6n (ASIM6n)
TXE6n: Bit 6 of ASIM6n
RXE6n: Bit 5 of ASIM6n
PM××: Port mode register
P××: Port output latch

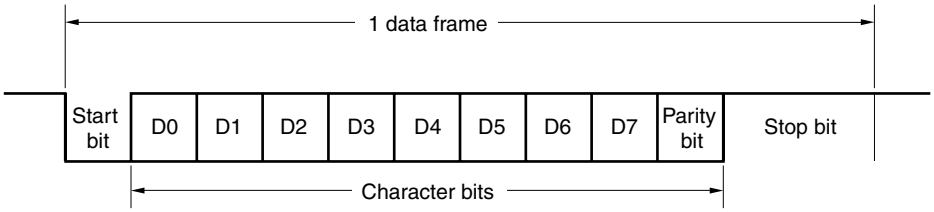
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

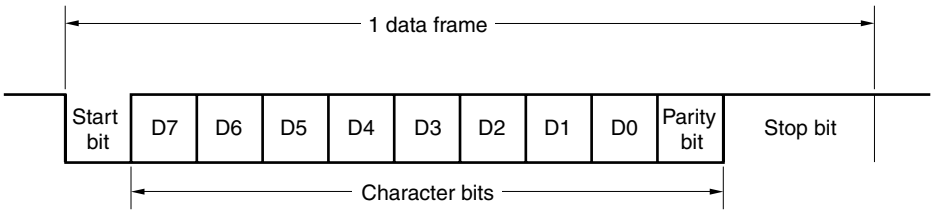
Figures 14-13 and 14-14 show the format and waveform example of the normal transmit/receive data.

Figure 14-13. Format of Normal UART Transmit/Receive Data

1. LSB-first transmission/reception



2. MSB-first transmission/reception



One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6n (ASIM6n).

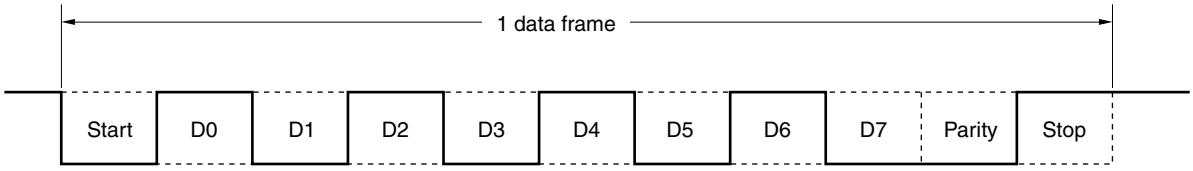
Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6n) of asynchronous serial interface control register 6n (ASICL6n).

Whether the TxD6n pin outputs normal or inverted data is specified by bit 0 (TXDLV6n) of ASICL6n.

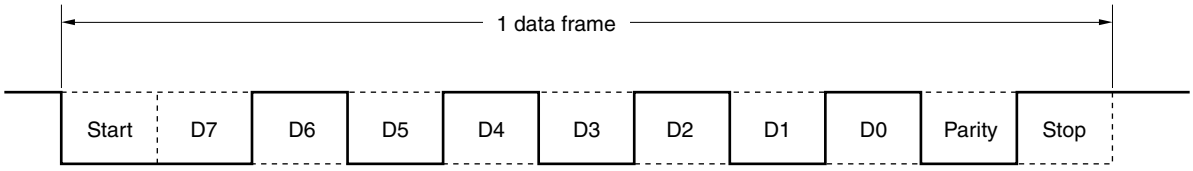
Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

Figure 14-14. Example of Normal UART Transmit/Receive Data Waveform

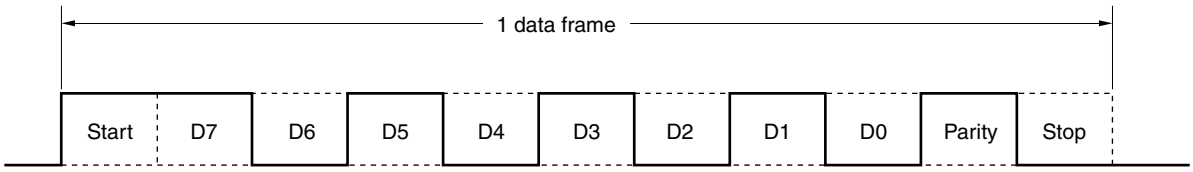
1. **Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



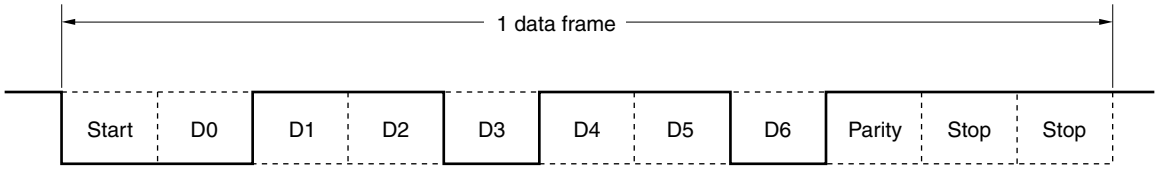
2. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



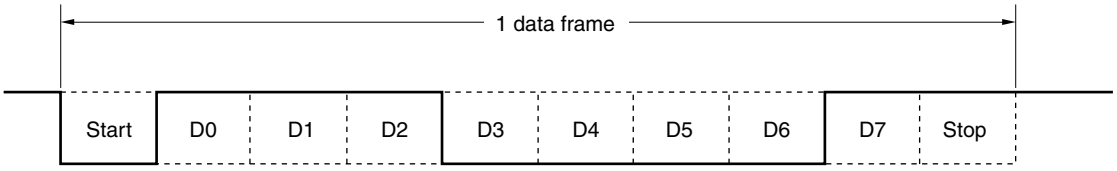
3. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6n pin inverted output**



4. **Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H**



5. **Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Communication data: 87H**



Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

Caution Fix the PS601 and PS600 bits to 0 when UART60 is used in LIN communication operation.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Normal transmission

When bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and bit 6 (TXE6n) of ASIM6n is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6n (TXB6n). The start bit, parity bit, and stop bit are automatically appended to the data.

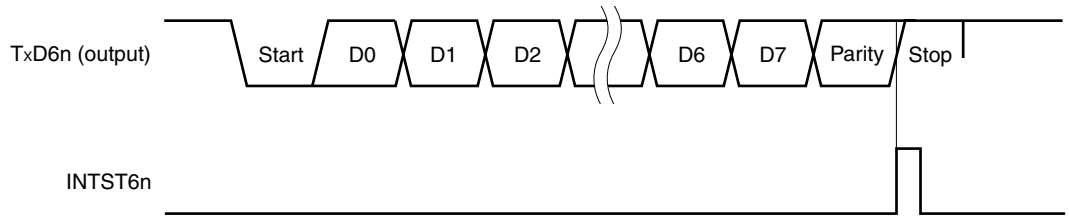
When transmission is started, the data in TXB6n is transferred to transmit shift register 6n (TXS6n). After that, the transmit data is sequentially output from TXS6n to the TxD6n pin. When transmission is completed, the parity and stop bits set by ASIM6n are appended and a transmission completion interrupt request (INTST6n) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6n.

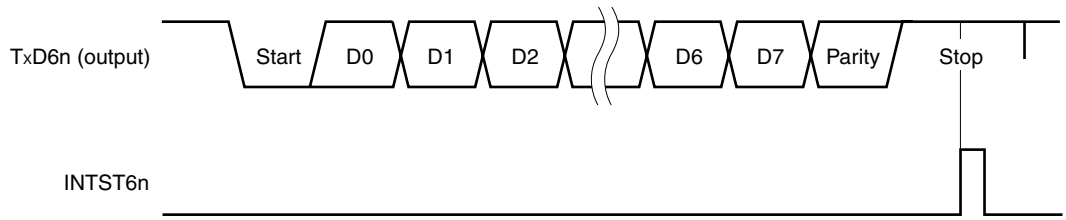
Figure 14-15 shows the timing of the transmission completion interrupt request (INTST6n). This interrupt occurs as soon as the last stop bit has been output.

Figure 14-15. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6n (TXB6n) as soon as transmit shift register 6n (TXS6n) has started its shift operation. Consequently, even while the INTST6n interrupt is being serviced after transmission of one data frame, data can be continuously transmitted and an efficient communication rate can be realized. In addition, the TXB6n register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6n) of asynchronous serial interface transmission status register 6n (ASIF6n) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6n register to check the transmission status and whether the TXB6n register can be written, and then write the data.

Cautions 1. The TXBF6n and TXSF6n flags of the ASIF6n register change from "10" to "11", and to "01" during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6n and TXSF6n flags for judgment. Read only the TXBF6n flag when executing continuous transmission.

2. When UART60 is use in LIN communication operation, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 60 (ASIF60) is 00H before writing transmit data to transmit buffer register 60 (TXB60).

TXBF6n	Writing to TXB6n Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6n register. Be sure to check that the TXBF6n flag is "0". If so, write the next transmit data (second byte) to the TXB6n register. If data is written to the TXB6n register while the TXBF6n flag is "1", the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6n flag.

TXSF6n	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

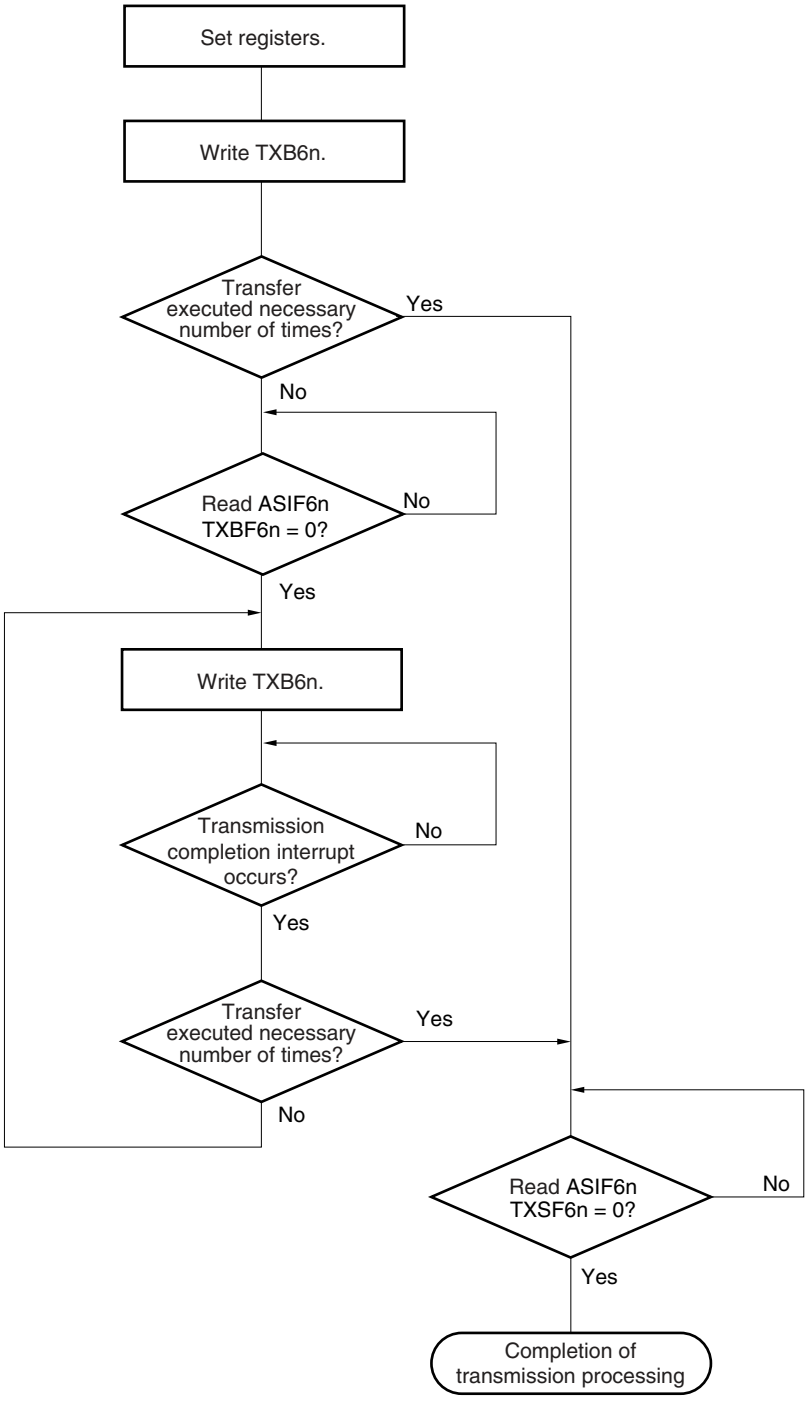
Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6n flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6n flag is "1", the transmit data cannot be guaranteed.

2. During continuous transmission, the next transmission may complete before execution of INTST6n interrupt servicing after transmission of one data frame. As a countermeasure, detection can be performed by developing a program that can count the number of transmit data and by referencing the TXSF6n flag.

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

Figure 14-16 shows an example of the continuous transmission processing flow.

Figure 14-16. Example of Continuous Transmission Processing Flow

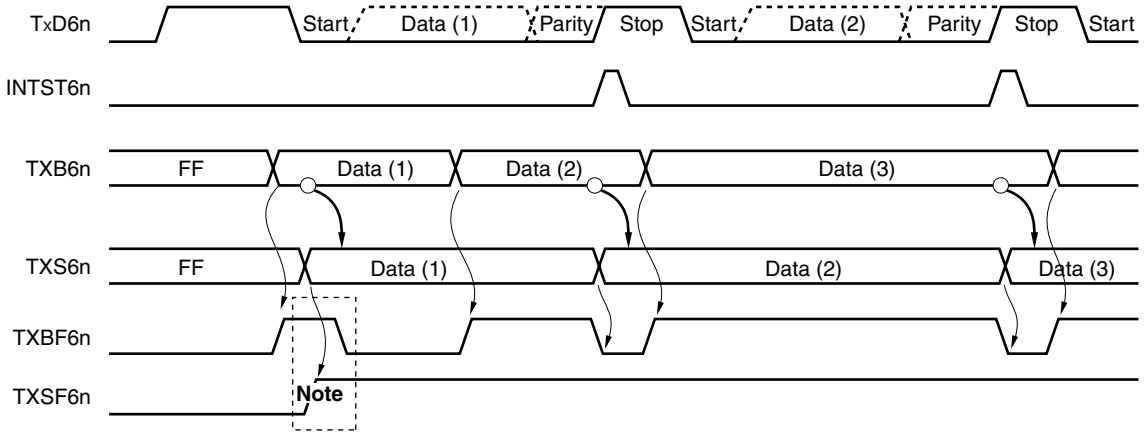


Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. TXB6n: Transmit buffer register 6n
ASIF6n: Asynchronous serial interface transmission status register 6n
TXBF6n: Bit 1 of ASIF6n (transmit buffer data flag)
TXSF6n: Bit 0 of ASIF6n (transmit shift register data flag)

Figure 14-17 shows the timing of starting continuous transmission, and Figure 14-18 shows the timing of ending continuous transmission.

Figure 14-17. Timing of Starting Continuous Transmission

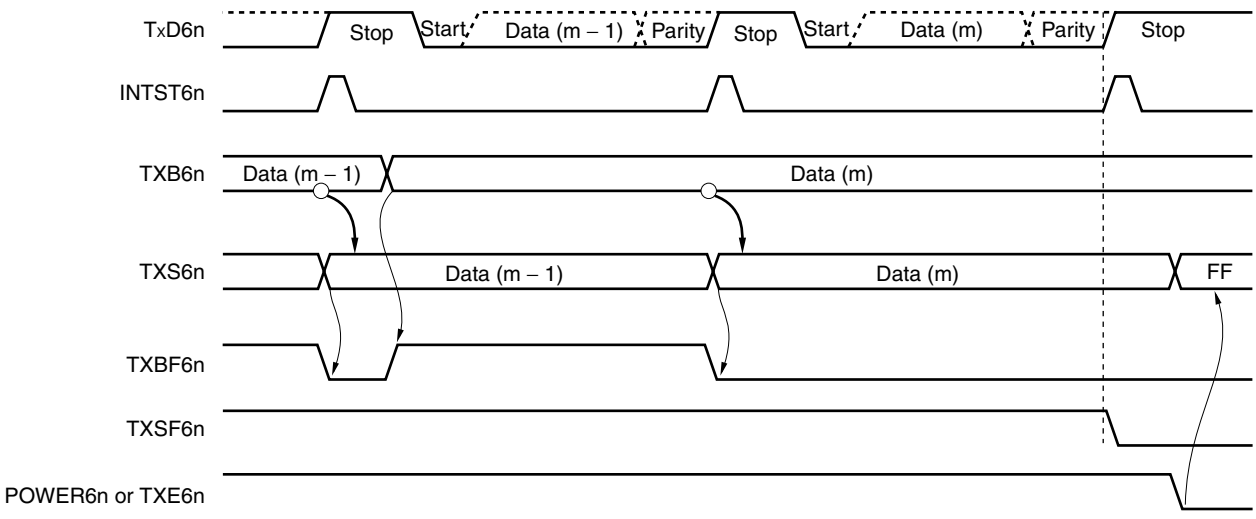


Note When ASIF6n is read, there is a period in which TXBF6n and TXSF6n = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6n bit.

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. TxD6n: TxD6n pin (output)
INTST6n: Interrupt request signal
TXB6n: Transmit buffer register 6n
TXS6n: Transmit shift register 6n
ASIF6n: Asynchronous serial interface transmission status register 6n
TXBF6n: Bit 1 of ASIF6n
TXSF6n: Bit 0 of ASIF6n

Figure 14-18. Timing of Ending Continuous Transmission



Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C
2. TxD6n: TxD6n pin (output)
INTST6n: Interrupt request signal
TXB6n: Transmit buffer register 6n
TXS6n: Transmit shift register 6n
ASIF6n: Asynchronous serial interface transmission status register 6n
TXBF6n: Bit 1 of ASIF6n
TXSF6n: Bit 0 of ASIF6n
POWER6n: Bit 7 of asynchronous serial interface operation mode register (ASIM6n)
TXE6n: Bit 6 of asynchronous serial interface operation mode register (ASIM6n)

(e) Normal reception

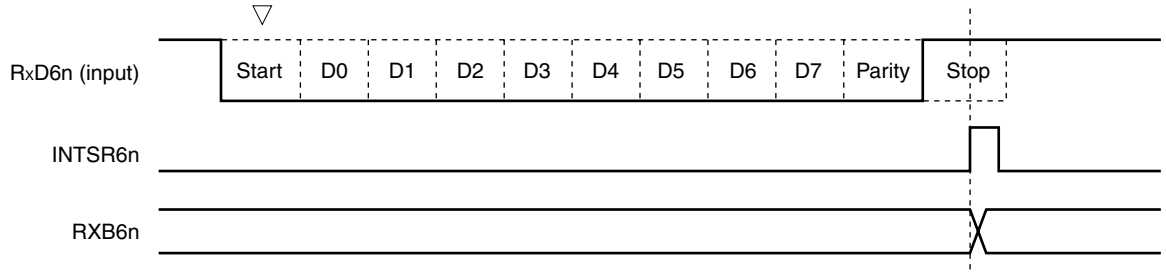
Reception is enabled and the RxD6n pin input is sampled when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and then bit 5 (RXE6n) of ASIM6n is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6n pin input is detected. When the set value of baud rate generator control register 6n (BRGC6n) has been counted, the RxD6n pin input is sampled again (in Figure 14-19). If the RxD6n pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register (RXS6n) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6n) is generated and the data of RXS6n is written to receive buffer register 6n (RXB6n). If an overrun error (OVE6n) occurs, however, the receive data is not written to RXB6n.

Even if a parity error (PE6n) occurs while reception is in progress, reception continues to the reception position of the stop bit, and a reception error interrupt (INTSR6n/INTSRE6n) is generated on completion of reception.

Figure 14-19. Reception Completion Interrupt Request Timing



Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

Cautions 1. If a reception error occurs, read ASIS6n and then RXB6n to clear the error flag. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.
2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.
3. Be sure to read asynchronous serial interface reception error status register 6n (ASIS6n) before reading RXB6n.

(f) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6n (ASIS6n) is set as a result of data reception, a reception error interrupt request (INTSR6n/INTSRE6n) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6n in the reception error interrupt (INTSR6n/INTSRE6n) servicing (see **Figure 14-6**).

The contents of ASIS6n are cleared to 0 when ASIS6n is read.

Table 14-3. Cause of Reception Error

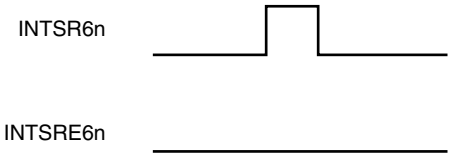
Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6n (RXB6n).

The reception error interrupt can be separated into reception completion interrupt (INTSR6n) and error interrupt (INTSRE6n) by clearing bit 0 (ISRM6n) of asynchronous serial interface operation mode register 6n (ASIM6n) to 0.

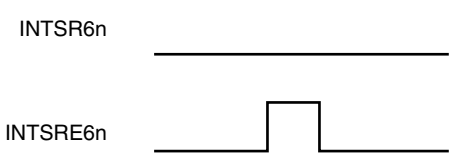
Figure 14-20. Reception Error Interrupt

1. If ISRM6n is cleared to 0 (reception completion interrupt (INTSR6n) and error interrupt (INTSRE6n) are separated)

(a) No error during reception

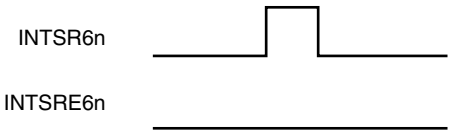


(b) Error during reception

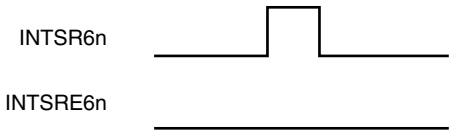


2. If ISRM6n is set to 1 (error interrupt is included in INTSR6n)

(a) No error during reception



(b) Error during reception



Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

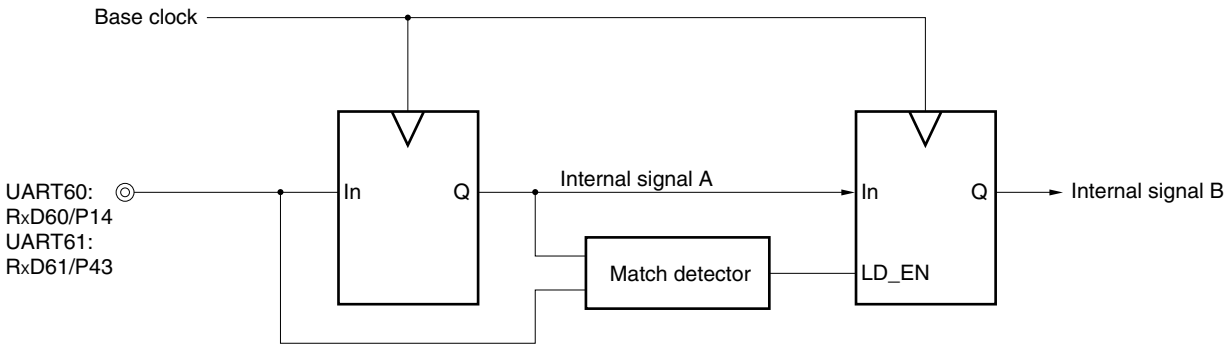
(g) Noise filter of receive data

The RxD6n signal is sampled with the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 14-21, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 14-21. Noise Filter Circuit



(h) SBF transmission

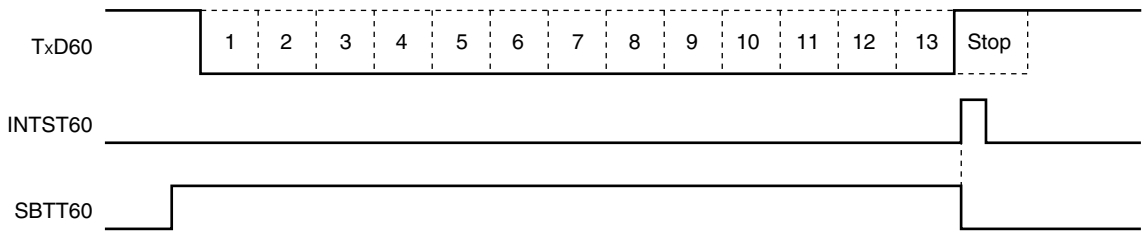
When UART60 is use in LIN communication operation, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, see **Figure 14-1 LIN Transmission Operation**.

When bit 7 (POWER60) of asynchronous serial interface mode register 60 (ASIM60) is set to 1, the TxD60 pin outputs high level. Next, when bit 6 (TXE60) of ASIM60 is set to 1, the transmission enabled status is entered, and SBF transmission is started by setting bit 5 (SBTT60) of asynchronous serial interface control register 60 (ASICL60) to 1.

Thereafter, a low level of bits 13 to 20 (set by bits 4 to 2 (SBL602 to SBL600) of ASICL60) is output. Following the end of SBF transmission, the transmission completion interrupt request (INTST60) is generated and SBTT60 is automatically cleared. Thereafter, the normal transmission mode is restored.

Transmission is suspended until the data to be transmitted next is written to transmit buffer register 60 (TXB60), or until SBTT60 is set to 1.

Figure 14-22. SBF Transmission



Remarks 1. n = 0: 78K0/KC2-C

n = 0, 1: 78K0/KE2-C

2. TxD60: TxD60 pin (output)

INTST60: Transmission completion interrupt request

SBTT60: Bit 5 of asynchronous serial interface control register 60 (ASICL60)

(i) SBF reception

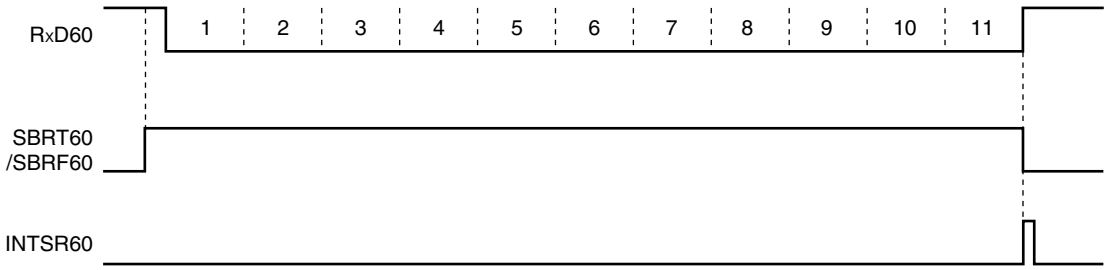
When UART60 is used in LIN communication operation, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, see **Figure 14-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER60) of asynchronous serial interface operation mode register 60 (ASIM60) is set to 1 and then bit 5 (RXE60) of ASIM60 is set to 1. SBF reception is enabled when bit 6 (SBRT60) of asynchronous serial interface control register 60 (ASICL60) is set to 1. In the SBF reception enabled status, the RxD60 pin is sampled and the start bit is detected in the same manner as the normal reception enable status.

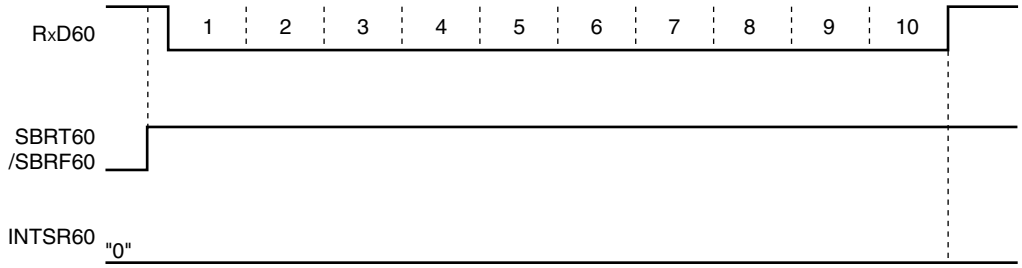
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 60 (RXS60) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR60) is generated as normal processing. At this time, the SBRF60 and SBRT60 bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE60, PE60, and FE60 (bits 0 to 2 of asynchronous serial interface reception error status register 60 (ASIS60)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 60 (RXS60) and receive buffer register 60 (RXB60) is not performed, and the reset value of FFH is retained. If the width of SBF is 10 bits or less, an interrupt does not occur as error processing after the stop bit has been received, and the SBF reception mode is restored. In this case, the SBRF60 and SBRT60 bits are not cleared.

Figure 14-23. SBF Reception

1. Normal SBF reception (stop bit is detected with a width of more than 10.5 bits)



2. SBF reception error (stop bit is detected with a width of 10.5 bits or less)



Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

2. RxD60: RxD60 pin (input)
SBRT60: Bit 6 of asynchronous serial interface control register 60 (ASICL60)
SBRF60: Bit 7 of ASICL60
INTSR60: Reception completion interrupt request

14.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 8-bit programmable counter, and generates a serial clock for transmission/reception of UART6n.

Separate 8-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

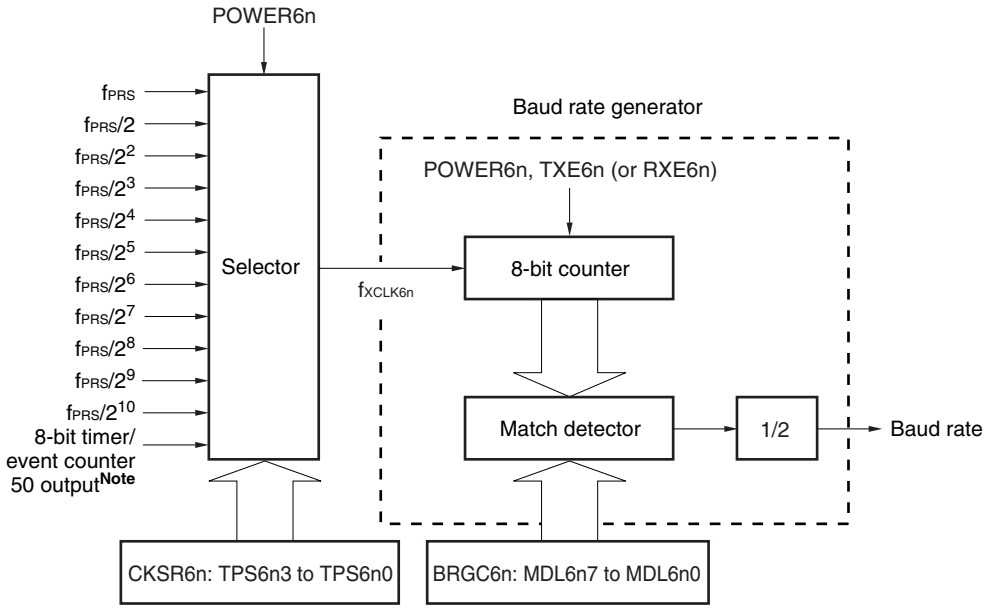
- Base clock
  The clock selected by bits 3 to 0 (TPS6n3 to TPS6n0) of clock selection register 6n (CKSR6n) is supplied to each module when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 1. This clock is called the base clock and its frequency is called f_{XCLK6n}. The base clock is fixed to low level when POWER6n = 0.

- Transmission counter
  This counter stops operation, cleared to 0, when bit 7 (POWER6n) or bit 6 (TXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 0.
  It starts counting when POWER6n = 1 and TXE6n = 1.
  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6n (TXB6n).
  If data are continuously transmitted, the counter is cleared to 0 again when one frame of data has been completely transmitted. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until POWER6n or TXE6n is cleared to 0.

- Reception counter
  This counter stops operation, cleared to 0, when bit 7 (POWER6n) or bit 5 (RXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 0.
  It starts counting when the start bit has been detected.
  The counter stops operation after one frame has been received, until the next start bit is detected.

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

Figure 14-24. Configuration of Baud Rate Generator



Note UART60 only

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C
2. POWER6n: Bit 7 of asynchronous serial interface operation mode register 6n (ASIM6n)
TXE6n: Bit 6 of ASIM6n
RXE6n: Bit 5 of ASIM6n
CKSR6n: Clock selection register 6n
BRGC6n: Baud rate generator control register 6n

(2) Generation of serial clock

A serial clock to be generated can be specified by using clock selection register 6n (CKSR6n) and baud rate generator control register 6n (BRGC6n).

The clock to be input to the 8-bit counter can be set by bits 3 to 0 (TPS6n3 to TPS6n0) of CKSR6n and the division value ($f_{XCLK6n}/4$ to $f_{XCLK6n}/255$) of the 8-bit counter can be set by bits 7 to 0 (MDL6n7 to MDL6n0) of BRGC6n.

14.4.4 Calculation of baud rate

(1) Baud rate calculation expression

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK6n}}{2 \times k}$ [bps]

f_{XCLK6n}: Frequency of base clock selected by TPS6n3 to TPS6n0 bits of CKSR6n register
k: Value set by MDL6n7 to MDL6n0 bits of BRGC6n register (k = 4, 5, 6, ..., 255)

Table 14-4. Set Value of TPS6n3 to TPS6n0

TPS6n3	TPS6n2	TPS6n1	TPS6n0	Base Clock (f_{XCLK6n}) Selection				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output[Notes 1, 2]				
Other than above				Setting prohibited				

Notes 1. UART60 only

2. Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)

  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)

  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

(2) Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 10 MHz = 10,000,000 Hz
Set value of MDL6n7 to MDL6n0 bits of BRGC6n register = 00100001B (k = 33)
Target baud rate = 153600 bps

Baud rate = 10 M / (2 × 33)
= 10000000 / (2 × 33) = 151,515 [bps]

Error = (151515/153600 − 1) × 100
= −1.357 [%]

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

(3) Example of setting baud rate

Table 14-5. Set Data of Baud Rate Generator

Baud Rate [bps]	f_{PRS} = 2.0 MHz				f_{PRS} = 5.0 MHz				f_{PRS} = 10.0 MHz				f_{PRS} = 20.0 MHz			
	TPS6n3 to TPS6n0	k	Calculated Value	ERR [%]	TPS6n3 to TPS6n0	k	Calculated Value	ERR [%]	TPS6n3 to TPS6n0	k	Calculated Value	ERR [%]	TPS6n3 to TPS6n0	k	Calculated Value	ERR [%]
300	8H	13	301	0.16	7H	65	301	0.16	8H	65	301	0.16	9H	65	301	0.16
600	7H	13	601	0.16	6H	65	601	0.16	7H	65	601	0.16	8H	65	601	0.16
1200	6H	13	1202	0.16	5H	65	1202	0.16	6H	65	1202	0.16	7H	65	1202	0.16
2400	5H	13	2404	0.16	4H	65	2404	0.16	5H	65	2404	0.16	6H	65	2404	0.16
4800	4H	13	4808	0.16	3H	65	4808	0.16	4H	65	4808	0.16	5H	65	4808	0.16
9600	3H	13	9615	0.16	2H	65	9615	0.16	3H	65	9615	0.16	4H	65	9615	0.16
19200	2H	13	19231	0.16	1H	65	19231	0.16	2H	65	19231	0.16	3H	65	19231	0.16
24000	1H	21	23810	–0.79	3H	13	24038	0.16	4H	13	24038	0.16	5H	13	24038	0.16
31250	1H	16	31250	0	4H	5	31250	0	5H	5	31250	0	6H	5	31250	0
38400	1H	13	38462	0.16	0H	65	38462	0.16	1H	65	38462	0.16	2H	65	38462	0.16
48000	0H	21	47619	–0.79	2H	13	48077	0.16	3H	13	48077	0.16	4H	13	48077	0.16
76800	0H	13	76923	0.16	0H	33	75758	–1.36	0H	65	76923	0.16	1H	65	76923	0.16
115200	0H	9	111111	–3.55	1H	11	113636	–1.36	0H	43	116279	0.94	0H	87	114943	–0.22
153600	–	–	–	–	1H	8	156250	1.73	0H	33	151515	–1.36	1H	33	151515	–1.36
312500	–	–	–	–	0H	8	312500	0	1H	8	312500	0	2H	8	312500	0
625000	–	–	–	–	0H	4	625000	0	1H	4	625000	0	2H	4	625000	0

Remarks 1. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

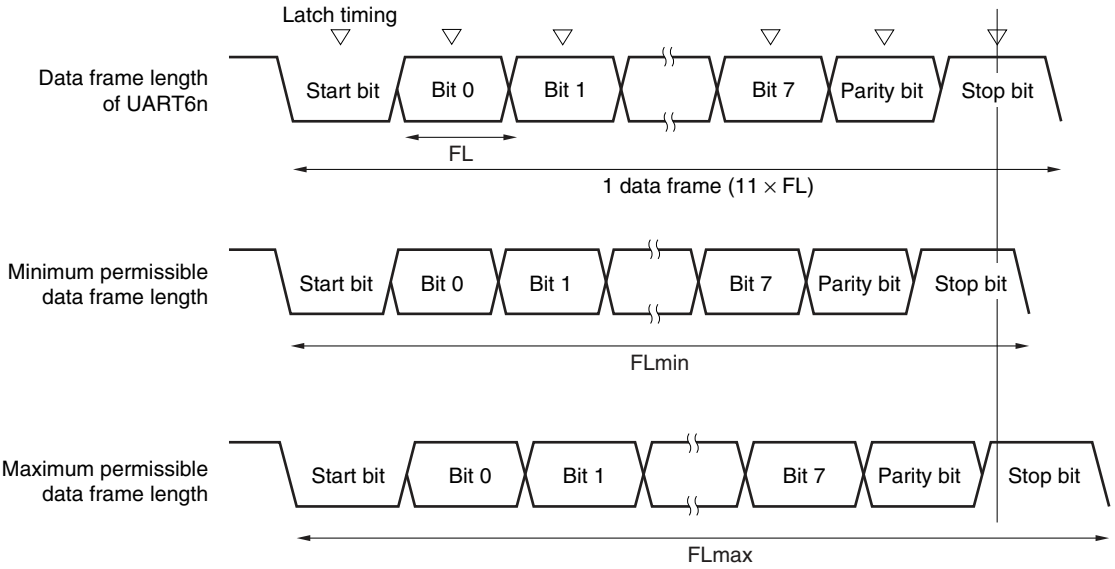
2. TPS6n3 to TPS6n0: Bits 3 to 0 of clock selection register 6n (CKSR6n) (setting of base clock (f_{XCLK6n}))
k: Value set by MDL6n7 to MDL6n0 bits of baud rate generator control register 6n (BRGC6n) (k = 4, 5, 6, ..., 255)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 14-25. Permissible Baud Rate Range During Reception



As shown in Figure 14-25, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6n (BRGC6n) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART6n
k: Set value of BRGC6n
FL: 1-bit data length
Margin of latch timing: 2 clocks

Remark n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

$$\text{Minimum permissible data frame length: FLmin} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

The permissible baud rate error between UART6n and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 14-6. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
4	+2.33%	–2.44%
8	+3.53%	–3.61%
20	+4.26%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

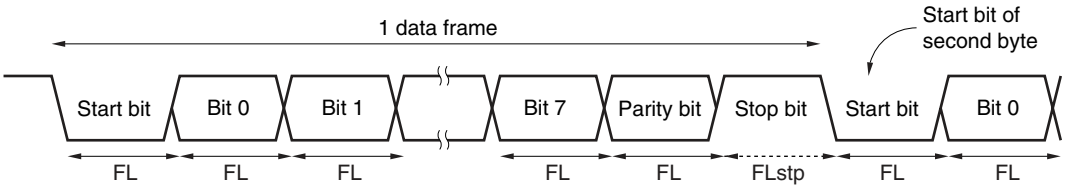
2. n = 0: 78K0/KC2-C
n = 0, 1: 78K0/KE2-C

3. k: Set value of BRGC6n

(5) Data frame length during continuous transmission

When data is continuously transmitted, the data frame length from a stop bit to the next start bit is extended by two clocks of base clock from the normal value. However, the result of communication is not affected because the timing is initialized on the reception side when the start bit is detected.

Figure 14-26. Data Frame Length During Continuous Transmission



Where the 1-bit data length is FL, the stop bit length is FLstp, and base clock frequency is f_{XCLK6n}, the following expression is satisfied.

$$FLstp = FL + 2/f_{XCLK6n}$$

Therefore, the data frame length during continuous transmission is:

$$\text{Data frame length} = 11 \times FL + 2/f_{XCLK6n}$$

CHAPTER 15 SERIAL INTERFACES CSI10 AND CSI11

	78K0/KC2-C (μPD78F0760, 78F0761, 78F0762)	78K0/KE2-C (μPD78F0763, 78F0764, 78F0765)
Serial interface CSI10	√	
Serial interface CSI11	√ Note 1	√ Note 2

Notes 1. The CSI11 of 78K0/KC2-C is not provided with the chip select input pin ($\overline{\text{SSI11}}$).

2. Only CSI11 of the 78K0/KE2-C supports SPI.

Caution If CSI10 is used, UART0 cannot be used.

Remark √: Mounted

15.1 Functions of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 have the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not performed and can enable a reduction in the power consumption.

For details, see **15.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to communicate 8-bit data using three lines: a serial clock line ($\overline{\text{SCK1n}}$) and two serial data lines (SI1n and SO1n).

The processing time of data communication can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

For details, see **15.4.2 3-wire serial I/O mode**.

Remark n = 0, 1

15.2 Configuration of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 include the following hardware.

Table 15-1. Configuration of Serial Interfaces CSI10 and CSI11

Item	Configuration
Controller	Transmit controller Clock start/stop controller & clock phase controller
Registers	Transmit buffer register 1n (SOTB1n) Serial I/O shift register 1n (SIO1n)
Control registers	Serial operation mode register 1n (CSIM1n) Serial clock selection register 1n (CSIC1n) Port mode register 1 (PM1) or port mode register 7 (PM7) Port register 1 (P1) or port register 7 (P7)

Remark n = 0, 1

Figure 15-1. Block Diagram of Serial Interface CSI10

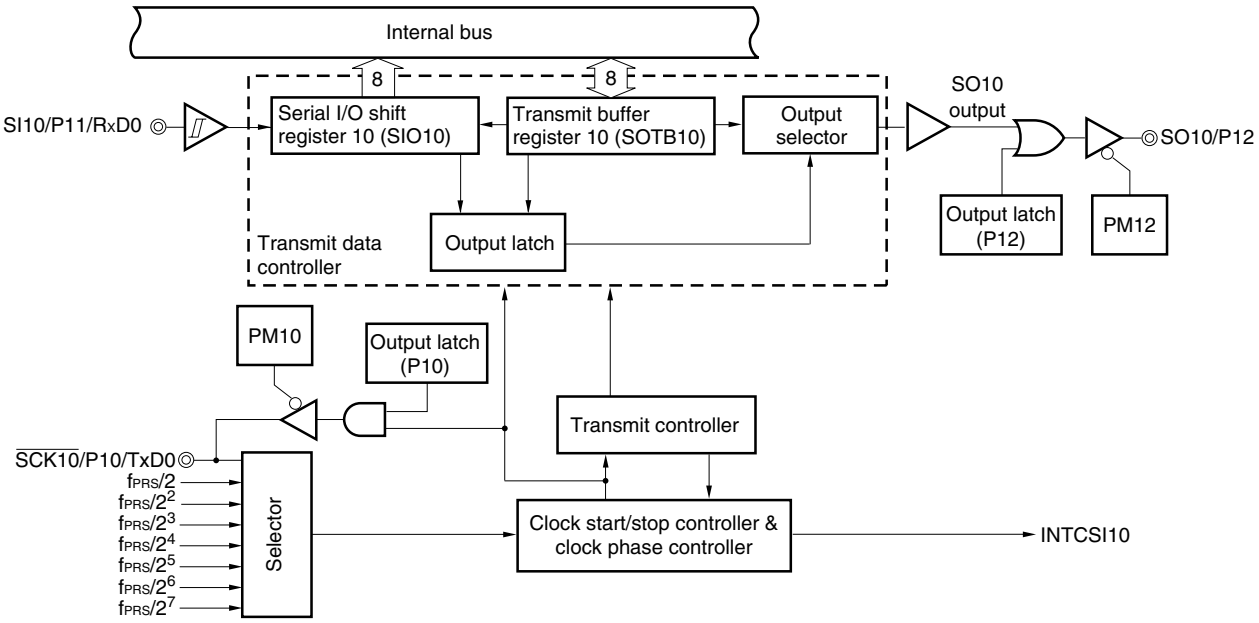


Figure 15-2. Block Diagram of Serial Interface CSI11 (78K0/KC2-C)

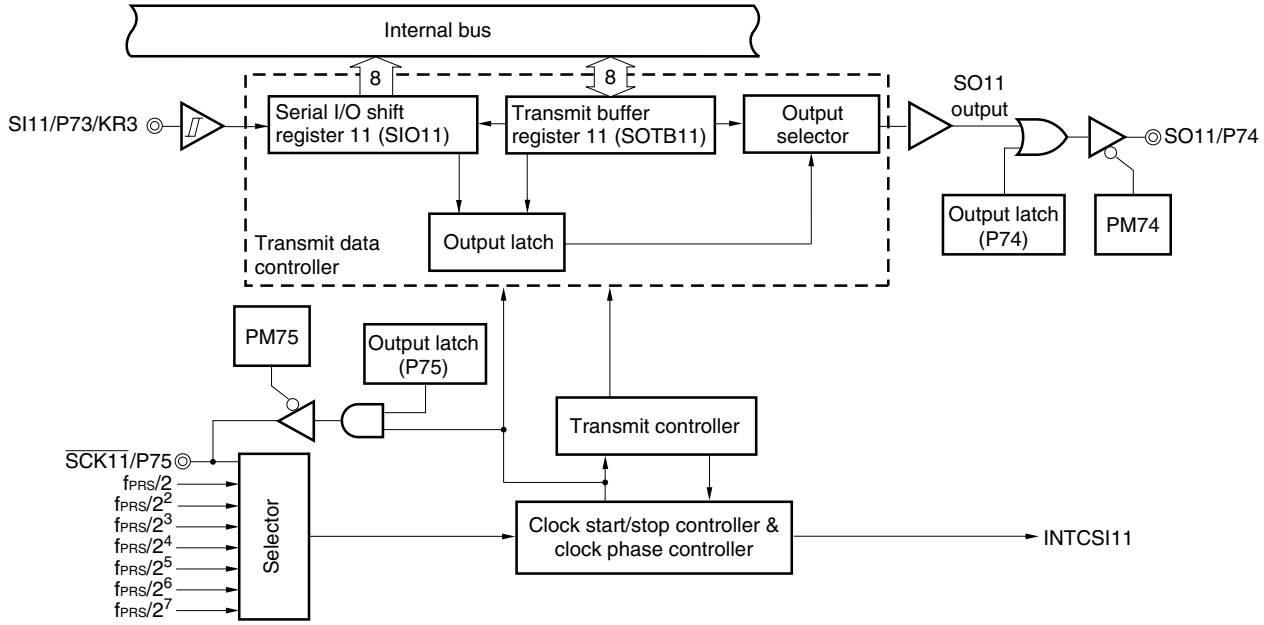
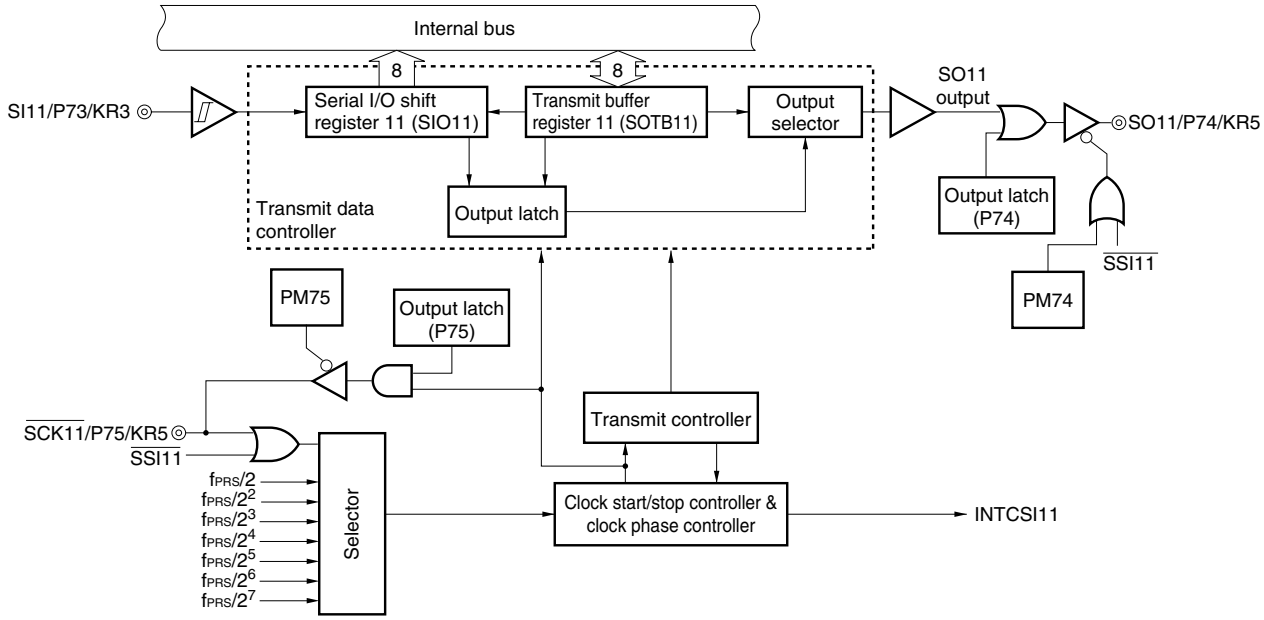


Figure 15-3. Block Diagram of Serial Interface CSI11 (78K0/KE2-C)



(1) Transmit buffer register 1n (SOTB1n)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB1n when bit 7 (CSIE1n) and bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1.

The data written to SOTB1n is converted from parallel data into serial data by serial I/O shift register 1n, and output to the serial output pin (SO1n).

SOTB1n can be written or read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SOTB1n when CSOT1n = 1 (during serial communication).

2. In the slave mode of the 78K0/KE2-C, transmission/reception is started when data is written to SOTB11 with a low level input to the $\overline{\text{SSI11}}$ pin. For details on the transmission/reception operation, see 15.4.2 (2) Communication operation.

(2) Serial I/O shift register 1n (SIO1n)

This is an 8-bit register that converts data from parallel data into serial data and vice versa.

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

Reception is started by reading data from SIO1n if bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

During reception, the data is read from the serial input pin (SI1n) to SIO1n.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SIO1n when CSOT1n = 1 (during serial communication).

2. In the slave mode of the 78K0/KE2-C, reception is started when data is read from SIO11 with a low level input to the $\overline{\text{SSI11}}$ pin. For details on the reception operation, see 15.4.2 (2) Communication operation.

Remark n = 0, 1

15.3 Registers Controlling Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 are controlled by the following four registers.

- Serial operation mode register 1n (CSIM1n)
- Serial clock selection register 1n (CSIC1n)
- Port mode registers (PM1, PM7)
- Port registers (P1, P7)

(1) Serial operation mode register 1n (CSIM1n)

CSIM1n is used to select the operation mode and enable or disable operation.

CSIM1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 15-4. Format of Serial Operation Mode Register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 2] and asynchronously resets the internal circuit[Note 3].
1	Enables operation

TRMD10[Note 4]	Transmit/receive mode control
0[Note 5]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10[Note 6]	First bit specification
0	MSB
1	LSB

CSOT10	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. To use P10/$\overline{\text{SCK10}}$/TxD0 and P12/SO10 as general-purpose ports, set CSIM10 in the default status (00H).

3. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

4. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

5. The SO10 output (see **Figure 15-1**) is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

6. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

Caution Be sure to clear bit 5 to 0.

Figure 15-5. Format of Serial Operation Mode Register 11 (CSIM11)

Address: FF88H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11[Note 2]	DIR11	0	0	0	CSOT11

CSIE11	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 3] and asynchronously resets the internal circuit[Note 4].
1	Enables operation

TRMD11[Note 5]	Transmit/receive mode control
0[Note 6]	Receive mode (transmission disabled).
1	Transmit/receive mode

SSE11[Notes 2, 7, 8]	$\overline{\text{SSI11}}$ pin use selection
0	$\overline{\text{SSI11}}$ pin is not used
1	$\overline{\text{SSI11}}$ pin is used

DIR11[Note 9]	First bit specification
0	MSB
1	LSB

CSOT11	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. 78K0/KE2-C only

3. To use following pins as general-purpose ports, set CSIM11 in the default status (00H).

- 78K0/KC2-C: P74/SO11, P75/$\overline{\text{SCK11}}$
- 78K0/KE2-C: P74/SO11/KR4, P75/$\overline{\text{SCK11}}$/KR5, P76/$\overline{\text{SSI11}}$/KR6

4. Bit 0 (CSOT11) of CSIM11 and serial I/O shift register 11 (SIO11) are reset.

5. Do not rewrite TRMD11 when CSOT11 = 1 (during serial communication).

6. The SO11 output (see **Figure 15-2**) is fixed to the low level when TRMD11 is 0. Reception is started when data is read from SIO11.

7. Do not rewrite SSE11 when CSOT11 = 1 (during serial communication).

8. Before setting this bit to 1, fix the $\overline{\text{SSI11}}$ pin input level to 0 or 1.

9. Do not rewrite DIR11 when CSOT11 = 1 (during serial communication).

(2) Serial clock selection register 1n (CSIC1n)

This register specifies the timing of the data transmission/reception and sets the serial clock.

CSIC1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-6. Format of Serial Clock Selection Register 10 (CSIC10)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	Specification of data transmission/reception timing	Type
0	0	$\overline{SCK10}$, SO10 (D7 D6 D5 D4 D3 D2 D1 D0), SI10 input timing	1
0	1	$\overline{SCK10}$, SO10 (D7 D6 D5 D4 D3 D2 D1 D0), SI10 input timing	2
1	0	$\overline{SCK10}$, SO10 (D7 D6 D5 D4 D3 D2 D1 D0), SI10 input timing	3
1	1	$\overline{SCK10}$, SO10 (D7 D6 D5 D4 D3 D2 D1 D0), SI10 input timing	4

CKS102	CKS101	CKS100	CSI10 serial clock selection[Note 1]				Mode
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	Setting prohibited	Master mode
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	15.63 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	External clock input from $\overline{SCK10}$[Note 2]					Slave mode

Notes 1. Set the serial clock to satisfy 6.25 MHz or less.

2. Do not start communication with the external clock from the $\overline{SCK10}$ pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Cautions 1. Do not write to CSIC10 while CSIE10 = 1 (operation enabled).

2. To use P10/$\overline{SCK10}$/TxD0 and P12/SO10 as general-purpose ports, set CSIC10 in the default status (00H).

Caution 3. The phase type of the data clock is type 1 after reset.

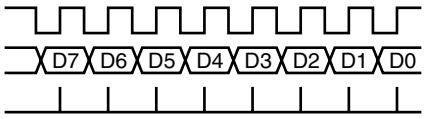
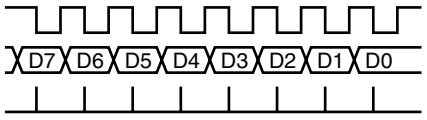
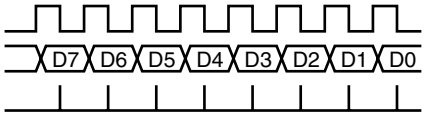
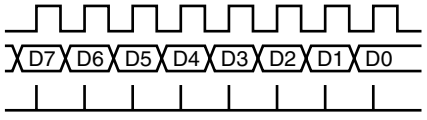
Remarks 1. n = 0, 1

2. f_{PRS}: Peripheral hardware clock frequency

Figure 15-7. Format of Serial Clock Selection Register 11 (CSIC11)

Address: FF89H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC11	0	0	0	CKP11	DAP11	CKS112	CKS111	CKS110

CKP11	DAP11	Specification of data transmission/reception timing	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS112	CKS111	CKS110	CSI11 serial clock selection[Notes 1]					Mode
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	Setting prohibited	Master mode
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	15.63 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	External clock input from $\overline{SCK11}$[Note 2]					Slave mode

Notes 1. Set the serial clock to satisfy 6.25 MHz or less.

2. Do not start communication with the external clock from the $\overline{SCK11}$ pin when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

Cautions 1. Do not write to CSIC11 while CSIE11 = 1 (operation enabled).

2. To use following pins as general-purpose ports, set CSIC11 in the default status (00H).

- 78K0/KC2-C: P74/SO11, P75/$\overline{\text{SCK11}}$
- 78K0/KE2-C: P74/SO11/KR4, P75/$\overline{\text{SCK11}}$/KR5

3. The phase type of the data clock is type 1 after reset.

Remark f_{PRS}: Peripheral hardware clock frequency

(3) Port mode registers 1 and 7 (PM1, PM7)

These registers set port 1 and 7 input/output in 1-bit units.

When using following pins as the clock output pins of the serial interface, clear the corresponding port mode register (PMxx) to 0, and set the output latches of Pxx to 1.

- 78K0/KC2-C: P10/$\overline{\text{SCK10}}$, P75/$\overline{\text{SCK11}}$
- 78K0/KE2-C: P10/$\overline{\text{SCK10}}$, P75/$\overline{\text{SCK11}}$/KR5

When using following pins as the data output pins of the serial interface, clear the corresponding port mode register (PMxx), and the output latches of Pxx to 0.

- 78K0/KC2-C: P12/SO10, P74/SO11
- 78K0/KE2-C: P12/SO10, P74/SO11/KR4

When using following pins as the clock input pins of the serial interface, data input pins, and chip select input pin, set corresponding port mode register (PMxx) to 1. At this time, the output latches of Pxx may be 0 or 1.

- 78K0/KC2-C: P10/$\overline{\text{SCK10}}$, P75/$\overline{\text{SCK11}}$, P11/SI10/RxD0, P73/SI11/KR3
- 78K0/KE2-C: P10/$\overline{\text{SCK10}}$, P75/$\overline{\text{SCK11}}$/KR5, P11/SI10/RxD0, P73/SI11/KR3, P76/$\overline{\text{SSI11}}$/KR6

The port mode register (PMxx) can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 15-8. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 15-9. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 78K0/KE2-C only

15.4 Operation of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

15.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, P12/SO10, P74/SO11/KR4[Note], P73/SI11/KR3, and P75/$\overline{\text{SCK11}}$/KR5[Note] pins can be used as ordinary I/O port pins in this mode.

Note The 78K0/KC2-C is not provided with the KR4 and KR5 pins.

(1) Register used

The operation stop mode is set by serial operation mode register 1n (CSIM1n).

To set the operation stop mode, clear bit 7 (CSIE1n) of CSIM1n to 0.

(a) Serial operation mode register 1n (CSIM1n)

CSIM1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CSIM1n to 00H.

Remark n = 0, 1

- Serial operation mode register 10 (CSIM10)

Address: FF80H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. To use P10/$\overline{\text{SCK10}}$/TxD0 and P12/SO10 as general-purpose ports, set CSIM10 in the default status (00H).

2. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

- Serial operation mode register 11 (CSIM11)

Address: FF88H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11	DIR11	0	0	0	CSOT11

CSIE11	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. To use following pins as general-purpose ports, set CSIM11 in the default status (00H).
- 78K0/KC2-C: P74/SO11, P75/$\overline{\text{SCK11}}$
- 78K0/KE2-C: P74/SO11/KR4, P75/$\overline{\text{SCK11}}$/KR5, P76/$\overline{\text{SSI11}}$/KR6

2. Bit 0 (CSOT11) of CSIM11 and serial I/O shift register 11 (SIO11) are reset.

15.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

In this mode, communication is executed by using three lines: the serial clock ($\overline{\text{SCK1n}}$), serial output (SO1n), and serial input (SI1n) lines.

(1) Registers used

- Serial operation mode register 1n (CSIM1n)
- Serial clock selection register 1n (CSIC1n)
- Port mode register 1 (PM1) or port mode register 7 (PM7)
- Port register 1 (P1) or port register 7 (P7)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC1n register (see **Figures 15-6** and **15-7**).
<2> Set bits 4 to 6 (DIR1n, SSE11 (serial interface CSI11 only), and TRMD1n) of the CSIM1n register (see **Figures 15-4** and **15-5**).
<3> Set bit 7 (CSIE1n) of the CSIM1n register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 1n (SOTB1n). → Data transmission/reception is started.
Read data from serial I/O shift register 1n (SIO1n). → Data reception is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

Remark n = 0, 1

The relationship between the register settings and pins is shown below.

Table 15-2. Relationship Between Register Settings and Pins (1/3)

(a) Serial interface CSI10

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 Operation	Pin Function		
									SI10/RxD0/P11	SO10/P12	$\overline{\text{SCK10}}$/TxD0/P10
0	0	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	Stop	RxD0/P11	P12Note 2	TxD0/P10Note 3
1	0	1	×	×Note 1	×Note 1	1	×	Slave receptionNote 4	SI10	P12Note 2	$\overline{\text{SCK10}}$ (input)Note 4
1	1	×Note 1	×Note 1	0	0	1	×	Slave transmissionNote 4	RxD0/P11	SO10	$\overline{\text{SCK10}}$ (input)Note 4
1	1	1	×	0	0	1	×	Slave transmission/ receptionNote 4	SI10	SO10	$\overline{\text{SCK10}}$ (input)Note 4
1	0	1	×	×Note 1	×Note 1	0	1	Master reception	SI10	P12Note 2	$\overline{\text{SCK10}}$ (output)
1	1	×Note 1	×Note 1	0	0	0	1	Master transmission	RxD0/P11	SO10	$\overline{\text{SCK10}}$ (output)
1	1	1	×	0	0	0	1	Master transmission/ reception	SI10	SO10	$\overline{\text{SCK10}}$ (output)

Notes 1. Can be set as port function.

2. To use P12/SO10 as general-purpose port, set the serial clock selection register 10 (CSIC10) in the default status (00H).

3. To use P10/$\overline{\text{SCK10}}$/TxD0 as port pins, clear CKP10 to 0.

4. To use the slave mode, set CKS102, CKS101, and CKS100 to 1, 1, 1.

Remark ×: don't care
CSIE10: Bit 7 of serial operation mode register 10 (CSIM10)
TRMD10: Bit 6 of CSIM10
CKP10: Bit 4 of serial clock selection register 10 (CSIC10)
CKS102, CKS101, CKS100: Bits 2 to 0 of CSIC10
PM1×: Port mode register
P1×: Port output latch

Table 15-2. Relationship Between Register Settings and Pins (2/3)

(b) Serial interface CSI11 (78K0/KC2-C)

CSIE11	TRMD11	SSE11	PM73	P73	PM74	P74	PM75	P75	CSI11 Operation	Pin Function		
										SI11/ KR3/P73	SO11/P74	$\overline{\text{SCK11}}$/P75
0	0	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	Stop	P73	P74Note 2	P75Note 3
1	0	0	1	×	×Note 1	×Note 1	1	×	Slave reception Note 4	SI11	P74Note 2	$\overline{\text{SCK11}}$ (input) Note 4
		1										
1	1	0	×Note 1	×Note 1	0	0	1	×	Slave transmission Note 4	KR3/P73	SO11	$\overline{\text{SCK11}}$ (input) Note 4
		1										
1	1	0	1	×	0	0	1	×	Slave transmission/ reception Note 4	SI11	SO11	$\overline{\text{SCK11}}$ (input) Note 4
		1										
1	0	0	1	×	×Note 1	×Note 1	0	1	Master reception	SI11	P74Note 2	$\overline{\text{SCK11}}$ (output)
1	1	0	×Note 1	×Note 1	0	0	0	1	Master transmission	KR3/P73	SO11	$\overline{\text{SCK11}}$ (output)
1	1	0	1	×	0	0	0	1	Master transmission/ reception	SI11	SO11	$\overline{\text{SCK11}}$ (output)

Notes 1. Can be set as port function.

2. To use P74/SO11 as general-purpose port, set the serial clock selection register 11 (CSIC11) in the default status (00H).

3 To use P75/$\overline{\text{SCK11}}$ as port pins, clear CKP11 to 0.

4 To use the slave mode, set CKS112, CKS111, and CKS110 to 1, 1, 1.

Remark ×: don't care
CSIE11: Bit 7 of serial operation mode register 11 (CSIM11)
TRMD11: Bit 6 of CSIM11
CKP11: Bit 4 of serial clock selection register 11 (CSIC11)
CKS112, CKS111, CKS110: Bits 2 to 0 of CSIC11
PM7×: Port mode register
P7×: Port output latch

Table 15-2. Relationship Between Register Settings and Pins (3/3)

(c) Serial interface CSI11 (78K0/KE2-C)

CSIE11	TRMD11	SSE11	PM73	P73	PM74	P74	PM75	P75	PM76	P76	CSI11 Operation	Pin Function			
												SI11/ KR3/ P73	SO11/ KR4/ P74	$\overline{\text{SCK11}}$/ KR5/ P75	$\overline{\text{SSI11}}$/ KR6/ P76
0	0	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	Stop	KR3/ P73	KR4/ P74Note 2	KR5/ P75	KR6/ P76Note 3
1	0	0	1	×	×Note 1	×Note 1	1	×	×Note 1	×Note 1	Slave reception Note 4	SI11	KR4/ P74Note 2	$\overline{\text{SCK11}}$ (input) Note 4	KR6/ P76
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	×Note 1	×Note 1	0	0	1	×	×Note 1	×Note 1	Slave transmission Note 4	KR3/ P73	SO11	$\overline{\text{SCK11}}$ (input) Note 4	KR6/ P76
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	1	×	0	0	1	×	×Note 1	×Note 1	Slave transmission/ reception Note 4	SI11	SO11	$\overline{\text{SCK11}}$ (input) Note 4	KR6/ P76
		1							1	×					$\overline{\text{SSI11}}$
1	0	0	1	×	×Note 1	×Note 1	0	1	×Note 1	×Note 1	Master reception	SI11	KR4/ P74Note 2	$\overline{\text{SCK11}}$ (output)	KR6/ P76
1	1	0	×Note 1	×Note 1	0	0	0	1	×Note 1	×Note 1	Master transmission	KR3/ P73	SO11	$\overline{\text{SCK11}}$ (output)	KR6/ P76
1	1	0	1	×	0	0	0	1	×Note 1	×Note 1	Master transmission/ reception	SI11	SO11	$\overline{\text{SCK11}}$ (output)	KR6/ P76

Notes 1. Can be set as port function.

2. To use P74/SO11/KR4 as general-purpose port, set the serial clock selection register 11 (CSIC11) in the default status (00H).

3 To use P75/$\overline{\text{SCK11}}$/KR5 as port pins, clear CKP11 to 0.

4 To use the slave mode, set CKS112, CKS111, and CKS110 to 1, 1, 1.

Remark ×: don't care
CSIE11: Bit 7 of serial operation mode register 11 (CSIM11)
TRMD11: Bit 6 of CSIM11
CKP11: Bit 4 of serial clock selection register 11 (CSIC11)
CKS112, CKS111, CKS110: Bits 2 to 0 of CSIC11
PM7×: Port mode register
P7×: Port output latch

(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1. Transmission/reception is started when a value is written to transmit buffer register 1n (SOTB1n). In addition, data can be received when bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

Reception is started when data is read from serial I/O shift register 1n (SIO1n).

However, communication is performed as follows if bit 5 (SSE11) of CSIM11 is 1 when serial interface CSI11 of the 78K0/KE2-C is in the slave mode.

<1> Low level input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception is started when SOTB11 is written, or reception is started when SIO11 is read.

<2> High level input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception or reception is held, therefore, even if SOTB11 is written or SIO11 is read, transmission/reception or reception will not be started.

<3> Data is written to SOTB11 or data is read from SIO11 while a high level is input to the $\overline{\text{SSI11}}$ pin, then a low level is input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception or reception is started.

<4> A high level is input to the $\overline{\text{SSI11}}$ pin during transmission/reception or reception
→ Transmission/reception or reception is suspended.

After communication has been started, bit 0 (CSOT1n) of CSIM1n is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF1n) is set, and CSOT1n is cleared to 0. Then the next communication is enabled.

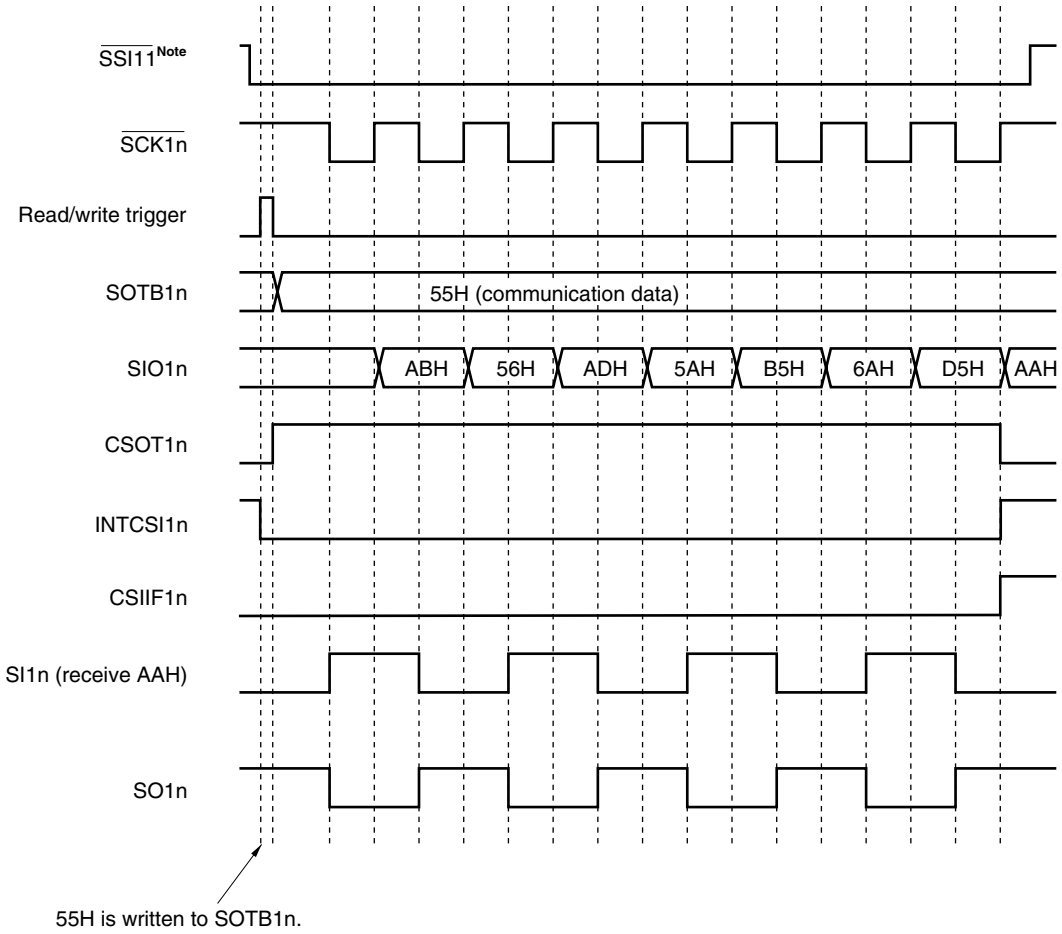
Cautions 1. Do not access the control register and data register when CSOT1n = 1 (during serial communication).

2. When using serial interface CSI11 of the 78K0/KE2-C, wait for the duration of at least one clock before the clock operation is started to change the level of the $\overline{\text{SSI11}}$ pin in the slave mode; otherwise, malfunctioning may occur.

Remark n = 0, 1

Figure 15-10. Timing in 3-Wire Serial I/O Mode (1/2)

(a) Transmission/reception timing (Type 1: TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 0, SSE11 = 1[Note])

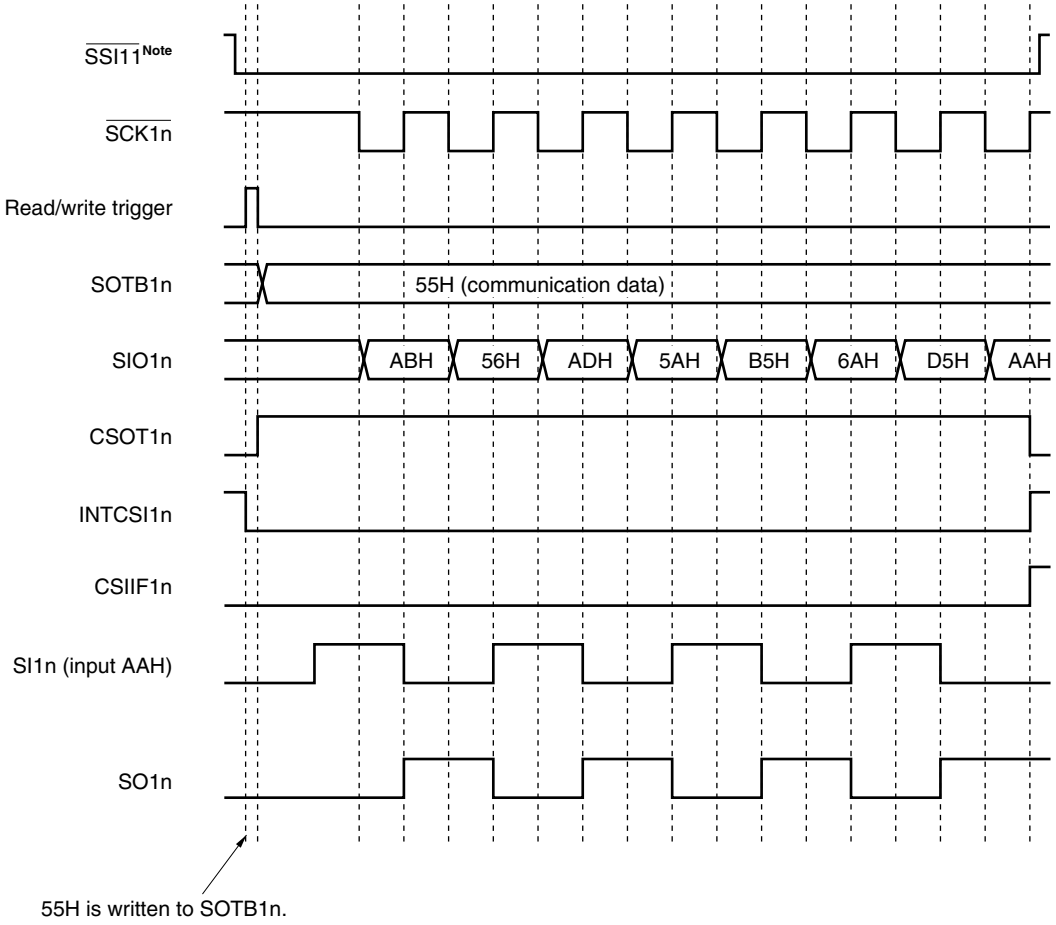


Note The SSE11 flag and $\overline{\text{SSI11}}$ pin are available only for serial interface CSI11 of the 78K0/KE2-C, and are used in the slave mode.

Remark n = 0, 1

Figure 15-10. Timing in 3-Wire Serial I/O Mode (2/2)

(b) Transmission/reception timing (Type 2: TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 1, SSE11 = 1[Note])

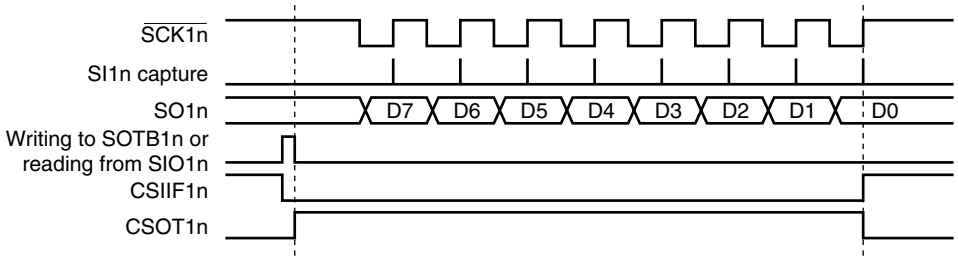


Note The SSE11 flag and $\overline{\text{SSI11}}$ pin are available only for serial interface CSI11 of the 78K0/KE2-C, and are used in the slave mode.

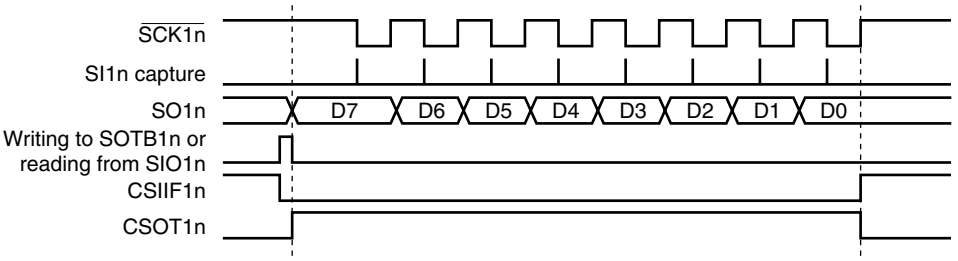
Remark n = 0, 1

Figure 15-11. Timing of Clock/Data Phase

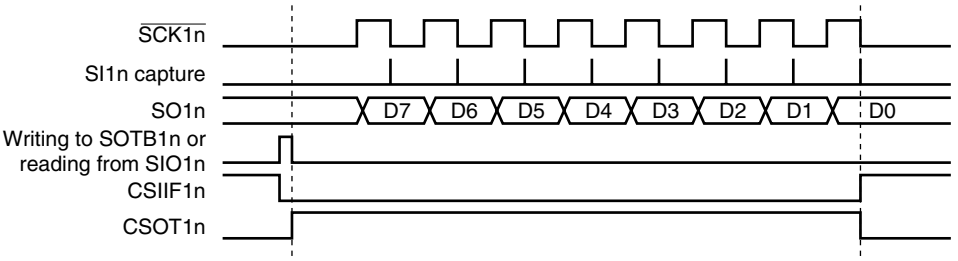
(a) Type 1: CKP1n = 0, DAP1n = 0, DIR1n = 0



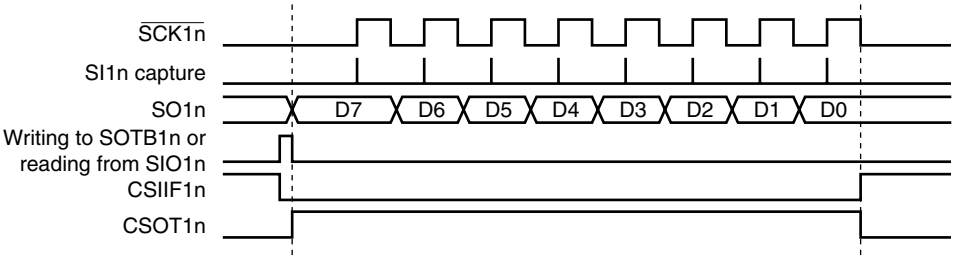
(b) Type 2: CKP1n = 0, DAP1n = 1, DIR1n = 0



(c) Type 3: CKP1n = 1, DAP1n = 0, DIR1n = 0



(d) Type 4: CKP1n = 1, DAP1n = 1, DIR1n = 0



Remarks 1. n = 0, 1

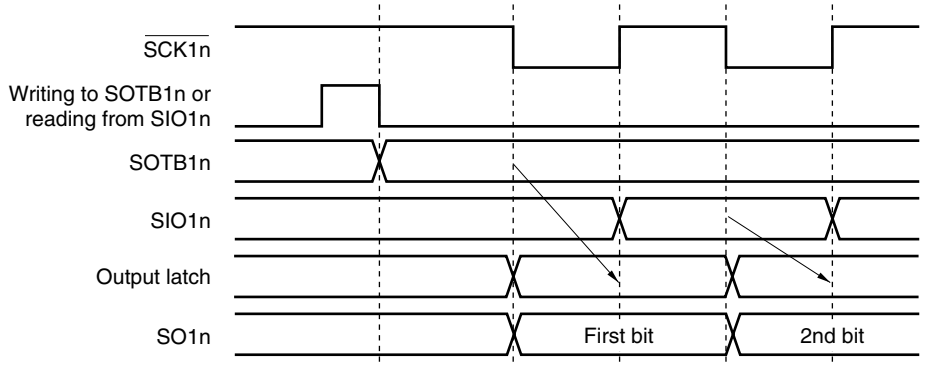
2. The above figure illustrates a communication operation where data is transmitted with the MSB first.

(3) Timing of output to SO1n pin (first bit)

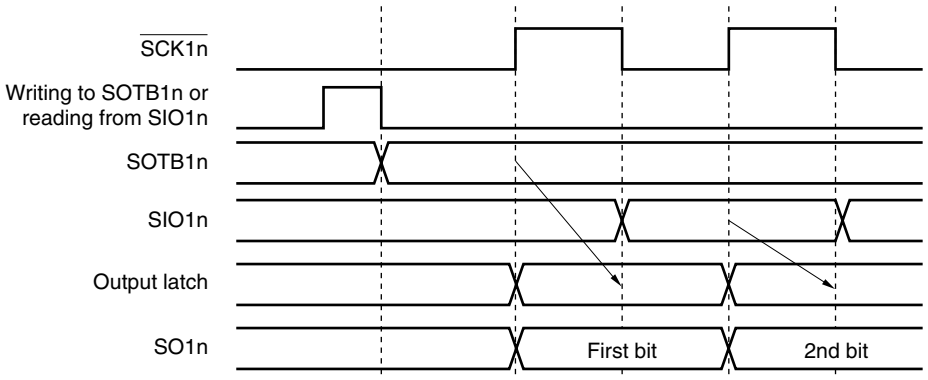
When communication is started, the value of transmit buffer register 1n (SOTB1n) is output from the SO1n pin. The output operation of the first bit at this time is described below.

Figure 15-12. Output Operation of First Bit (1/2)

(a) Type 1: CKP1n = 0, DAP1n = 0



(b) Type 3: CKP1n = 1, DAP1n = 0



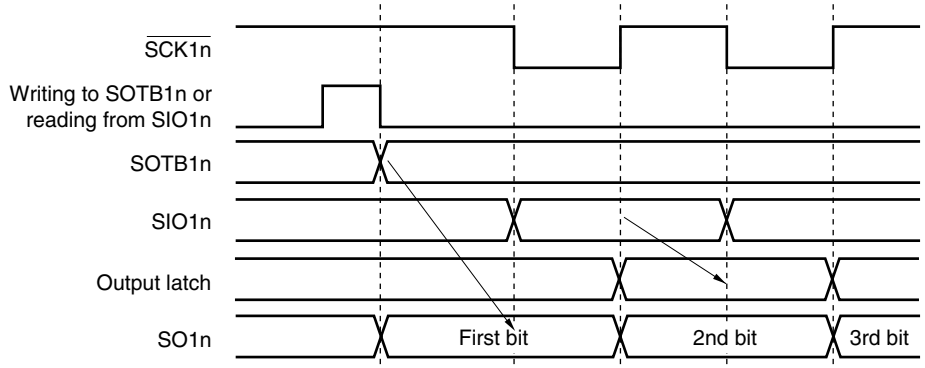
The first bit is directly latched by the SOTB1n register to the output latch at the falling (or rising) edge of $\overline{SCK1n}$, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next rising (or falling) edge of $\overline{SCK1n}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin.

The second and subsequent bits are latched by the SIO1n register to the output latch at the next falling (or rising) edge of $\overline{SCK1n}$, and the data is output from the SO1n pin.

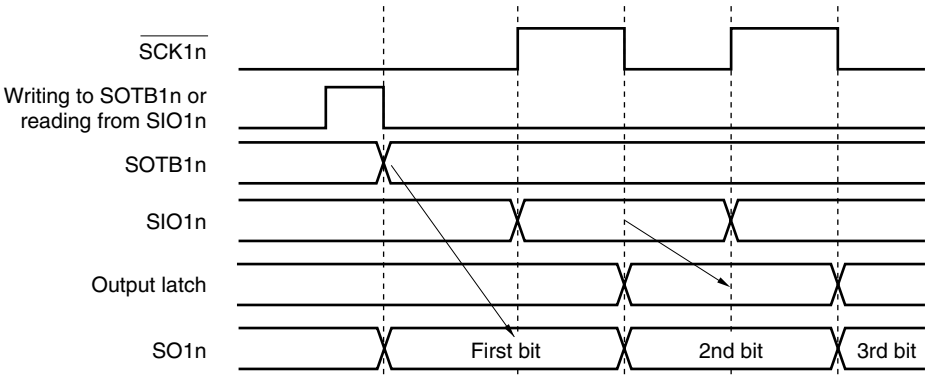
Remark n = 0, 1

Figure 15-12. Output Operation of First Bit (2/2)

(c) Type 2: CKP1n = 0, DAP1n = 1



(d) Type 4: CKP1n = 1, DAP1n = 1



The first bit is directly latched by the SOTB1n register at the falling edge of the write signal of the SOTB1n register or the read signal of the SIO1n register, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next falling (or rising) edge of $\overline{\text{SCK1n}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin.

The second and subsequent bits are latched by the SIO1n register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK1n}}$, and the data is output from the SO1n pin.

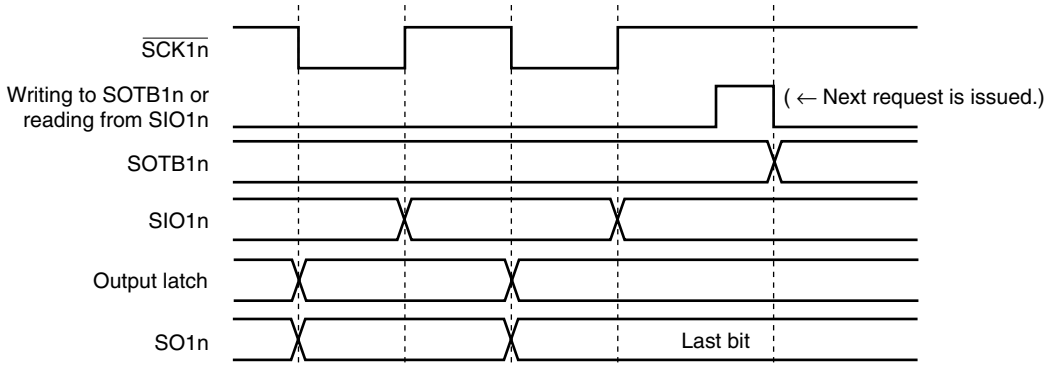
Remark n = 0, 1

(4) Output value of SO1n pin (last bit)

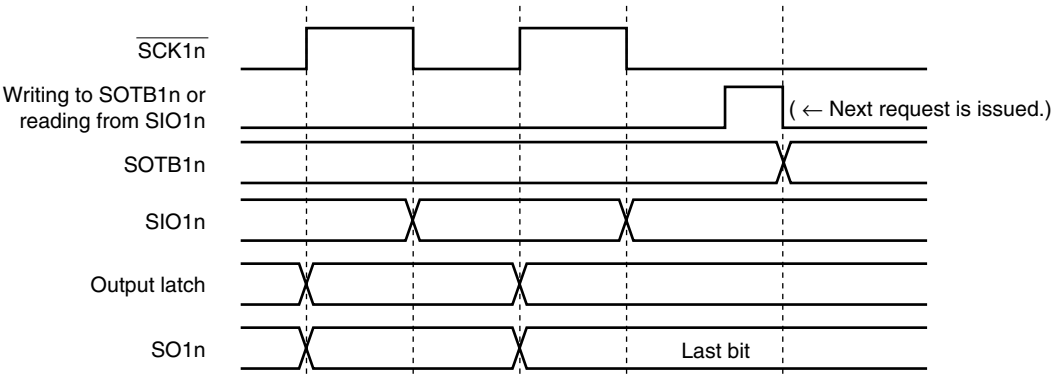
After communication has been completed, the SO1n pin holds the output value of the last bit.

Figure 15-13. Output Value of SO1n Pin (Last Bit) (1/2)

(a) Type 1: CKP1n = 0, DAP1n = 0



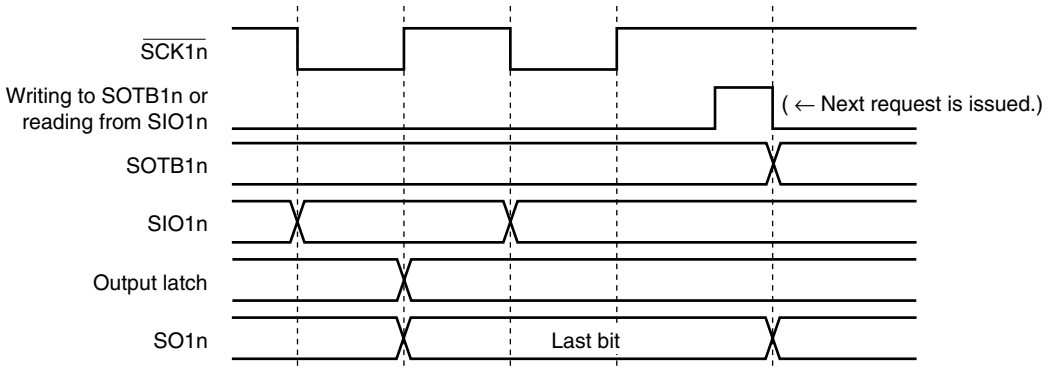
(b) Type 3: CKP1n = 1, DAP1n = 0



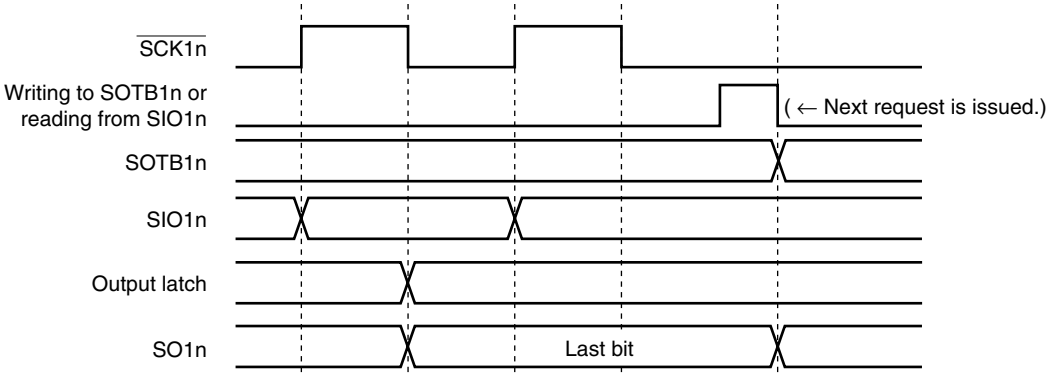
Remark n = 0, 1

Figure 15-13. Output Value of SO1n Pin (Last Bit) (2/2)

(c) Type 2: CKP1n = 0, DAP1n = 1



(d) Type 4: CKP1n = 1, DAP1n = 1



Remark n = 0, 1

(5) SO1n output (see Figures 15-1 and 15-2)

The status of the SO1n output is as follows depending on the setting of CSIE1n, TRMD1n, DAP1n, and DIR1n.

Table 15-3. SO1n Output Status

CSIE1n	TRMD1n	DAP1n	DIR1n	SO1n Output[Note 1]
CSIE1n = 0[Note 2]	TRMD1n = 0[Notes 2, 3]	–	–	Low level output[Note 2]
	TRMD1n = 1	DAP1n = 0	–	Low level output
		DAP1n = 1	DIR1n = 0	Value of bit 7 of SOTB1n
			DIR1n = 1	Value of bit 0 of SOTB1n
CSIE1n = 1	TRMD1n = 0[Note 3]	–	–	Low level output
	TRMD1n = 1	–	–	Transmission data[Note 4]

Notes 1. The actual output of following pins is determined according to PM12 and P12 or PM74 and P74, as well as the SO1n output.

- 78K0/KC2-C: SO10/P12, SO11/P74
- 78K0/KE2-C: SO10/P12, SO11/KR4/P74

2. This is a status after reset.

3. To use the following pins as general-purpose port, set the serial clock selection register 1n (CSIC1n) in the default status (00H).

- 78K0/KC2-C: P12/SO10, P74/SO11
- 78K0/KE2-C: P12/SO10, P74/SO11/KR4

4. After transmission has been completed, the SO1n pin holds the output value of the last bit of transmission data.

Caution If a value is written to CSIE1n, TRMD1n, DAP1n, and DIR1n, the output value of SO1n changes.

Remark n = 0, 1

CHAPTER 16 SERIAL INTERFACES IICA00, IICA01, AND IICA02

Caution If IICA00, IICA01, and IICA02 are used, key interrupt function cannot be used.

16.1 Functions of Serial Interfaces IICA00, IICA01, and IICA02

Serial interfaces IICA00, IICA01, and IICA02 have the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCLAn and SDAAn pins are used for open drain outputs, IICA0n require pull-up resistors for the serial clock line and the serial data bus line.

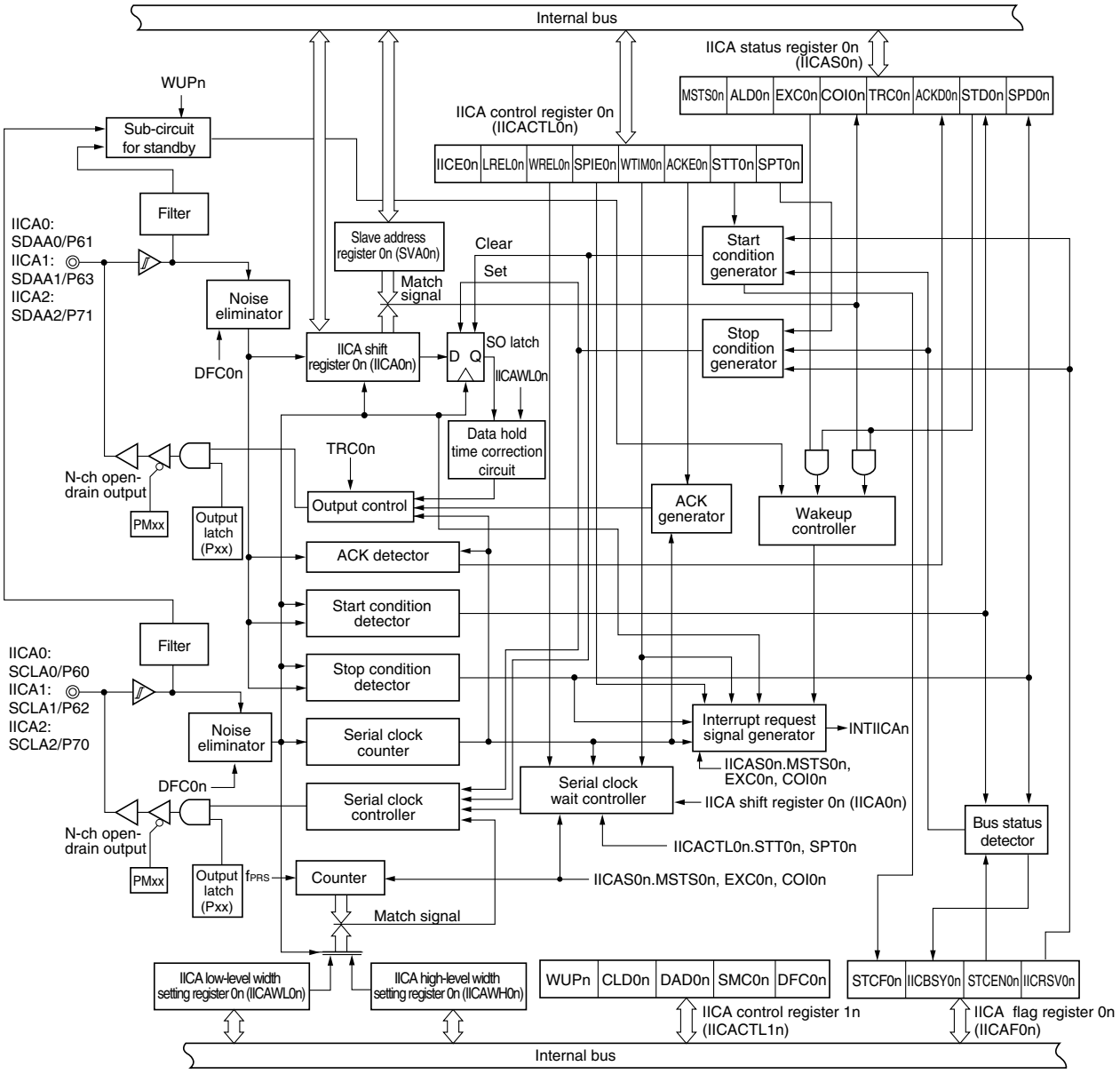
(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICAn) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register 1n (IICACTL1n).

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-1 shows a block diagram of serial interfaces IICA00, IICA01, and IICA02.

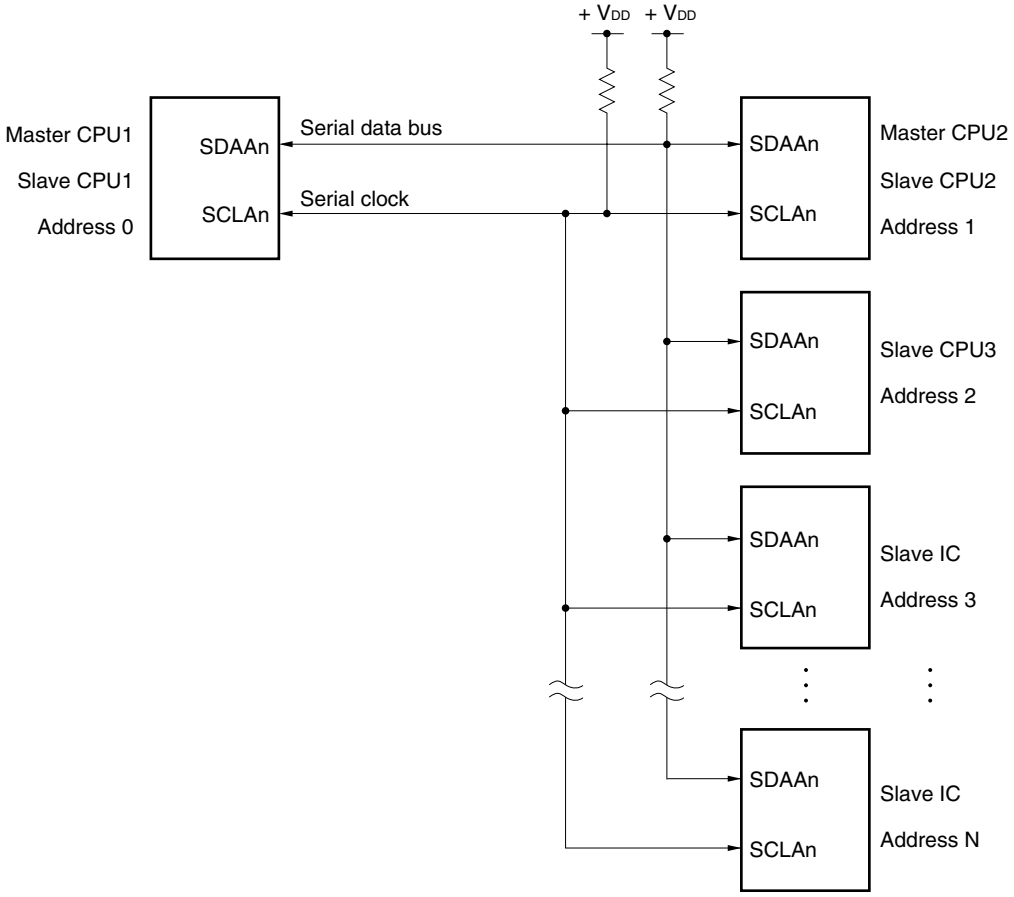
Figure 16-1. Block Diagram of Serial Interface IICA



Remark Serial interface IICA00: n = 0, PMxx = PM60, PM61, Pxx = P60, P61
Serial interface IICA01: n = 1, PMxx = PM62, PM63, Pxx = P62, P63
Serial interface IICA02: n = 2, PMxx = PM70, PM71, Pxx = P70, P71

Figure 16-2 shows a serial bus configuration example.

Figure 16-2. Serial Bus Configuration Example Using I²C Bus



Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.2 Configuration of Serial Interfaces IICA00, IICA01, and IICA02

Serial interfaces IICA00, IICA01, and IICA02 include the following hardware.

Table 16-1. Configuration of Serial Interfaces IICA00, IICA01, and IICA02

Item	Configuration
Registers	IICA shift register 0n (IICA0n) Slave address register 0n (SVA0n)
Control registers	IICA control register 0n (IICACTL0n) IICA status register 0n (IICAS0n) IICA flag register 0n (IICAF0n) IICA control register 1n (IICACTL1n) IICA low-level width setting register 0n (IICAWL0n) IICA high-level width setting register 0n (IICAWH0n) Port mode registers 6, 7 (PM6, PM7) Port registers 6, 7 (P6, P7)

(1) IICA shift register 0n (IICA0n)

IICA0n 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 IICA0n.

Cancel the wait state and start data transfer by writing data to IICA0n during the wait period.

IICA0n can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IICA0n to 00H.

Figure 16-3. Format of IICA Shift Register 0n (IICA0n)

Address: FFB0H (IICA00), FF85H (IICA01), FF3DH (IICA02) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA0n								

Cautions 1. Do not write data to IICA0n during data transfer.

2. Write or read IICA0n only during the wait period. Accessing IICA0n in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IICA0n can be written only once after the communication trigger bit (STT0n) is set to 1.

3. When communication is reserved, write data to IICA0n register after the interrupt triggered by a stop condition is detected.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(2) Slave address register 0n (SVA0n)

This register stores local addresses when in slave mode.

SVA0n can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0n = 1 (while the start condition is detected).

Reset signal generation clears SVA0n to 00H.

Figure 16-4. Format of Slave Address Register 0n (SVA0n)

Address: FFB1H (SVA00), FF86H (SVA01), FF9EH (SVA02) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0n								0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAAn pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICAn) when the address received by this register matches the address value set to the slave address register 0n (SVA0n) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICAn).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by WTIM0n bit)
- Interrupt request generated when a stop condition is detected (set by SPIE0n bit)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLAn pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

Remarks 1. WTIM0n bit: Bit 3 of IICA control register 0n (IICACTL0n)
SPIE0n bit: Bit 4 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(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 STT0n bit is set to 1.

However, in the communication reservation disabled status (IICRSV0n bit = 1), when the bus is not released (IICBSY0n bit = 1), start condition requests are ignored and the STCF0n bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0n 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 STCEN0n bit.

Remarks 1. STT0n bit: Bit 1 of IICA control register 0n (IICACTL0n)
SPT0n bit: Bit 0 of IICA control register 0n (IICACTL0n)
IICRSV0n bit: Bit 0 of IICA flag register 0n (IICAF0n)
IICBSY0n bit: Bit 6 of IICA flag register 0n (IICAF0n)
STCF0n bit: Bit 7 of IICA flag register 0n (IICAF0n)
STCEN0n bit: Bit 1 of IICA flag register 0n (IICAF0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.3 Registers Controlling Serial Interfaces IICA00, IICA01, and IICA02

Serial interfaces IICA00, IICA01, and IICA02 are controlled by the following eight registers.

- IICA control register 0n (IICACTL0n)
- IICA status register 0n (IICAS0n)
- IICA flag register 0n (IICAF0n)
- IICA control register 1n (IICACTL1n)
- IICA low-level width setting register 0n (IICAWL0n)
- IICA high-level width setting register 0n (IICAWH0n)
- Port mode registers 6, 7 (PM6, PM7)
- Port registers 6, 7 (P6, P7)

(1) IICA control register 0n (IICACTL0n)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

IICACTL0n can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0n, WTIM0n, and ACKE0n bits while IICE0n bit = 0 or during the wait period. These bits can be set at the same time when the IICE0n bit is set from "0" to "1".

Reset signal generation clears this register to 00H.

Figure 16-5. Format of IICA Control Register 0n (IICACTL0n) (1/5)

Address: FFB2H (IICACTL00), FF8AH (IICACTL01), FF7AH (IICACTL02) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICACTL0n	IICE0n	LREL0n	WREL0n	SPIE0n	WTIM0n	ACKE0n	STT0n	SPT0n

IICE0n	I^2C operation enable	
0	Stop operation. Reset the IICA status register 0n (IICAS0n)Note. Stop internal operation.	
1	Enable operation.	
Be sure to set this bit (1) while the SCLAn and SDAAn lines are at high level.		
Condition for clearing (IICE0n = 0)		Condition for setting (IICE0n = 1)
• Cleared by instruction • Reset		• Set by instruction

Note The IICAS0n register, the STCF0n and IICBSY0n bits of the IICAF0n register, and the CLD0n and DAD0n bits of the IICACTL1n register are reset.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-5. Format of IICA Control Register 0n (IICACTL0n) (2/5)

LREL0n[Note 1]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register 0n (IICACTL0n) and IICA status register 0n (IICAS0n) are cleared to 0. • STT0n • SPT0n • MSTS0n • EXC0n • COI0n • TRC0n • ACKD0n • STD0n
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL0n = 0)	Condition for setting (LREL0n = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0n[Note 1]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WREL0n is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0n = 1), the SDAAn line goes into the high impedance state (TRC0n = 0).	
Condition for clearing (WREL0n = 0)	Condition for setting (WREL0n = 1)
• Automatically cleared after execution • Reset	• Set by instruction

SPIE0n[Note 2]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If WUPn bit of the IICA control register 1 (IICACTL1n) is 1, no stop condition interrupt will be generated even if SPIE0n = 1.	
Condition for clearing (SPIE0n = 0)	Condition for setting (SPIE0n = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0n is 0.

2. The signal of this bit is invalid while IICE0n is 0. Set this bit during that period.

Caution The start condition is detected immediately after I^2C is enabled to operate (IICE0n = 1) while the SCLAn line is at high level and the SDAAn line is at low level. Immediately after enabling I^2C to operate (IICE0n = 1), set LREL0n (1) by using a 1-bit memory manipulation instruction.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-5. Format of IICA Control Register 0n (IICACTL0n) (3/5)

WTIM0n[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 (WTIM0n = 0)	Condition for setting (WTIM0n = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0n Notes 1, 2	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.
Condition for clearing (ACKE0n = 0)	Condition for setting (ACKE0n = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0n 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.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-5. Format of IICA Control Register 0n (IICACTL0n) (4/5)

STT0n[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY0n = 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 (IICRSV0n = 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 (IICRSV0n = 1) Even if this bit is set (1), the STT0n bit is cleared and the STT0n clear flag (STCF0n) 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 ACKE0n 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 SPT0n.
- Setting STT0n to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT0n = 0)	Condition for setting (STT0n = 1)
• Cleared by setting STT0n to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0n = 1 (exit from communications) • When IICE0n = 0 (operation stop) • Reset	• Set by instruction

Note The signal of this bit is invalid while IICE0n is 0.

Remarks 1. Bit 1 (STT0n) becomes 0 when it is read after data setting.

2. IICRSV0n: Bit 0 of IICA flag register 0n (IICAF0n)
STCF0n: Bit 7 of IICA flag register 0n (IICAF0n)

3. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-5. Format of IICA Control Register 0n (IICACTL0n) (5/5)

SPT0n	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer). After the SDAAn line goes to low level, either set the SCLAn line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDAAn line changes from low level to high level and a stop condition is generated.

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 ACKE0n 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 STT0n.
- SPT0n can be set to 1 only when in master mode.
- When WTIM0n has been cleared to 0, if SPT0n 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. WTIM0n should be changed from 0 to 1 during the wait period following the output of eight clocks, and SPT0n should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting SPT0n to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT0n = 0)	Condition for setting (SPT0n = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0n = 1 (exit from communications) • When IICE0n = 0 (operation stop) • Reset	• Set by instruction

Caution When bit 3 (TRC0n) of the IICA status register 0n (IICAS0n) is set to 1 (transmission status), bit 5 (WREL0n) of the IICA control register 0n (IICACTL0n) is set to 1 during the ninth clock and wait is canceled, after which TRC bit is cleared (reception status) and the SDA0n line is set to high impedance. Release the wait performed while TRC0n bit is 1 (transmission status) by writing to the IICA shift register 0n.

Remarks 1. Bit 0 (SPT0n) becomes 0 when it is read after data setting.

2. Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

(2) IICA status register 0n (IICAS0n)

This register indicates the status of I^2C.

IICAS0n is read by a 1-bit or 8-bit memory manipulation instruction only when STT0n = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICAS0n register while the address match wakeup function is enabled (WUPn = 1) in STOP mode is prohibited. When WUPn bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICAn interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE0n = 1) the interrupt generated by detecting a stop condition and read the IICAS0n register after the interrupt has been detected.

Figure 16-6. Format of IICA Status Register 0n (IICAS0n) (1/3)

Address: FFB6H (IICAS00), FFEDH (IICAS01), FF7DH (IICAS02) After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICAS0n	MSTS0n	ALD0n	EXC0n	COI0n	TRC0n	ACKD0n	STD0n	SPD0n

MSTS0n	Master status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0n = 0)	Condition for setting (MSTS0n = 1)
• When a stop condition is detected • When ALD0n = 1 (arbitration loss) • Cleared by LREL0n = 1 (exit from communications) • When IICE0n changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0n	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”. MSTS0n is cleared.

Condition for clearing (ALD0n = 0)	Condition for setting (ALD0n = 1)
• Automatically cleared after IICAS0n is read[Note] • When IICE0n changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

EXC0n	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0n = 0)	Condition for setting (EXC0n = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0n = 1 (exit from communications) • When IICE0n 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 IICAS0n. Therefore, when using the ALD0n bit, read the data of this bit before the data of the other bits.

Remarks 1. LREL0n: Bit 6 of IICA control register 0n (IICACTL0n)
IICE0n: Bit 7 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-6. Format of IICA Status Register 0n (IICAS0n) (2/3)

COI0n	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0n = 0)	Condition for setting (COI0n = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0n = 1 (exit from communications) • When IICE0n changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0n (SVA0n)) (set at the rising edge of the eighth clock).

TRC0n	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAAn line is set for high impedance.
1	Transmit status. The value in the SO0n latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte's ninth clock).

Condition for clearing (TRC0n = 0)	Condition for setting (TRC0n = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0n = 1 (exit from communications) • When IICE0n changes from 1 to 0 (operation stop) • Cleared by WREL0n = 1[Note] (wait cancel) • When ALD0n changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS0n, EXC0n, COI0n = 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) <When not used for communication>	<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 (TRC0n) of the IICA status register 0n (IICAS0n) is set to 1 (transmission status), bit 5 (WREL0n) of the IICA control register 0n (IICACTL0n) is set to 1 during the ninth clock and wait is canceled, after which TRC0n bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the wait performed while TRC bit is 1 (transmission status) by writing to the IICA shift register 0n.

Remarks 1. LREL0n: Bit 6 of IICA control register 0n (IICACTL0n)
IICE0n: Bit 7 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-6. Format of IICA Status Register 0n (IICAS0n) (3/3)

ACKD0n	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0n = 0)	Condition for setting (ACKD0n = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0n = 1 (exit from communications) • When IICE0n changes from 1 to 0 (operation stop) • Reset	• After the SDAAn line is set to low level at the rising edge of SCLAn's ninth clock

STD0n	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 (STD0n = 0)	Condition for setting (STD0n = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0n = 1 (exit from communications) • When IICE0n changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0n	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 (SPD0n = 0)	Condition for setting (SPD0n = 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 IICE0n changes from 1 to 0 (operation stop) • When WUPn changes from 1 to 0 • Reset	• When a stop condition is detected

<R>

Remarks 1. LREL0n: Bit 6 of IICA control register 0n (IICACTL0n)
IICE0n: Bit 7 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

(3) IICA flag register 0n (IICAF0n)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

IICAF0n can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STCF0n and IICBSY0n bits are read-only.

The IICRSV0n bit can be used to enable/disable the communication reservation function.

STCEN0n can be used to set the initial value of the IICBSY0n bit.

IICRSV0n and STCEN0n can be written only when the operation of I^2C is disabled (bit 7 (IICE0n) of IICA control register 0n (IICACTL0n) = 0). When operation is enabled, the IICAF0n register can be read.

Reset signal generation clears this register to 00H.

Figure 16-7. Format of IICA Flag Register 0n (IICAF0n)

Address: FF84H (IICAF00), FF87H (IICAF01), FF7CH (IICAF02) After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICAF0n	STCF0n	IICBSY0n	0	0	0	0	STCEN0n	IICRSV0n

STCF0n	STT0n clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT0n flag

Condition for clearing (STCF0n = 0)	Condition for setting (STCF0n = 1)
• Cleared by STT0n = 1 • When IICE0n = 0 (operation stop) • Reset	• Generating start condition unsuccessful and STT0n cleared to 0 when communication reservation is disabled (IICRSV0n = 1).

IICBSY0n	I^2C bus status flag
0	Bus release status (communication initial status when STCEN0n = 1)
1	Bus communication status (communication initial status when STCEN0n = 0)

Condition for clearing (IICBSY0n = 0)	Condition for setting (IICBSY0n = 1)
• Detection of stop condition • When IICE0n = 0 (operation stop) • Reset	• Detection of start condition • Setting of IICE0n when STCEN0n = 0

STCEN0n	Initial start enable trigger
0	After operation is enabled (IICE0n = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0n = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0n = 0)	Condition for setting (STCEN0n = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV0n	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0n = 0)	Condition for setting (IICRSV0n = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to STCEN0n only when the operation is stopped (IICE0n = 0).

2. As the bus release status (IICBSY0n = 0) is recognized regardless of the actual bus status when STCEN0n = 1, when generating the first start condition (STT0n = 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 IICRSV0n only when the operation is stopped (IICE0n = 0).

Remarks 1. STT0n: Bit 1 of IICA control register 0n (IICACTL0n)
IICE0n: Bit 7 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

(4) IICA control register 1n (IICACTL1n)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLAn and SDAAn pins.

IICACTL1n can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0n and DAD0n bits are read-only.

Set IICACTL1n, except the WUPn bit, while bit 7 (IICE0n) of IICA control register 0n (IICACTL0n) is 0.

Reset signal generation clears this register to 00H.

Figure 16-8. Format of IICA Control Register 1n (IICACTL1n) (1/2)

Address: FFB3H (IICACTL10), FF8BH (IICACTL11), FF7BH (IICACTL12) After reset: 00H R/W[Note 1]

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	0
IICACTL1n	WUPn	0	CLD0n	DAD0n	SMC0n	DFC0n	0	0

WUPn	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 WUPn = 1, execute the STOP instruction at least three clocks after setting (1) WUPn bit (see **Figure 16-21 Flow When Setting WUPn = 1**). Clear (0) WUPn after the address has matched or an extension code has been received. The subsequent communication can be entered by clearing (0) WUPn. (The wait must be released and transmit data must be written after WUPn has been cleared (0).) The interrupt timing when the address has matched or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if the SPIE0n bit is set to 1.	
Condition for clearing (WUPn = 0)	Condition for setting (WUPn = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when MSTS0n, EXC0n, and COI0n are “0”, and STD0n also “0” (communication not entered))[Note 2]

CLD0n	Detection of SCLAn pin level (valid only when IICE0n = 1)
0	The SCLAn pin was detected at low level.
1	The SCLAn pin was detected at high level.
Condition for clearing (CLD0n = 0)	Condition for setting (CLD0n = 1)
• When the SCLAn pin is at low level • When IICE0n = 0 (operation stop) • Reset	• When the SCLAn pin is at high level

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

Figure 16-8. Format of IICA Control Register 1n (IICACTL1n) (2/2)

DAD0n	Detection of SDAAn pin level (valid only when IICE0n = 1)
0	The SDAAn pin was detected at low level.
1	The SDAAn pin was detected at high level.

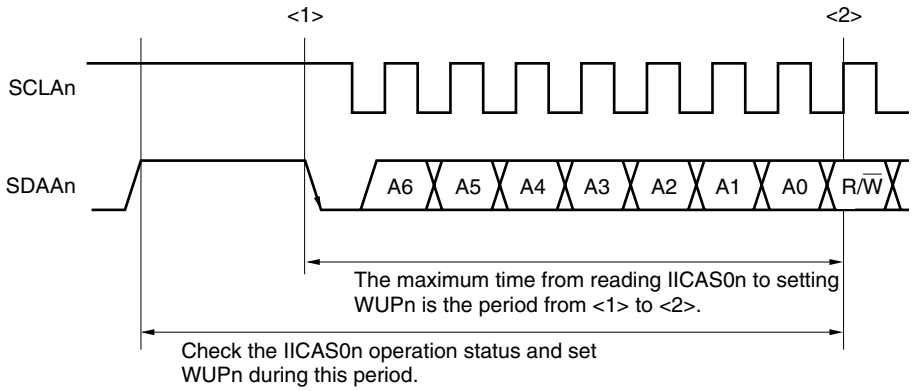
Condition for clearing (DAD0n = 0)	Condition for setting (DAD0n = 1)
• When the SDAAn pin is at low level • When IICE0n = 0 (operation stop) • Reset	• When the SDAAn pin is at high level

SMC0n	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC0n	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 DFC0n bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

Notes 1. Bits 4 and 5 are read-only.

2. The status of IICAS0n must be checked and WUPn must be set during the period shown below.



Remarks 1. IICE0n: Bit 7 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(5) IICA low-level width setting register 0n (IICAWL0n)

This register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by serial interface IICA0n being in master mode. This register is also used to set the data hold time ($t_{HD:DAT}$). The data hold time is determined according to the higher 6 bits of IICAWL0n.

IICAWL0n can be set by an 8-bit memory manipulation instruction.

Set IICAWL0n when bit 7 (IICE0n) of IICA control register 0n (IICACTL0n) is 0.

Reset signal generation sets this register to FFH.

Figure 16-9. Format of IICA Low-Level Width Setting Register 0n (IICAWL0n)

Address: FFB8H (IICAWL00), FFEEH (IICAWL01), FF7EH (IICAWL02) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICAWL0n								

Set values of data hold time

(6) IICA high-level width setting register 0n (IICAWH0n)

This register is used to set the high-level width (t_{HIGH}) of the SCLAn pin signal that is output by serial interface IICA0n being in master mode.

IICAWH0n can be set by an 8-bit memory manipulation instruction.

Set IICAWH0n when bit 7 (IICE0n) of IICA control register 0n (IICACTL0n) is 0.

Reset signal generation sets this register to FFH.

Figure 16-10. Format of IICA High-Level Width Setting Register 0n (IICAWH0n)

Address: FFB9H (IICAWH00), FFEFH (IICAWH01), FF7FH (IICAWH02) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICAWH0n								

Remarks 1. For how to set the transfer clock by using the IICAWL0n and IICAWH0n registers, see **16.4.2 Setting transfer clock by using IICAWL0n and IICAWH0n registers**.

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(7) Port mode registers 6, 7 (PM6, PM7)

This register sets the input/output of ports 6 and 7 in 1-bit units.

When using the P60/SCLA0, P62/SCLA1, and P70/SCLA2 pins as clock I/O and the P61/SDAA0, P63/SDAA1, P71/SDAA2 pins as serial data I/O, clear PM60 to PM63, PM70, and PM71 to 0, and set the output latches of P60 to P61, P70, and P71 to 1.

Set IICE0n (bit 7 of IICA control register 0n (IICACTL0n)) to 1 before setting the output mode because the P60/SCLA0, P62/SCLA1, P70/SCLA2 and P61/SDAA0, P63/SDAA1, P71/SDDA2 pins output a low level (fixed) when IICE0n is 0.

PM6 and PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 16-11. Format of Port Mode Registers 6, 7 (PM6, PM7)

Address: FF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	PM63	PM62	PM61	PM60

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77Note	PM76Note	PM75	PM74	PM73	PM72	PM71	PM70

PMmn	Pmn pin I/O mode selection (m = 6, 7; n = 0 to 7, mn = 60 to 63, 70 to 77)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 78K0/KE2-C only

16.4 I^2C Bus Mode Functions

16.4.1 Pin configuration

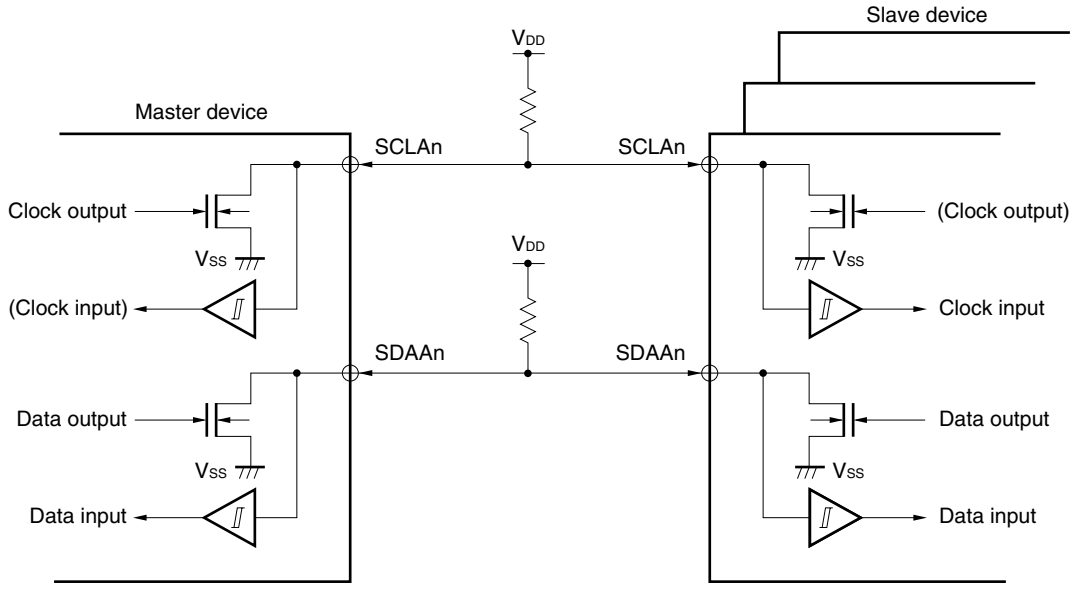
The serial clock pin (SCLAn) and serial data bus pin (SDAAn) are configured as follows.

(1) SCLAn .... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDAAn .... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 16-12. Pin Configuration Diagram



Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.4.2 Setting transfer clock by using IICAWL0n and IICAWH0n registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{PRS}}{IICAWL0n + IICAWH0n + f_{PRS}(t_R + t_F)}$$

At this time, the optimal setting values of IICAWL0n and IICAWH0n registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICAWL0n = \frac{0.52}{\text{Transfer clock}} \times f_{PRS}$$

$$IICAWH0n = (\frac{0.48}{\text{Transfer clock}} - t_R - t_F) \times f_{PRS}$$

- When the normal mode

$$IICAWL0n = \frac{0.47}{\text{Transfer clock}} \times f_{PRS}$$

$$IICAWH0n = (\frac{0.53}{\text{Transfer clock}} - t_R - t_F) \times f_{PRS}$$

Remark The data hold time is as follow, depending on the IICAWL0n setting.

$$\text{Data hold time} = \frac{\text{Higher 6 bits of IICAWL0n}}{f_{PRS}}$$

Example <Condition>

- Transfer clock = 400 kHz (fast mode)
  - f_{PRS} = 20 MHz
  - IICAWL0n = 26 (higher 6 bits = 6)

–> $\frac{6}{20000000} = 0.3\ [\mu s]$

Caution The data hold time must be set within 0.9 μs in the fast mode or 3.45 μs in the normal mode.

(2) Setting IICAWL0n and IICAWH0n registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICAWL0n = 1.3\ [\mu s] \times f_{PRS}$$

$$IICAWH0n = (1.2\ [\mu s] - t_R - t_F) \times f_{PRS}$$

- When the normal mode

$$IICAWL0n = 4.7\ [\mu s] \times f_{PRS}$$

$$IICAWH0n = (5.3\ [\mu s] - t_R - t_F) \times f_{PRS}$$

Caution Note the minimum f_{PRS} operation frequency when setting the transfer clock. The minimum f_{PRS} operation frequency for serial interface IICA0n is determined according to the mode.

Fast mode: f_{PRS} = 3.5 MHz (MIN.)

Normal mode: f_{PRS} = 1 MHz (MIN.)

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

Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistance and wire load.

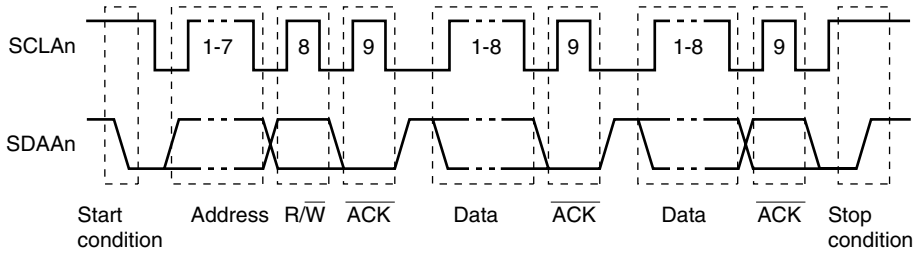
2. IICAWL0n: IICA low-level width setting register 0n
IICAWH0n: IICA high-level width setting register 0n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{PRS}: Peripheral hardware clock frequency

3. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5 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 16-13 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 16-13. I^2C Bus Serial Data Transfer Timing



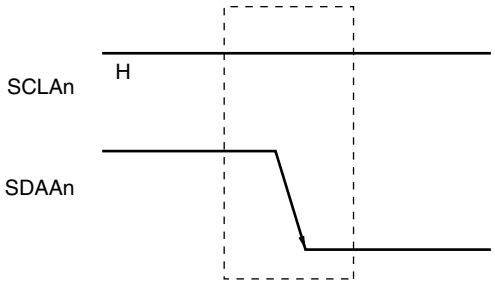
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLAn) is continuously output by the master device. However, in the slave device, the SCLAn's low level period can be extended and a wait can be inserted.

16.5.1 Start conditions

A start condition is met when the SCLAn pin is at high level and the SDAAn pin changes from high level to low level. The start conditions for the SCLAn pin and SDAAn pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 16-14. Start Conditions



A start condition is output when bit 1 (STT0n) of IICA control register 0 (IICACTL0n) is set (1) after a stop condition has been detected (SPD0n: Bit 0 of the IICA status register 0n (IICAS0n) = 1). When a start condition is detected, bit 1 (STD0n) of IICAS0n is set (1).

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

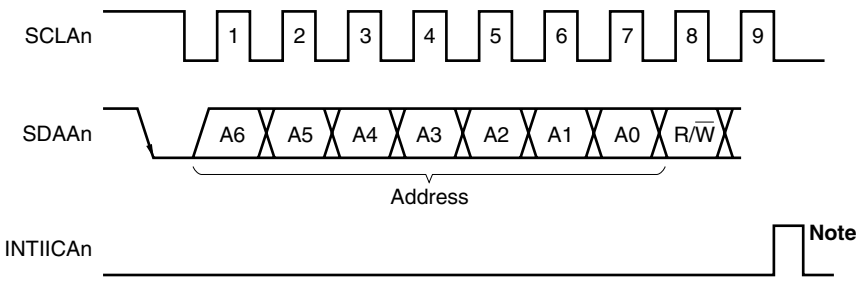
16.5.2 Addresses

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 0n (SVA0n). If the address data matches the SVA0n values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 16-15. Address



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **16.5.3 Transfer direction specification** are written to the IICA shift register 0n (IICA0n). The received addresses are written to IICA0n.

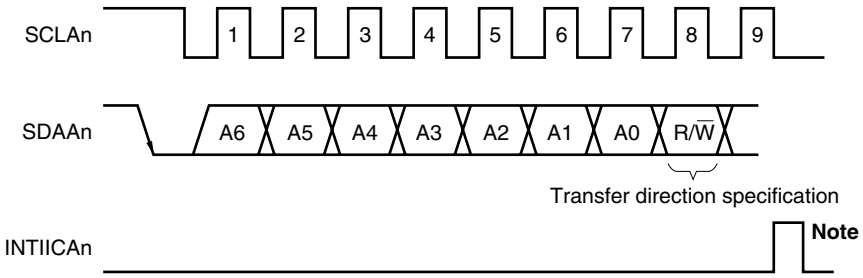
The slave address is assigned to the higher 7 bits of IICA0n.

16.5.3 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 16-16. Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

<R>16.5.4 Acknowledge ($\overline{\text{ACK}}$)

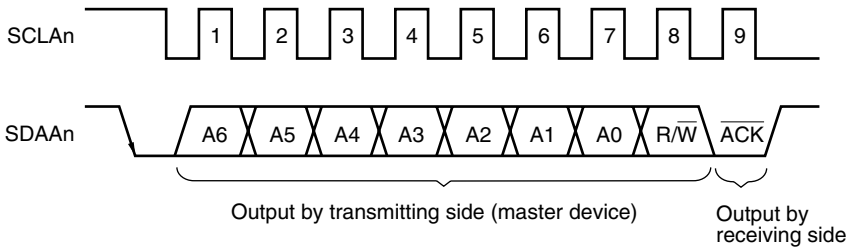
(1) Reason for using acknowledge ($\overline{\text{ACK}}$)

An acknowledge ($\overline{\text{ACK}}$) is used by the transmitting side to check the state of the receiving side. The receiving side returns an $\overline{\text{ACK}}$ each time it receives 8-bit data, indicating that communication was carried out normally. After sending 8-bit data, receiving an $\overline{\text{ACK}}$, and returning an $\overline{\text{ACK}}$ from the receiving side, the transmitting side continues processing assuming that the data has been received correctly. For this reason, basic communication on the I^2C bus consists of 9 bits: 8 bits of data followed by an acknowledge bit.

(2) Acknowledge timing

The receiving side returns an $\overline{\text{ACK}}$ (indicating normal reception) by setting the SDA signal to low level at the 9th bit. The acknowledge timing with respect to the address is shown in Figure 16-17 below.

Figure 16-17. $\overline{\text{ACK}}$



(3) If an $\overline{\text{ACK}}$ is not returned

If an $\overline{\text{ACK}}$ is not returned, it indicates that communication cannot continue due to a problem on the receiving side. Non-return of an $\overline{\text{ACK}}$ takes the following three forms:

<1> An ACK is not returned when an address is sent

This occurs when the slave device specified by the address does not exist or when the specified slave device cannot respond to I^2C bus communication while processing the transferred data.

<2> An ACK is not returned when the slave device receives the data

This occurs when there is an error in the data received by the slave device or when the slave device cannot receive the data at all.

<3> An $\overline{\text{ACK}}$ is not returned when the master device receives data

This occurs when the master device wants to notify the slave device that it has received the final data and no more data is required. The slave device confirms this and stops driving the I^2C bus, enabling the master device to generate a stop condition.

(4) Processing when an $\overline{\text{ACK}}$ is not returned

If an $\overline{\text{ACK}}$ is not returned, the transmitting side stops transmitting data. The master device then either generates a stop condition and ends communication or issues a restart condition and starts a new communication.

(5) Processing when an $\overline{\text{ACK}}$ is not returned

$\overline{\text{ACK}}$ can be generated by IICA at two different times. One is when the master device specifies an address. In this case, IICA compares the value of the SVA0 register with the transmitted address and automatically generates an $\overline{\text{ACK}}$ if the values match. If the values do not match, an $\overline{\text{ACK}}$ is not generated. (When receiving an extension code, an $\overline{\text{ACK}}$ can be generated by setting the ACKE0n bit to 1 in advance.)

The other is when data is sent from the transmitting side. In this case, an $\overline{\text{ACK}}$ can be specified to be automatically generated by setting bit 2 (ACKE0n) of IICA control register 0n (IICACTL0n) to 1.

(6) Controlling the ACKE0n bit on IICA

How an $\overline{\text{ACK}}$ is generated when data is received differs as follows depending on the setting of the wait timing.

<1> When 8-clock wait state is selected (bit 3 (WTIM0n) of IICACTL0n register = 0):
By setting the ACKE0n bit to 1 before releasing the wait state, an $\overline{\text{ACK}}$ is generated at the falling edge of the eighth clock of the SCLAn pin.

<2> When 9-clock wait state is selected (bit 3 (WTIM0n) of IICACTL0n register = 1):
An $\overline{\text{ACK}}$ can be generated at the next transfer by setting the ACKE0n bit to 1 in advance. Then, if the slave device cannot or does not need to receive any more data, generation of an $\overline{\text{ACK}}$ at the next communication can be canceled by clearing the ACKE0n bit to 0.

(7) Example of controlling the ACKE0n bit on IICA

<1> If the slave device cannot receive any more data during slave reception (TRC0n = 0), or it is not necessary to receive the next data, wait 9 clock cycles and then clear the ACKE0n bit to 0 to notify the master device that the slave device can no longer receive data.

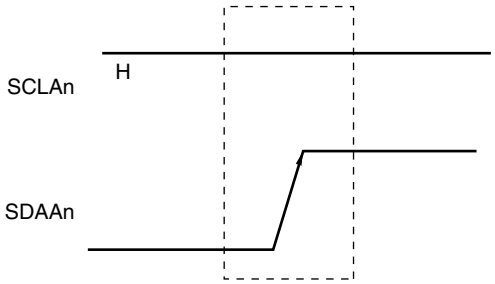
<2> If the master device does not need the next data during master reception (TRC0n = 0), wait for 8 clock cycles and then clear the ACKE0n bit to 0 to stop an $\overline{\text{ACK}}$ being generated. This will inform the slave device at the transmitting side that data reception is complete (and transmission has stopped).

16.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 16-18. Stop Condition



A stop condition is generated when bit 0 (SPT0n) of IICA control register 0n (IICACTL0n) is set to 1. When the stop condition is detected, bit 0 (SPD0n) of the IICA status register 0n (IICAS0n) is set to 1 and INTIICAn is generated when bit 4 (SPIE0n) of IICACTL0n is set to 1.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

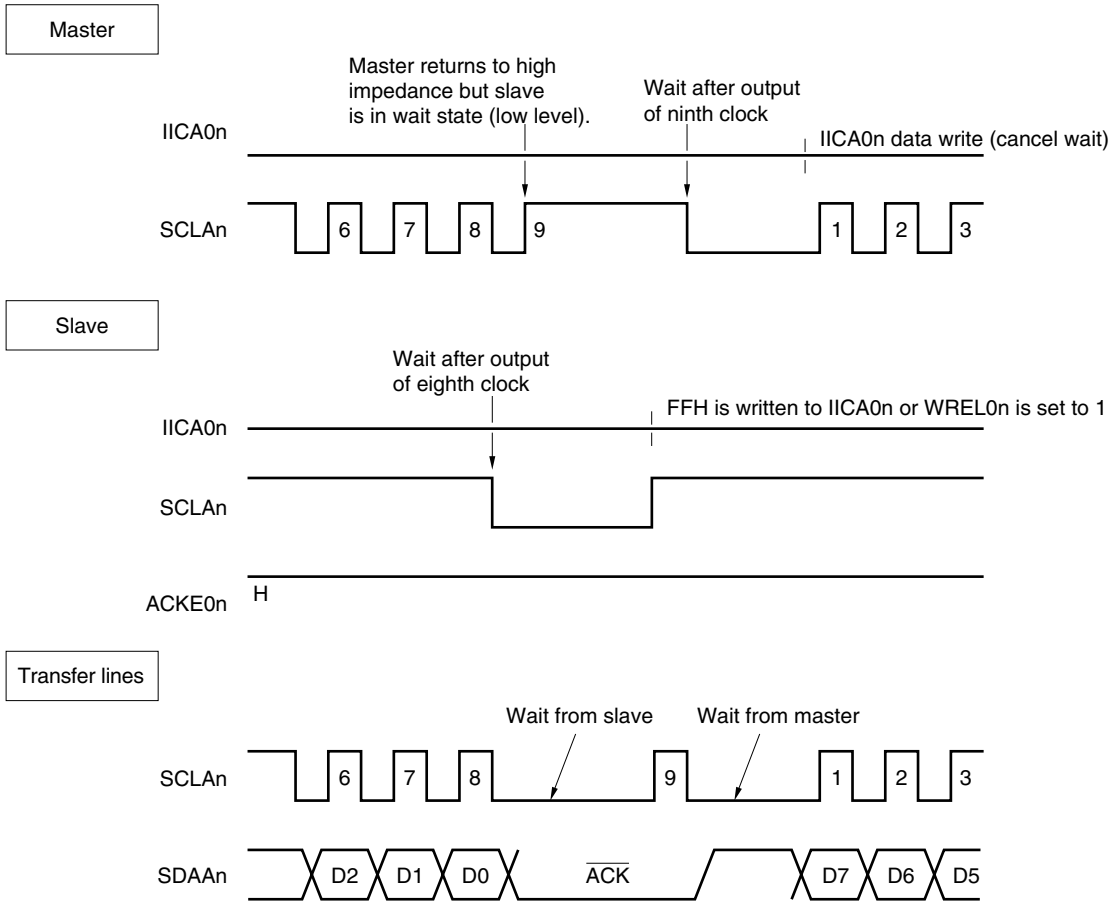
16.5.6 Wait

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 SCLAn 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 16-19. Wait (1/2)

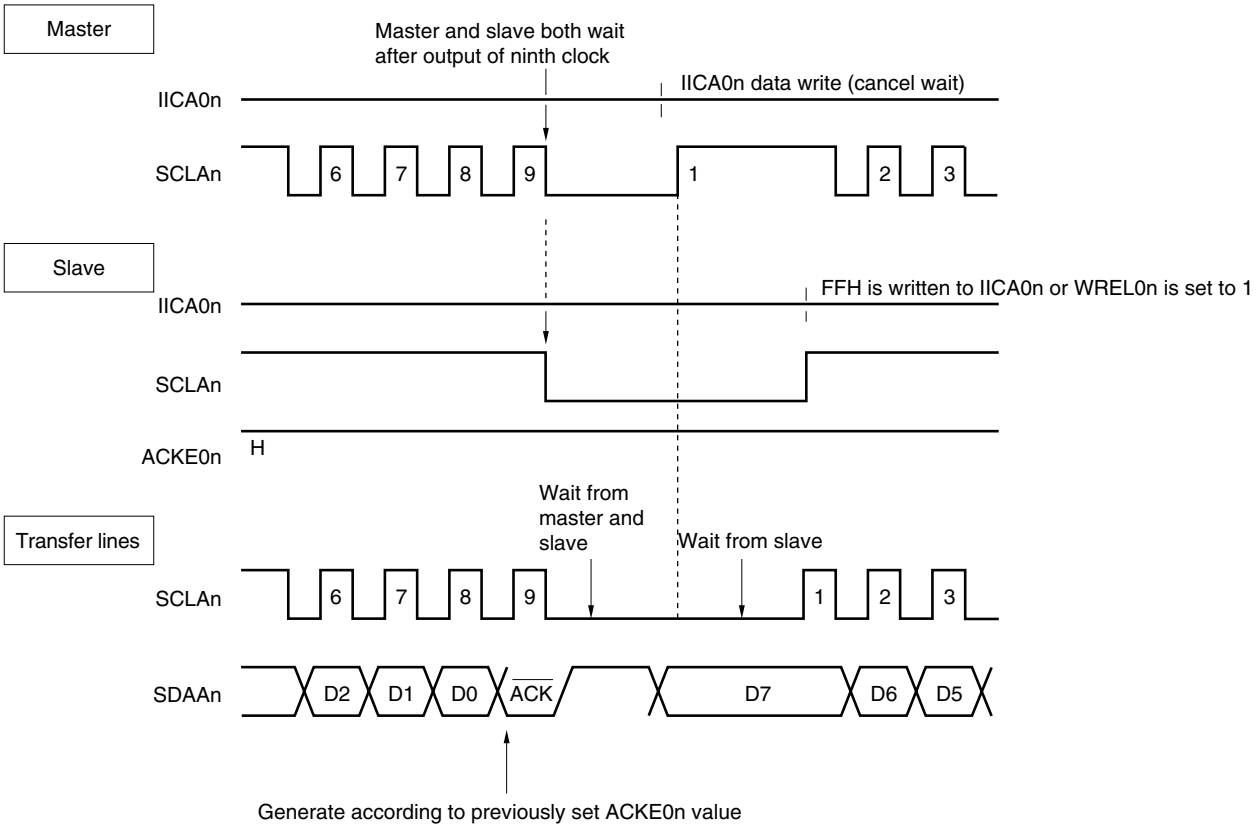
(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0n = 1)



Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-19. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0n = 1)



A wait may be automatically generated depending on the setting of bit 3 (WTIM0n) of IICA control register 0n (IICACTL0n).

Normally, the receiving side cancels the wait state when bit 5 (WREL0n) of IICACTL0n is set to 1 or when FFH is written to the IICA shift register 0n (IICA0n), and the transmitting side cancels the wait state when data is written to IICA0n.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0n) of IICACTL0n to 1
- By setting bit 0 (SPT0n) of IICACTL0n to 1

Remarks 1. ACKE0n: Bit 2 of IICA control register 0n (IICACTL0n)
WREL0n: Bit 5 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to IICA shift register 0n (IICA0n)
- Setting bit 5 (WREL0n) of IICA control register 0n (IICACTL0n) (canceling wait)
- Setting bit 1 (STT0n) of IICACTL0n register (generating start condition)[Note]
- Setting bit 0 (SPT0n) of IICACTL0n 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 IICA0n.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0n) of IICA control register 0n (IICACTL0n) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0n) of IICACTL0n to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0n) of IICACTL0n to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IICA0n after canceling a wait state by setting WREL0n to 1, an incorrect value may be output to SDAAn because the timing for changing the SDAAn line conflicts with the timing for writing IICA0n.

In addition to the above, communication is stopped if IICE0n is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I^2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0n) of IICACTL0n, so that the wait state can be canceled.

Caution If a processing to cancel a wait state executed when WUPn (bit 7 of IICA control register 1n (IICACTL1n)) = 1, the wait state will not be canceled.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5.8 Interrupt request (INTIICAn) generation timing and wait control

The setting of bit 3 (WTIM0n) of IICA control register 0n (IICACTL0n) determines the timing by which INTIICAn is generated and the corresponding wait control, as shown in Table 16-2.

Table 16-2. INTIICAn Generation Timing and Wait Control

WTIM0n	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 INTIICAn 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 0n (SVA0n).
At this point, $\overline{ACK}$ is generated regardless of the value set to IICACTL0n's bit 2 (ACKE0n). For a slave device that has received an extension code, INTIICAn occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICAn is generated at the falling edge of the 9th clock, but wait does not occur.

2. If the received address does not match the contents of the slave address register 0n (SVA0n) and extension code is not received, neither INTIICAn 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 WTIM0n bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0n bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0n bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0n bit.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IICA shift register (IICA0n)
- Setting bit 5 (WREL0n) of IICA control register 0n (IICACTL0n) (canceling wait)
- Setting bit 1 (STT0n) of IICACTL0n register (generating start condition)[Note]
- Setting bit 0 (SPT0n) of IICACTL0n register (generating stop condition)[Note]

Note Master only.

When an 8-clock wait has been selected (WTIM0n = 0), the presence/absence of $\overline{\text{ACK}}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICAn is generated when a stop condition is detected (only when SPIE0n = 1).

16.5.9 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 (INTIICAn) occurs when a local address has been set to the slave address register 0n (SVA0n) and when the address set to SVA0n matches the slave address sent by the master device, or when an extension code has been received.

16.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register 0n (IICA0n) of the transmitting device, so the IICA0n data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0n) is set to 1 for extension code reception and an interrupt request (INTIICAn) is issued at the falling edge of the eighth clock. The local address stored in the slave address register 0n (SVA0n) is not affected.

(2) If “11110××0” is set to SVA0n by a 10-bit address transfer and “11110××0” is transferred from the master device, the results are as follows. Note that INTIICAn occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0n = 1
- Seven bits of data match: COI0n = 1

(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 (LREL0n) of the IICA control register 0n (IICACTL0n) to 1 to set the standby mode for the next communication operation.

Table 16-3. Bit Definitions of Main Extension Code

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 (for address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (for read command issuance after address match)

Remarks 1. EXC0n: Bit 5 of IICA status register 0n (IICAS0n)
COI0n: Bit 4 of IICA status register 0n (IICAS0n)

2. For extension codes other than the above, refer to THE I^2C-BUS SPECIFICATION published by NXP.

3. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5.12 Arbitration

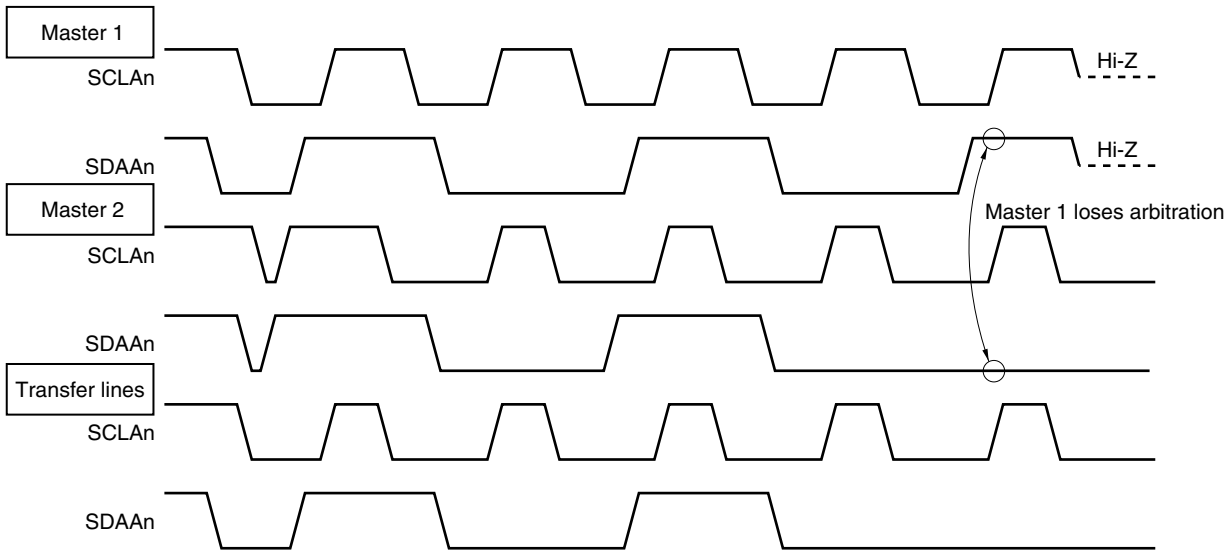
When several master devices simultaneously generate a start condition (when STT0n is set to 1 before STD0n is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD0n) in the IICA status register 0n (IICAS0n) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0n = 1 setting that has been made by software.

For details of interrupt request timing, refer to **16.5.8 Interrupt request (INTIICAn) generation timing and wait control**.

Figure 16-20. Arbitration Timing Example



Remarks 1. STD0n: Bit 1 of IICA status register 0n (IICAS0n)
STT0n: Bit 1 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Table 16-4. 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 SPIE0n = 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 SPIE0n = 1)[Note 2]
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCLAn is at low level while attempting to generate a restart condition	

Notes 1. When WTIM0n (bit 3 of IICA control register 0n (IICACTL0n)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0n = 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 SPIE0n = 1 for master device operation.

Remarks 1. SPIE0n: Bit 4 of IICA control register 0n (IICACTL0n)

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

<R>16.5.13 Wakeup function

(1) Overview of wakeup function

The wakeup function is used to generate an interrupt request signal (INTIICA0n) when a local address or extension code is received while the microcontroller is operating as a slave device on the I^2C bus. This function makes processing more efficient by preventing unnecessary INTIICA0n signal from occurring when addresses do not match.

(2) Operation of wakeup function

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, bit 4 (SPIE0n) of IICA control register 0n (IICACTL0n) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

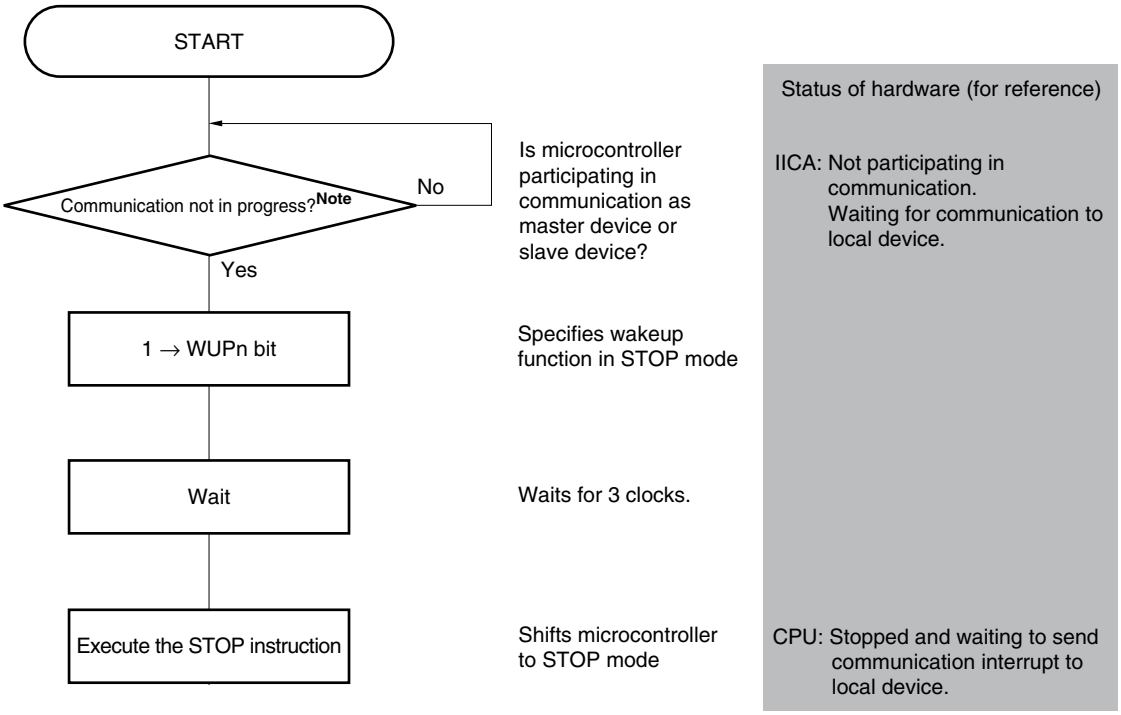
(3) Wakeup function in STOP mode

By shifting the microcontroller to STOP mode, the start of communication to a local address can be paused, saving power. To use the wakeup function in the STOP mode, set WUPn to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0n) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear the WUPn bit to 0 after this interrupt has been generated.

(4) Controlling the wakeup function in STOP mode

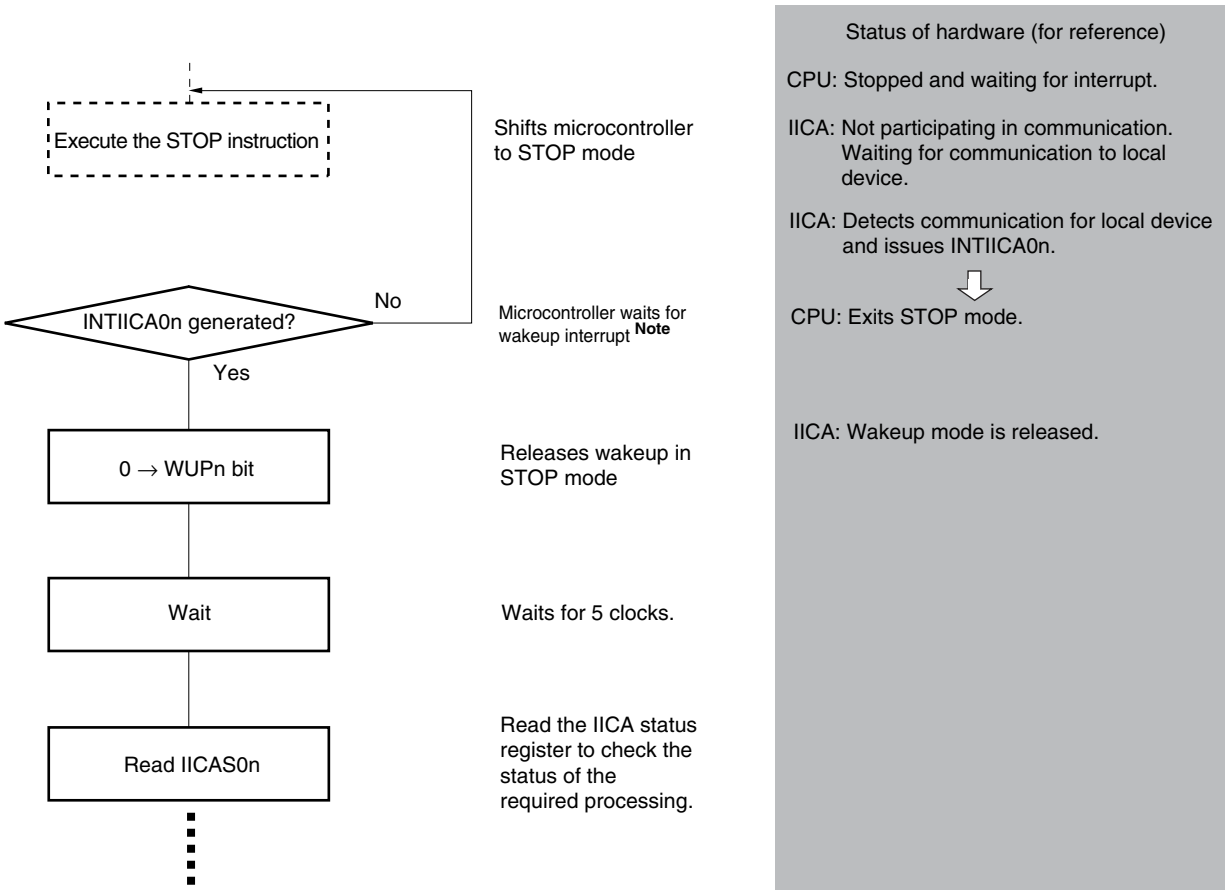
Figure 16-21 illustrates the flow for using the wakeup function in STOP mode, and Figure 16-22 illustrates how normal operation is restored when an address match occurs.

Figure 16-21. Flow for Using Wakeup Function in STOP Mode



Note MSTS0n = STD0n = EXC0n = C0I0n = 0

Figure 16-22. Flow of Restoring Normal Operation When Address Match Occurs



Note If STOP mode is released by a source other than the INTIICA0n interrupt, STOP mode is re-entered immediately after interrupt servicing finishes (if the microcontroller will continue to operate as a slave device).

16.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV0n) of IICA flag register 0n (IICAF0n) = 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 (LREL0n) of IICA control register 0n (IICACTL0n) to 1 and saving communication).

If bit 1 (STT0n) of IICACTL0n 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 0n (IICA0n) after bit 4 (SPIE0n) of IICACTL0n was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IICA0n before the stop condition is detected is invalid.
When STT0n has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode)..... communication reservation

Check whether the communication reservation operates or not by using MSTS0n (bit 7 of the IICA status register 0n (IICAS0n)) after STT0n is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

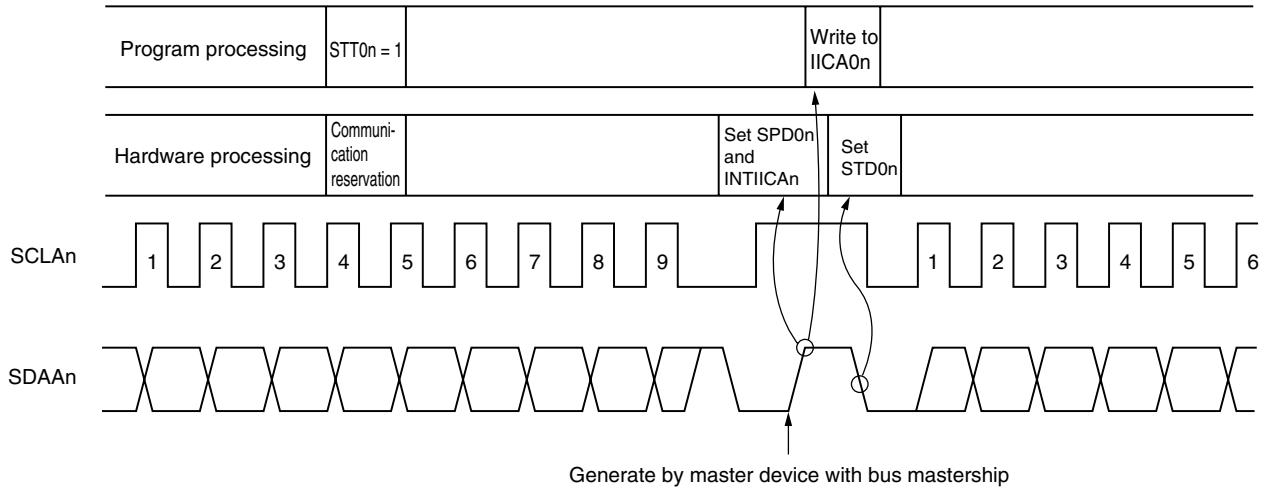
Wait time from setting STT0n = 1 to checking the MSTS0n flag:
(IICAWL0n setting value + IICAWH0n setting value + 4) + $t_F \times 2 \times f_{PRS}$ (clocks)

Remarks 1. IICAWL0n: IICA low-level width setting register 0n
IICAWH0n: IICA high-level width setting register 0n
t_F: SDAAn and SCLAn signal falling times
f_{PRS}: Peripheral hardware clock frequency

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-24 shows the communication reservation timing.

Figure 16-24. Communication Reservation Timing



Communication reservations are accepted via the timing shown in Figure 16-25. After bit 1 (STD0n) of the IICA status register 0n (IICAS0n) is set to 1, a communication reservation can be made by setting bit 1 (STT0n) of IICA control register 0n (IICACTL0n) to 1 before a stop condition is detected.

Figure 16-25. Timing for Accepting Communication Reservations

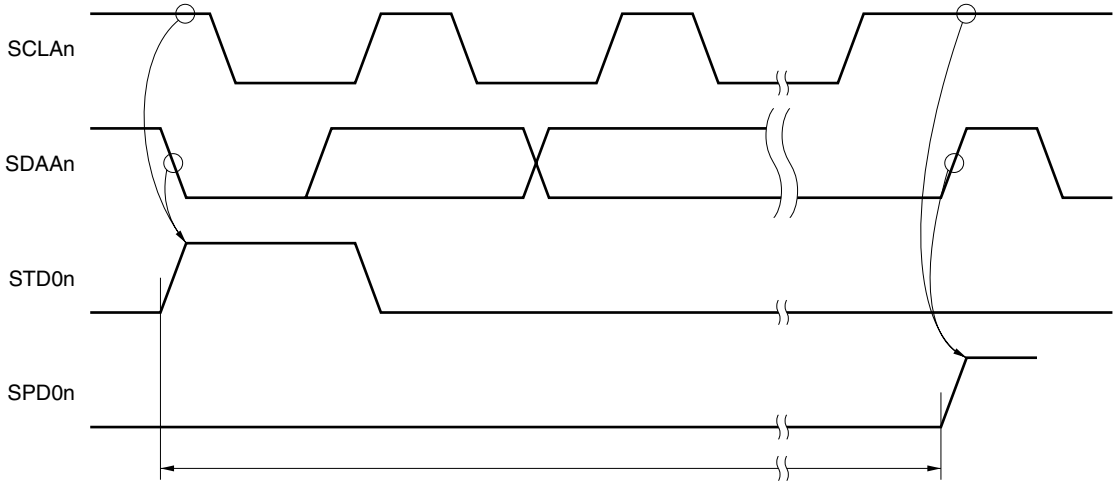
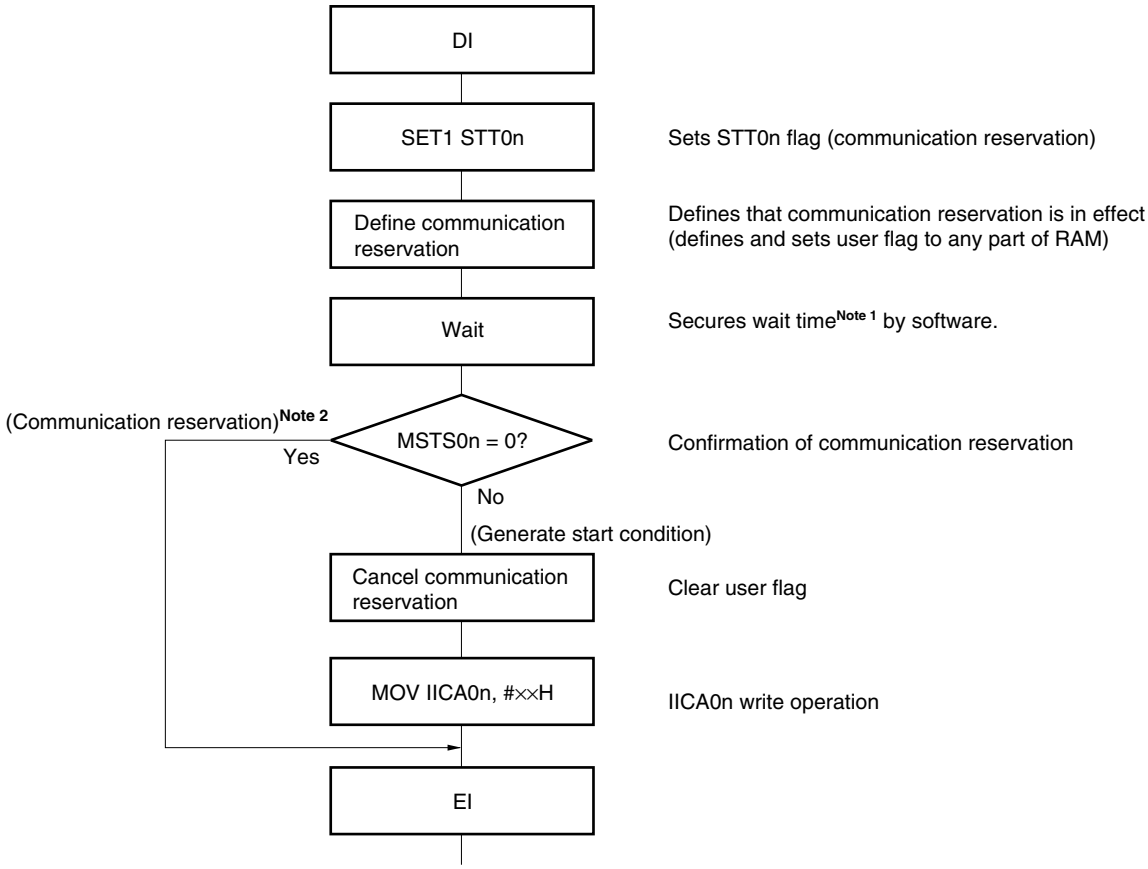


Figure 16-26 shows the communication reservation protocol.

Remark IICA0n: IICA shift register 0n
STT0n: Bit 1 of IICA control register 0n (IICACTL0n)
STD0n: Bit 1 of IICA status register 0n (IICAS0n)
SPD0n: Bit 0 of IICA status register 0n (IICAS0n)

Figure 16-26. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICAWL0n setting value + IICAWH0n setting value + 4) + $t_F \times 2 \times f_{PRS}$ (clocks)

2. The communication reservation operation executes a write to the IICA shift register 0n (IICA0n) when a stop condition interrupt request occurs.

Remarks 1. STT0n: Bit 1 of IICA control register 0n (IICACTL0n)
MSTS0n: Bit 7 of IICA status register 0n (IICAS0n)
IICA0n: IICA shift register 0n
IICAWL0n: IICA low-level width setting register 0n
IICAWH0n: IICA high-level width setting register 0n
t_F: SDAAn and SCLAn signal falling times
(refer to **CHAPTER 30 ELECTRICAL SPECIFICATIONS**)
f_{PRS}: Peripheral hardware clock frequency

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register 0n (IICAF0n) = 1)

When bit 1 (STT0n) of IICA control register 0n (IICACTL0n) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL0n) of IICACTL0n to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF0n (bit 7 of IICAF0n). It takes up to 5 clocks until STCF0n is set to 1 after setting STT0n = 1. Therefore, secure the time by software.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5.15 Cautions

(1) When STCEN0n (bit 1 of IICA flag register 0n (IICAF0n)) = 0

Immediately after I^2C operation is enabled (IICE0n = 1), the bus communication status (IICBSY0n (bit 6 of IICAF0n) = 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 1n (IICACTL1n).
<2> Set bit 7 (IICE0n) of IICA control register 0n (IICACTL0n) to 1.
<3> Set bit 0 (SPT0n) of IICACTL0n to 1.

(2) When STCEN0n = 1

Immediately after I^2C operation is enabled (IICE0n = 1), the bus released status (IICBSY0n = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0n (bit 1 of IICA control register 0n (IICACTL0n)) = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(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 SDAAn pin is low and the SCLAn pin is high, the macro of I^2C recognizes that the SDAAn pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{ACK}$ is returned, but this interferes with other I^2C communications. To avoid this, start I^2C in the following sequence.

<1> Clear bit 4 (SPIE0n) of IICACTL0n to 0 to disable generation of an interrupt request signal (INTIICAn) when the stop condition is detected.
<2> Set bit 7 (IICE0n) of IICACTL0n to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0n) of IICACTL0n to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting IICE0n to 1), to forcibly disable detection.

(4) Setting STT0n and SPT0n (bits 1 and 0 of IICACTL0n) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set SPIE0n (bit 4 of IICACTL0n) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IICA0n 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 SPIE0n to 1 when MSTS0n (bit 7 of IICAS0n) is detected by software.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0/Kx2-C microcontrollers 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 78K0/Kx2-C microcontrollers take 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 78K0/Kx2-C microcontrollers lose 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 78K0/Kx2-C microcontrollers are used as the I^2C bus slave is shown below.

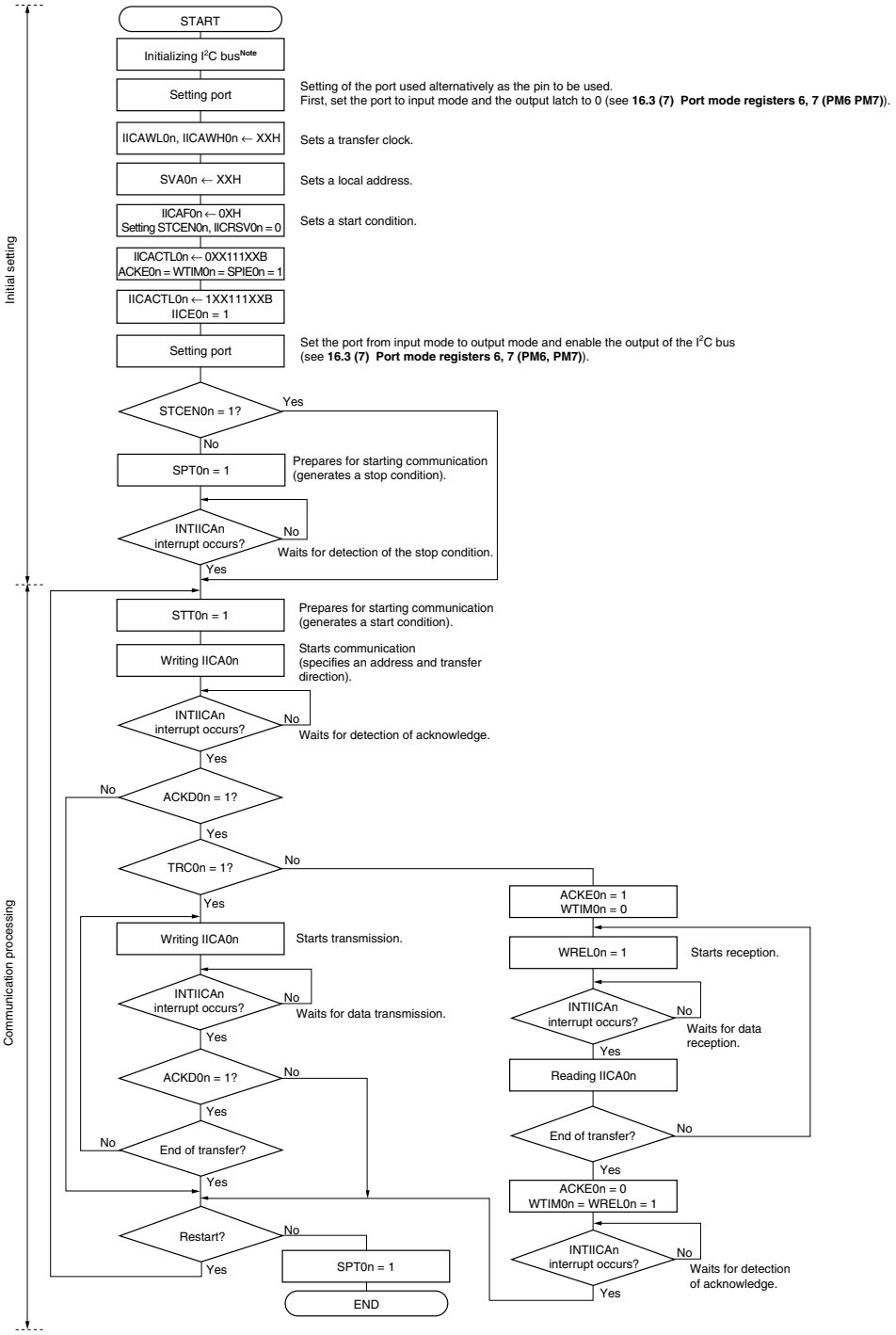
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(1) Master operation in single-master system

Figure 16-27. Master Operation in Single-Master System

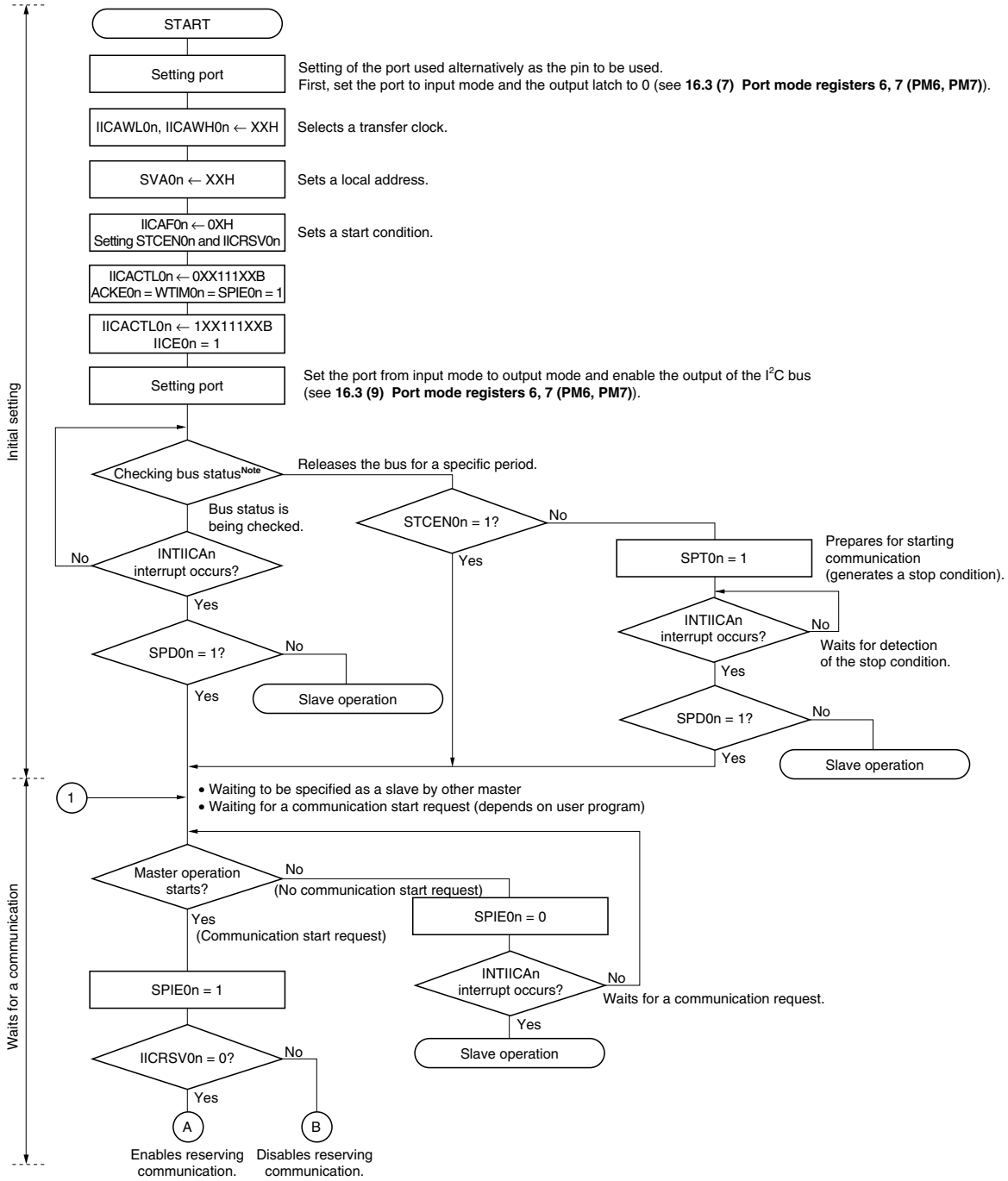


Note Release (SCLAn and SDAAn pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

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

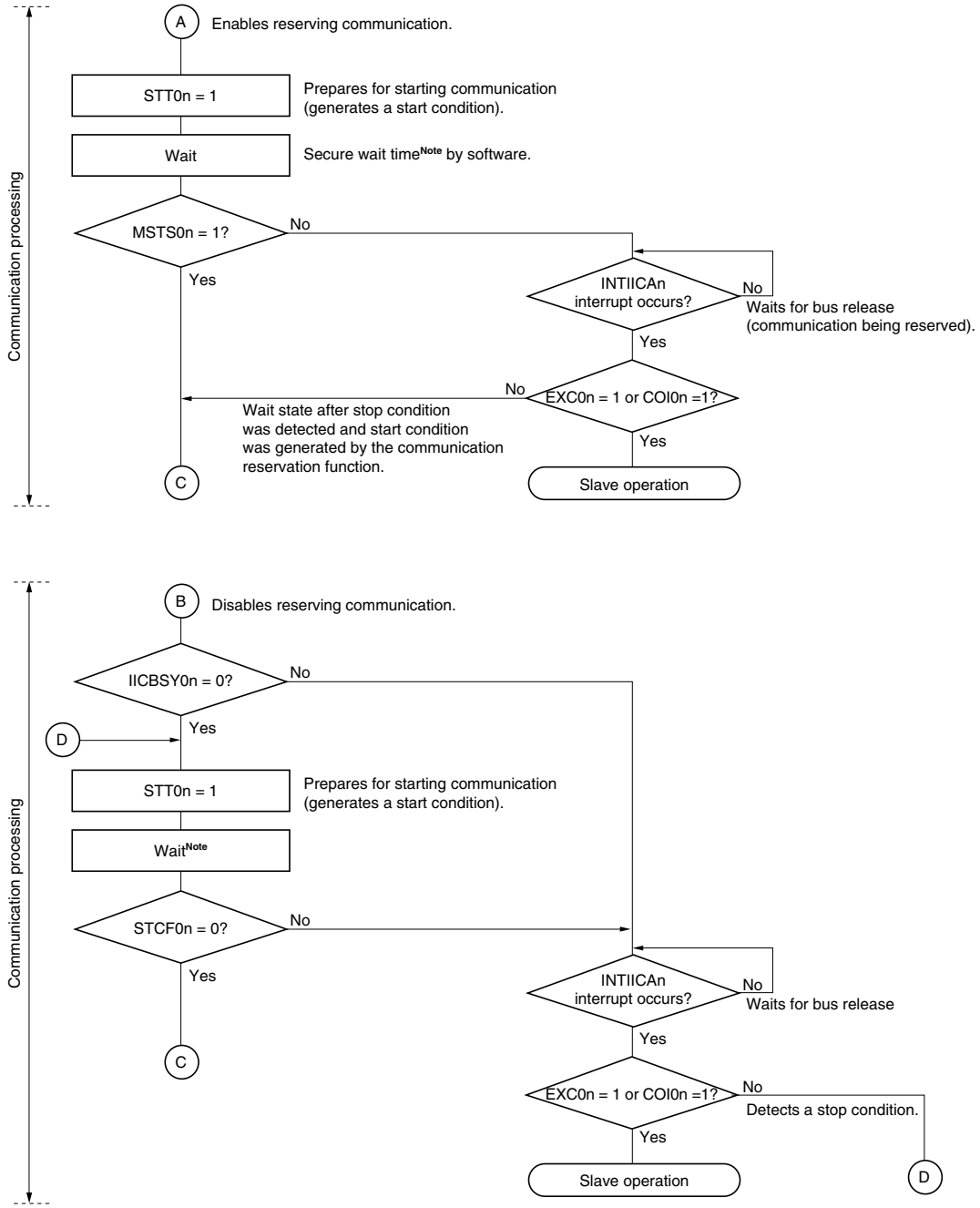
Figure 16-28. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD0n bit = 1, DAD0n bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-28. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.

(IICAWL0n setting value + IICAWH0n setting value + 4) + $t_F \times 2 \times f_{PRS}$ (clocks)

Remarks 1. IICAWL0n: IICA low-level width setting register 0n

IICAWH0n: IICA high-level width setting register 0n

t_F: SDAAn and SCLAn signal falling times

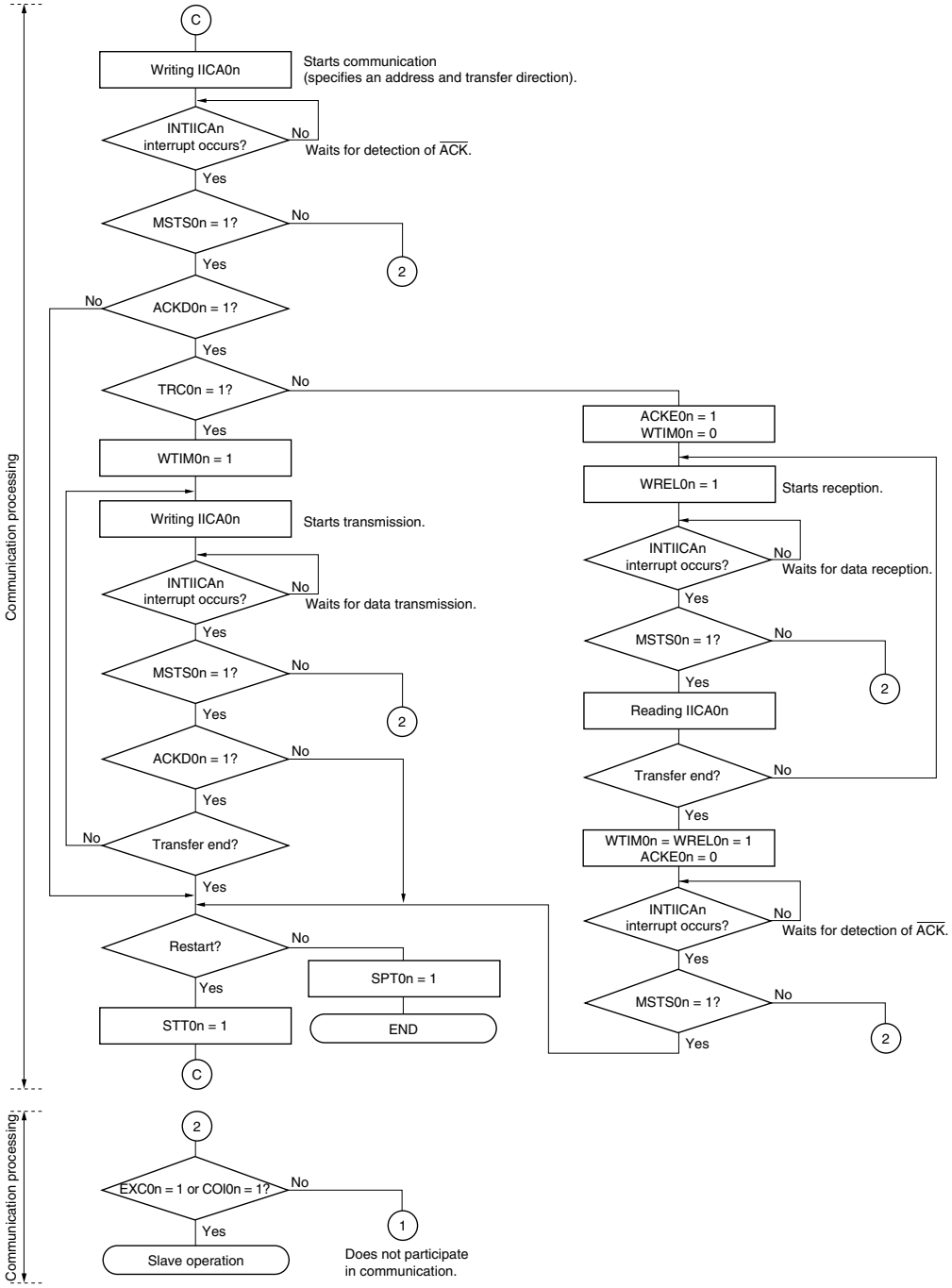
f_{PRS}: Peripheral hardware clock frequency

2. Serial interface IICA00: n = 0

Serial interface IICA01: n = 1

Serial interface IICA02: n = 2

Figure 16-28. 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 MSTS0n bit each time interrupt INTIICAn 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 IICAS0n and IICAF0n registers each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

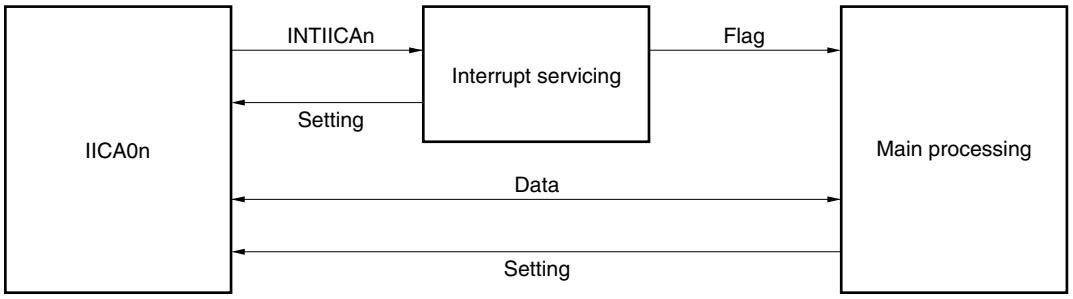
4. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICAn.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as TRC0n.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

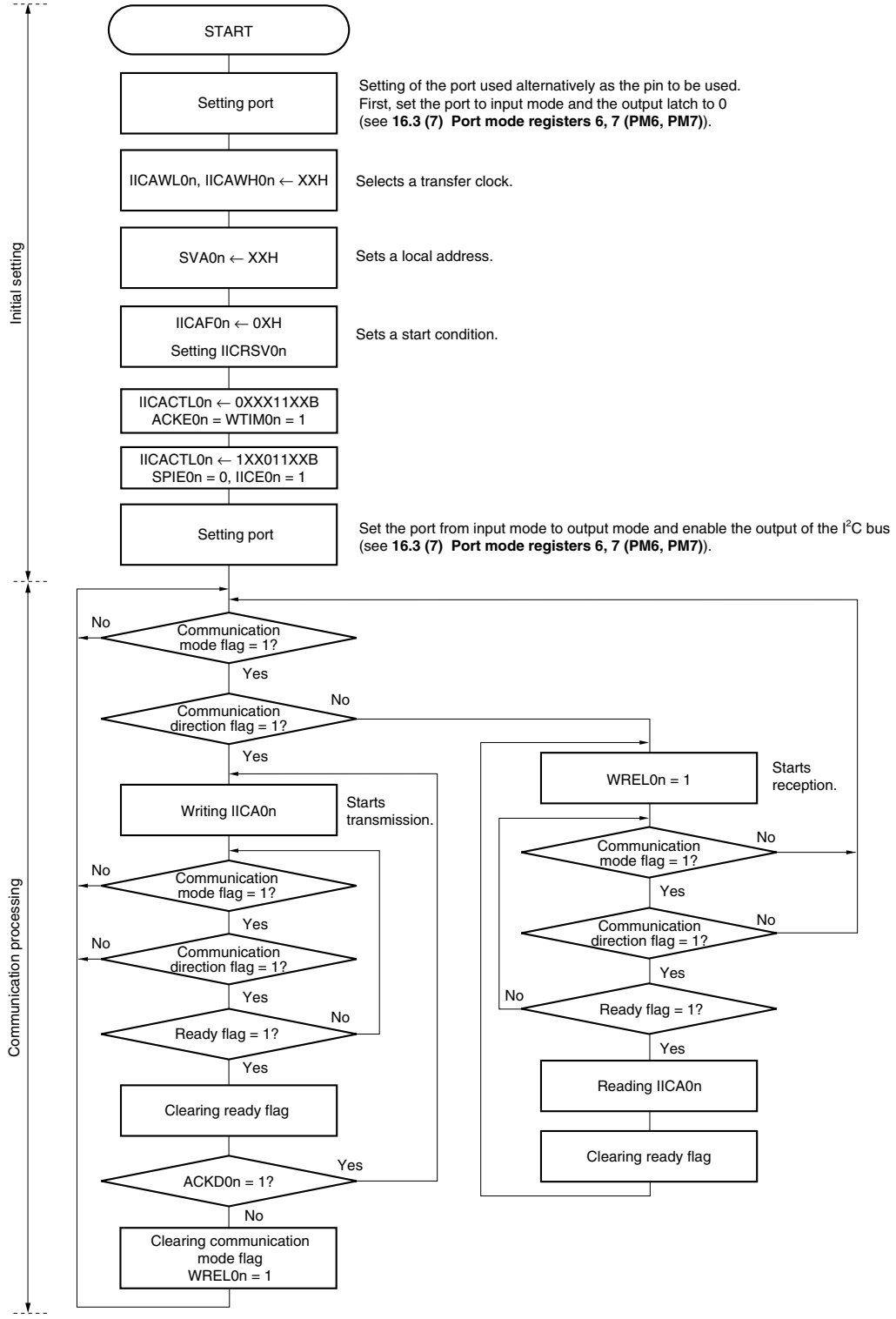
The main processing of the slave operation is explained next.

Start serial interface IICA0n 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 16-29. Slave Operation Flowchart (1)



Remarks 1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

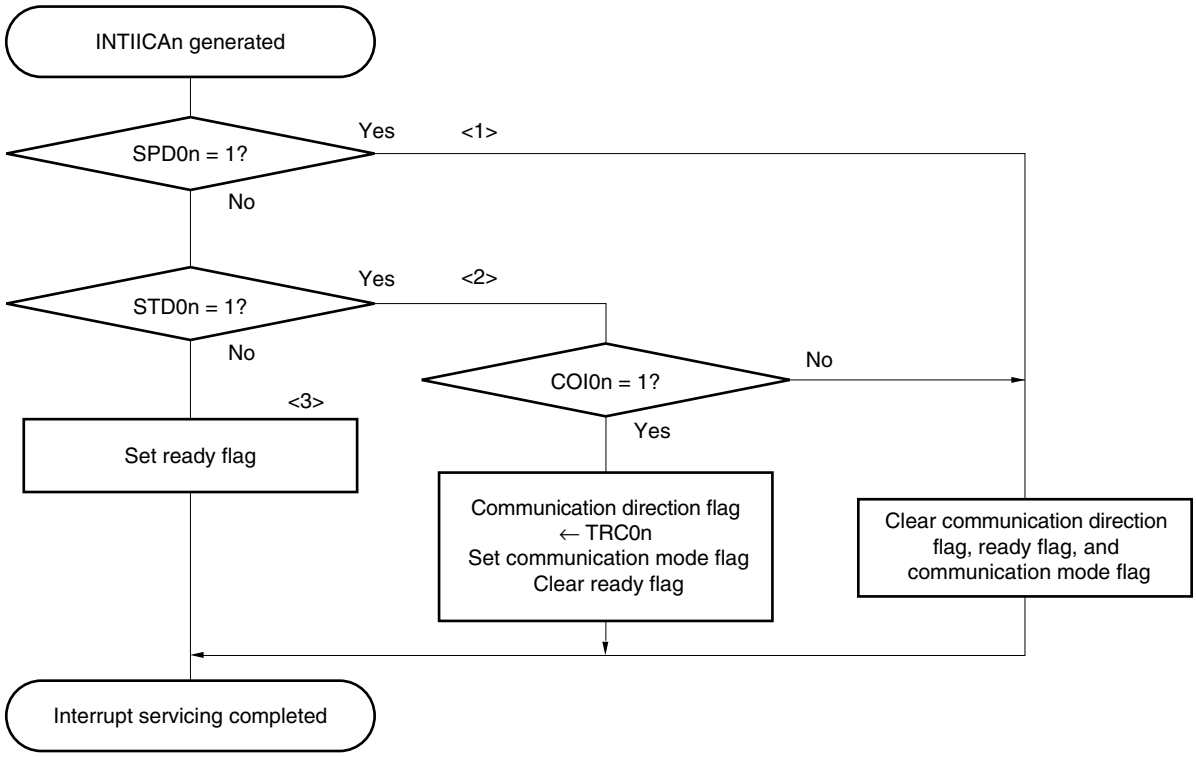
2. Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICAn interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 16-30 Slave Operation Flowchart (2).

Figure 16-30. Slave Operation Flowchart (2)



Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

16.5.17 Timing of I²C interrupt request (INTIICAn) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICAS0n register when the INTIICAn signal is generated are shown below.

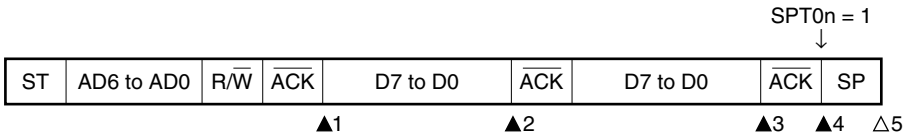
Remarks 1. ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

2. Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0n = 0

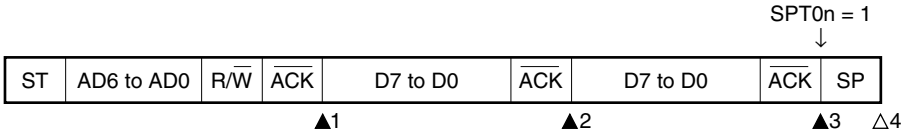


▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×000B
▲3: IICAS0n = 1000×000B (Sets WTIM0n to 1)[Note]
▲4: IICAS0n = 1000××00B (Sets SPT0n to 1)[Note]
△5: IICAS0n = 00000001B

Note To generate a stop condition, set WTIM0n to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1

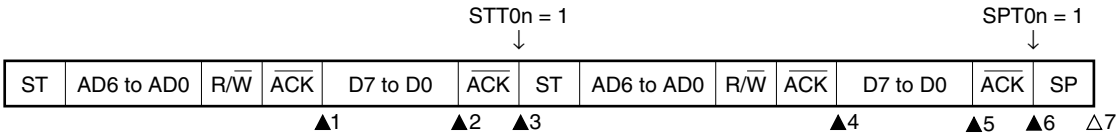


▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×100B
▲3: IICAS0n = 1000××00B (Sets SPT0n to 1)
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0n = 0

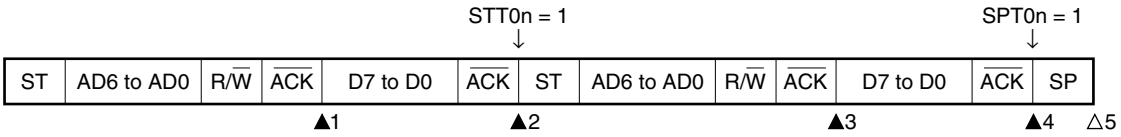


▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×000B (Sets WTIM0n to 1)[Note 1]
▲3: IICAS0n = 1000××00B (Clears WTIM0n to 0[Note 2], sets STT0n to 1)
▲4: IICAS0n = 1000×110B
▲5: IICAS0n = 1000×000B (Sets WTIM0n to 1)[Note 3]
▲6: IICAS0n = 1000××00B (Sets SPT0n to 1)
△7: IICAS0n = 00000001B

Notes 1. To generate a start condition, set WTIM0n to 1 and change the timing for generating the INTIICAn interrupt request signal.
2. Clear WTIM0n to 0 to restore the original setting.
3. To generate a stop condition, set WTIM0n to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1

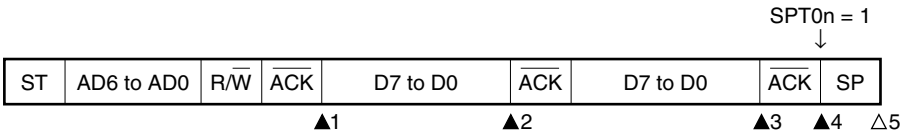


▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000××00B (Sets STT0n to 1)
▲3: IICAS0n = 1000×110B
▲4: IICAS0n = 1000××00B (Sets SPT0n to 1)
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0n = 0

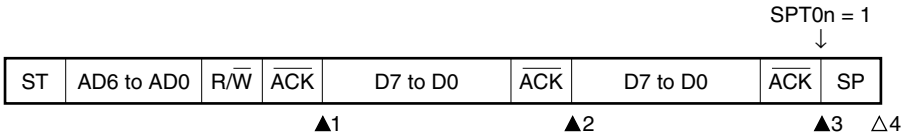


▲1: IICAS0n = 1010×110B
▲2: IICAS0n = 1010×000B
▲3: IICAS0n = 1010×000B (Sets WTIM0n to 1)[Note]
▲4: IICAS0n = 1010××00B (Sets SPT0n to 1)
△5: IICAS0n = 00000001B

Note To generate a stop condition, set WTIM0n to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1



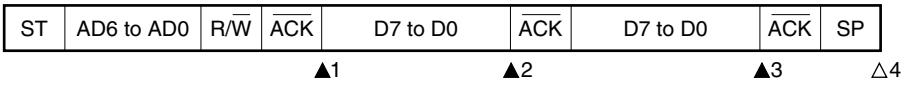
▲1: IICAS0n = 1010×110B
▲2: IICAS0n = 1010×100B
▲3: IICAS0n = 1010××00B (Sets SPT0n to 1)
△4: IICAS0n = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

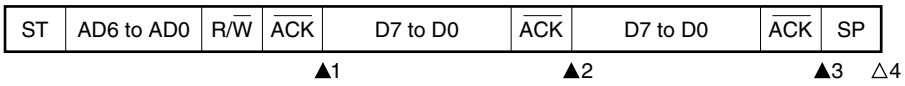
(i) When WTIM0n = 0



▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001×000B
▲3: IICAS0n = 0001×000B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1

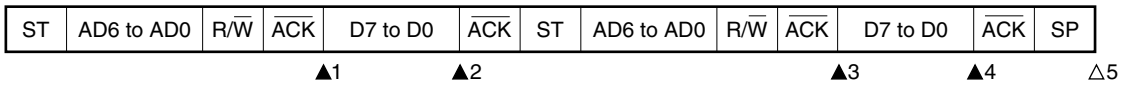


▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001×100B
▲3: IICAS0n = 0001××00B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

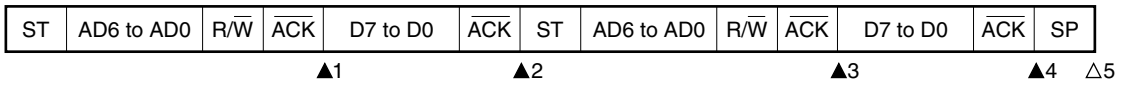
(i) When WTIM0n = 0 (after restart, matches with SVA0n)



▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001×000B
▲3: IICAS0n = 0001×110B
▲4: IICAS0n = 0001×000B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1 (after restart, matches with SVA0n)

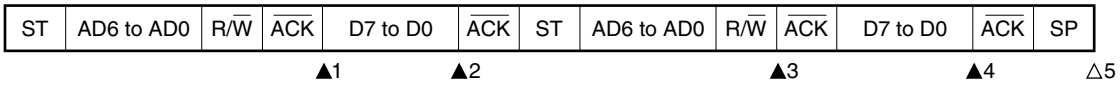


▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001××00B
▲3: IICAS0n = 0001×110B
▲4: IICAS0n = 0001××00B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

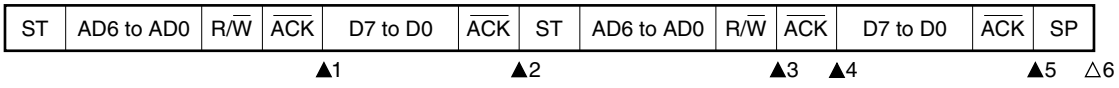
(i) When WTIM0n = 0 (after restart, does not match address (= extension code))



▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001×000B
▲3: IICAS0n = 0010×010B
▲4: IICAS0n = 0010×000B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1 (after restart, does not match address (= extension code))

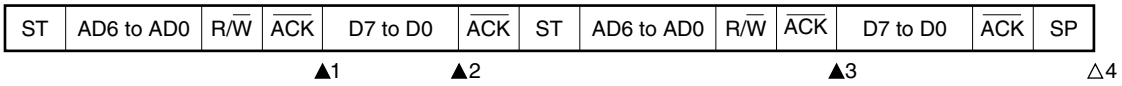


▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001××00B
▲3: IICAS0n = 0010×010B
▲4: IICAS0n = 0010×110B
▲5: IICAS0n = 0010××00B
△6: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0n = 0 (after restart, does not match address (= not extension code))

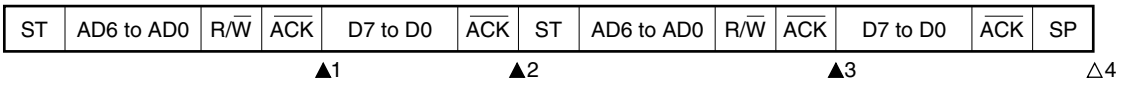


▲1 ▲2 ▲3 △4

▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001×000B
▲3: IICAS0n = 00000110B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1 (after restart, does not match address (= not extension code))



▲1 ▲2 ▲3 △4

▲1: IICAS0n = 0001×110B
▲2: IICAS0n = 0001××00B
▲3: IICAS0n = 00000110B
△4: IICAS0n = 00000001B

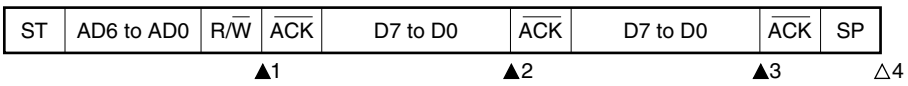
Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(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

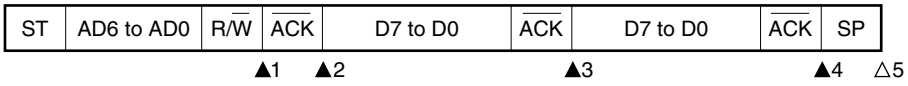
(i) When WTIM0n = 0



▲1: IICAS0n = 0010×010B
▲2: IICAS0n = 0010×000B
▲3: IICAS0n = 0010×000B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1

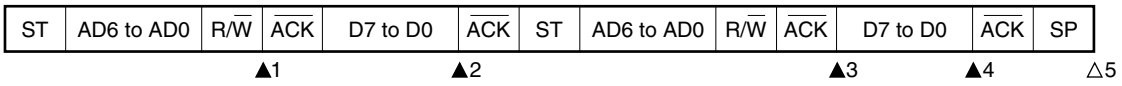


▲1: IICAS0n = 0010×010B
▲2: IICAS0n = 0010×110B
▲3: IICAS0n = 0010×100B
▲4: IICAS0n = 0010××00B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

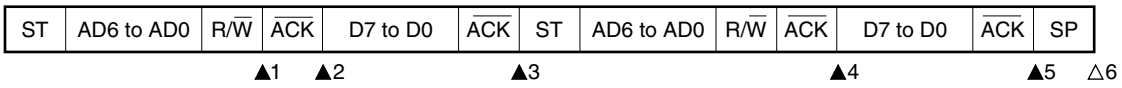
(i) When WTIM0n = 0 (after restart, matches SVA0n)



▲1: IICAS0n = 0010×010B
▲2: IICAS0n = 0010×000B
▲3: IICAS0n = 0001×110B
▲4: IICAS0n = 0001×000B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1 (after restart, matches SVA0n)

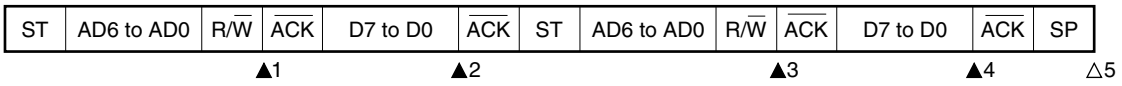


▲1: IICAS0n = 0010×010B
▲2: IICAS0n = 0010×110B
▲3: IICAS0n = 0010××00B
▲4: IICAS0n = 0001×110B
▲5: IICAS0n = 0001××00B
△6: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

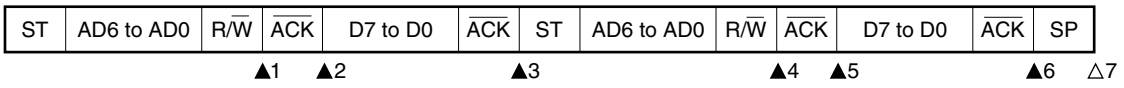
(i) When WTIM0n = 0 (after restart, extension code reception)



▲1: IICAS0n = 0010×010B
▲2: IICAS0n = 0010×000B
▲3: IICAS0n = 0010×010B
▲4: IICAS0n = 0010×000B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1 (after restart, extension code reception)



▲1: IICAS0n = 0010×010B
▲2: IICAS0n = 0010×110B
▲3: IICAS0n = 0010××00B
▲4: IICAS0n = 0010×010B
▲5: IICAS0n = 0010×110B
▲6: IICAS0n = 0010××00B
△7: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

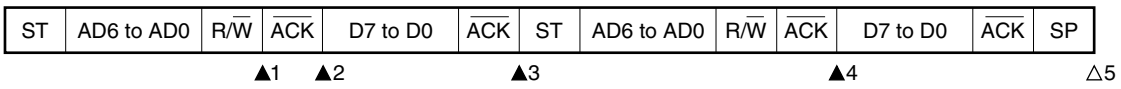
(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0n = 0 (after restart, does not match address (= not extension code))

▲1: IICAS0n = 00100010B
▲2: IICAS0n = 00100000B
▲3: IICAS0n = 00000110B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1 (after restart, does not match address (= not extension code))

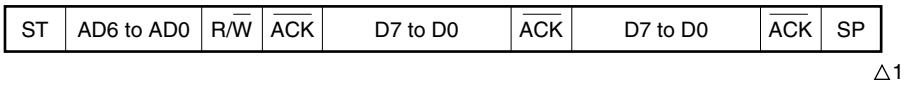


▲1: IICAS0n = 00100010B
▲2: IICAS0n = 00100110B
▲3: IICAS0n = 00100×00B
▲4: IICAS0n = 00000110B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICAS0n = 00000001B

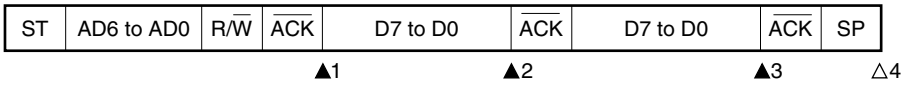
Remark △: Generated only when SPIE0n = 1

(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 MSTS0n bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

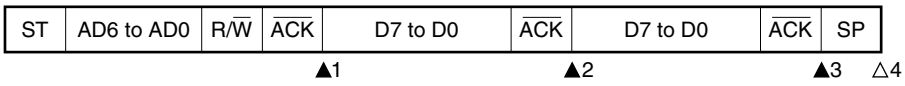
(i) When WTIM0n = 0



▲1: IICAS0n = 0101×110B
▲2: IICAS0n = 0001×000B
▲3: IICAS0n = 0001×000B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1



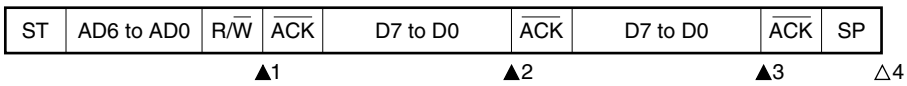
▲1 ▲2 ▲3 △4

▲1: IICAS0n = 0101×110B
▲2: IICAS0n = 0001×100B
▲3: IICAS0n = 0001××00B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIM0n = 0

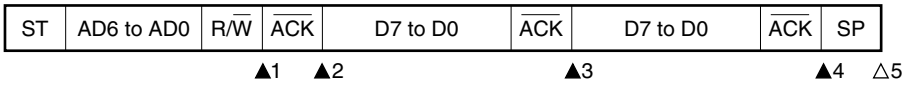


▲1 ▲2 ▲3 △4

▲1: IICAS0n = 0110×010B
▲2: IICAS0n = 0010×000B
▲3: IICAS0n = 0010×000B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1



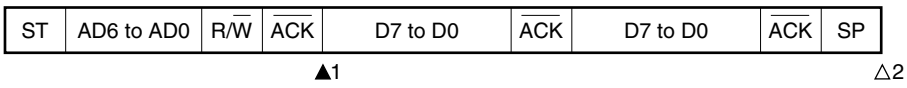
▲1: IICAS0n = 0110×010B
▲2: IICAS0n = 0010×110B
▲3: IICAS0n = 0010×100B
▲4: IICAS0n = 0010××00B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 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 MSTS0n bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

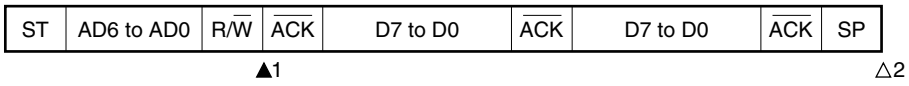
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0n = 1)



▲1: IICAS0n = 01000110B
△2: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1

(b) When arbitration loss occurs during transmission of extension code

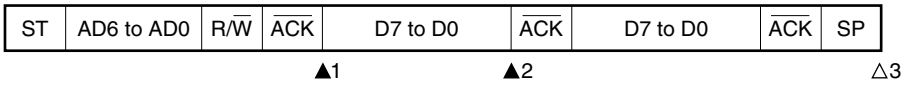


▲1: IICAS0n = 0110×010B
Sets LREL0n = 1 by software
△2: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

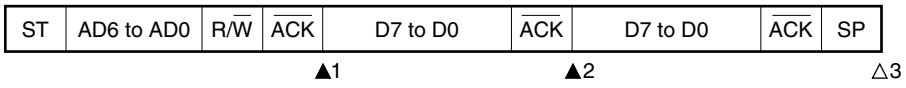
(i) When WTIM0n = 0



▲1: IICAS0n = 10001110B
▲2: IICAS0n = 01000000B
△3: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1

(ii) When WTIM0n = 1

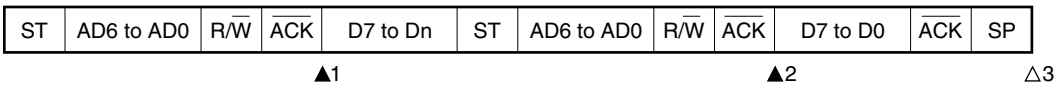


▲1: IICAS0n = 10001110B
▲2: IICAS0n = 01000100B
△3: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1

(d) When loss occurs due to restart condition during data transfer

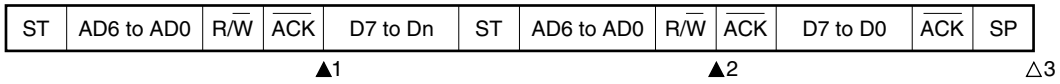
(i) Not extension code (Example: unmatches with SVA0n)



▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 01000110B
△3: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care
n = 6 to 0

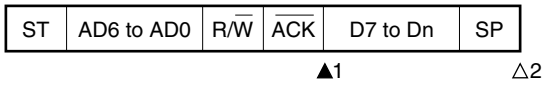
(ii) Extension code



▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 01100010B
Sets LREL0n = 1 by software
△3: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care
n = 6 to 0

(e) When loss occurs due to stop condition during data transfer

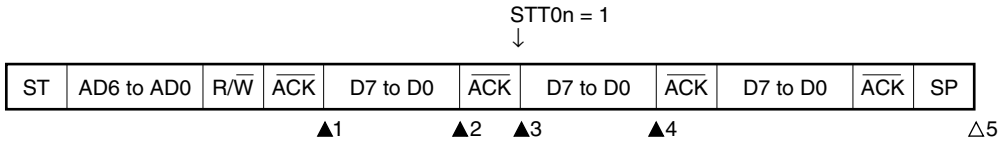


▲1: IICAS0n = 10000110B
△2: IICAS0n = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 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

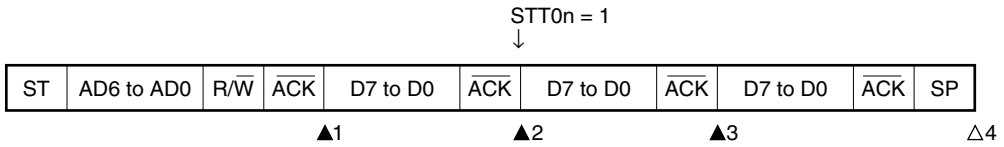
(i) When WTIM0n = 0



▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×000B (Sets WTIM0n to 1)
▲3: IICAS0n = 1000×100B (Clears WTIM0n to 0)
▲4: IICAS0n = 01000000B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1

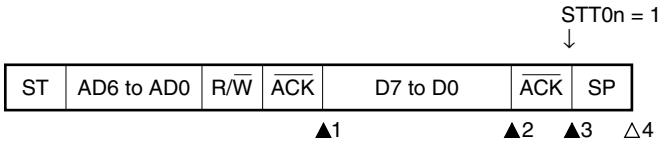


▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×100B (Sets STT0n to 1)
▲3: IICAS0n = 01000100B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

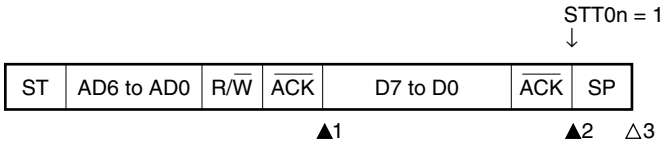
(i) When WTIM0n = 0



▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×000B (Sets WTIM0n to 1)
▲3: IICAS0n = 1000××00B (Sets STT0n to 1)
△4: IICAS0n = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1

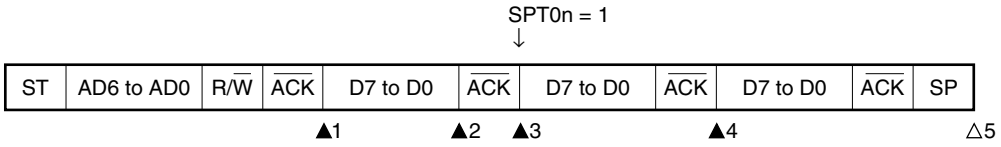


▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000××00B (Sets STT0n to 1)
△3: IICAS0n = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

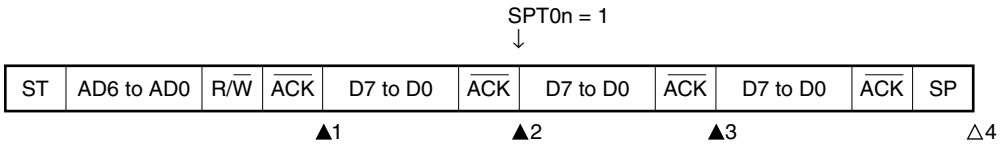
(i) When WTIM0n = 0



▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×000B (Sets WTIM0n to 1)
▲3: IICAS0n = 1000×100B (Clears WTIM0n to 0)
▲4: IICAS0n = 01000100B
△5: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

(ii) When WTIM0n = 1



▲1: IICAS0n = 1000×110B
▲2: IICAS0n = 1000×100B (Sets SPT0n to 1)
▲3: IICAS0n = 01000100B
△4: IICAS0n = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0n = 1
×: Don't care

16.6 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 TRC0n bit (bit 3 of the IICA status register 0n (IICAS0n)), which specifies the data transfer direction, and then starts serial communication with the slave device.

Figures 16-31 and 16-32 show timing charts of the data communication.

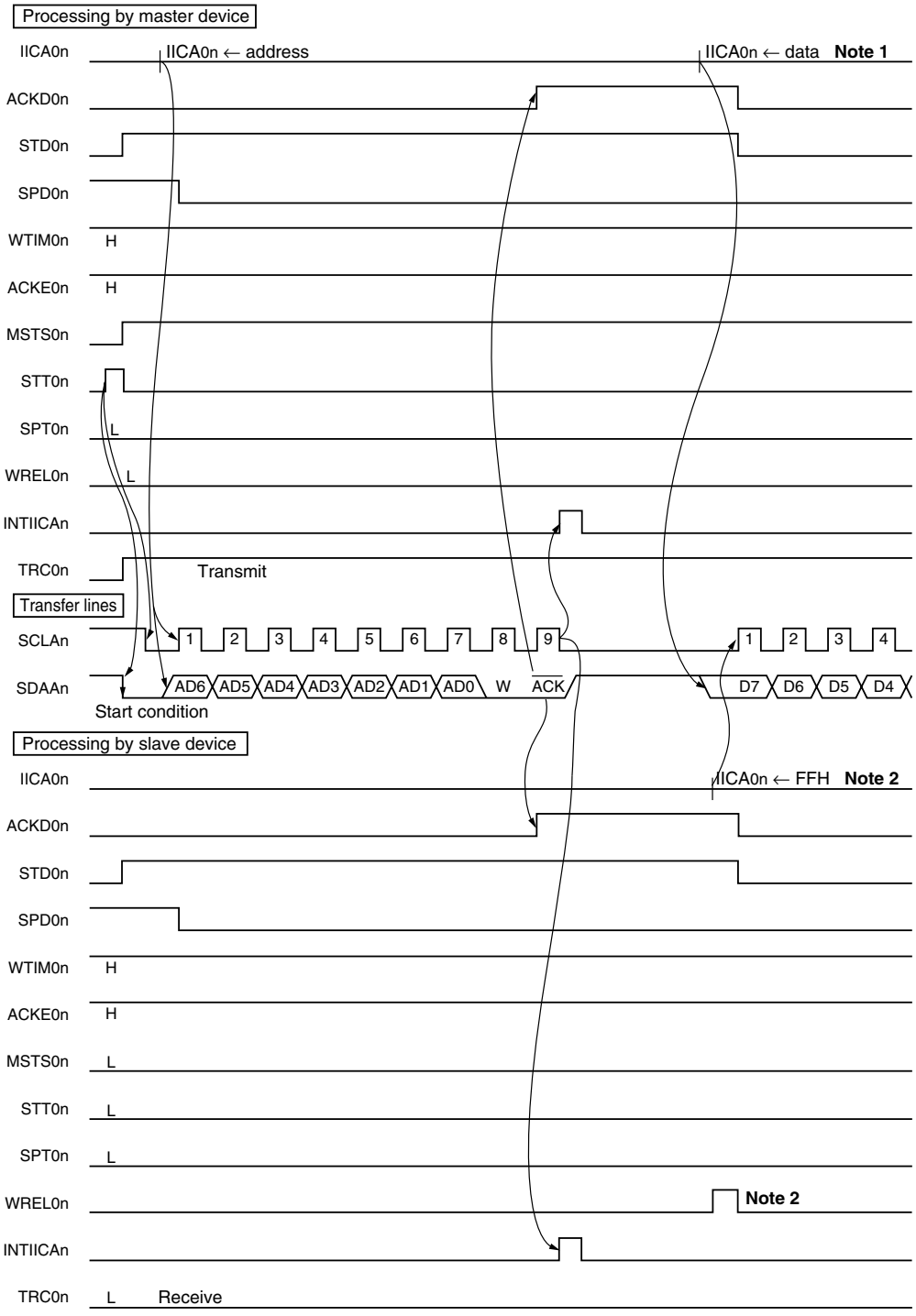
The IICA shift register 0n (IICA0n)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

Data input via the SDAAn pin is captured into IICA0n at the rising edge of SCLAn.

Remark Serial interface IICA00: n = 0
Serial interface IICA01: n = 1
Serial interface IICA02: n = 2

Figure 16-31. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(1) Start condition ~ address



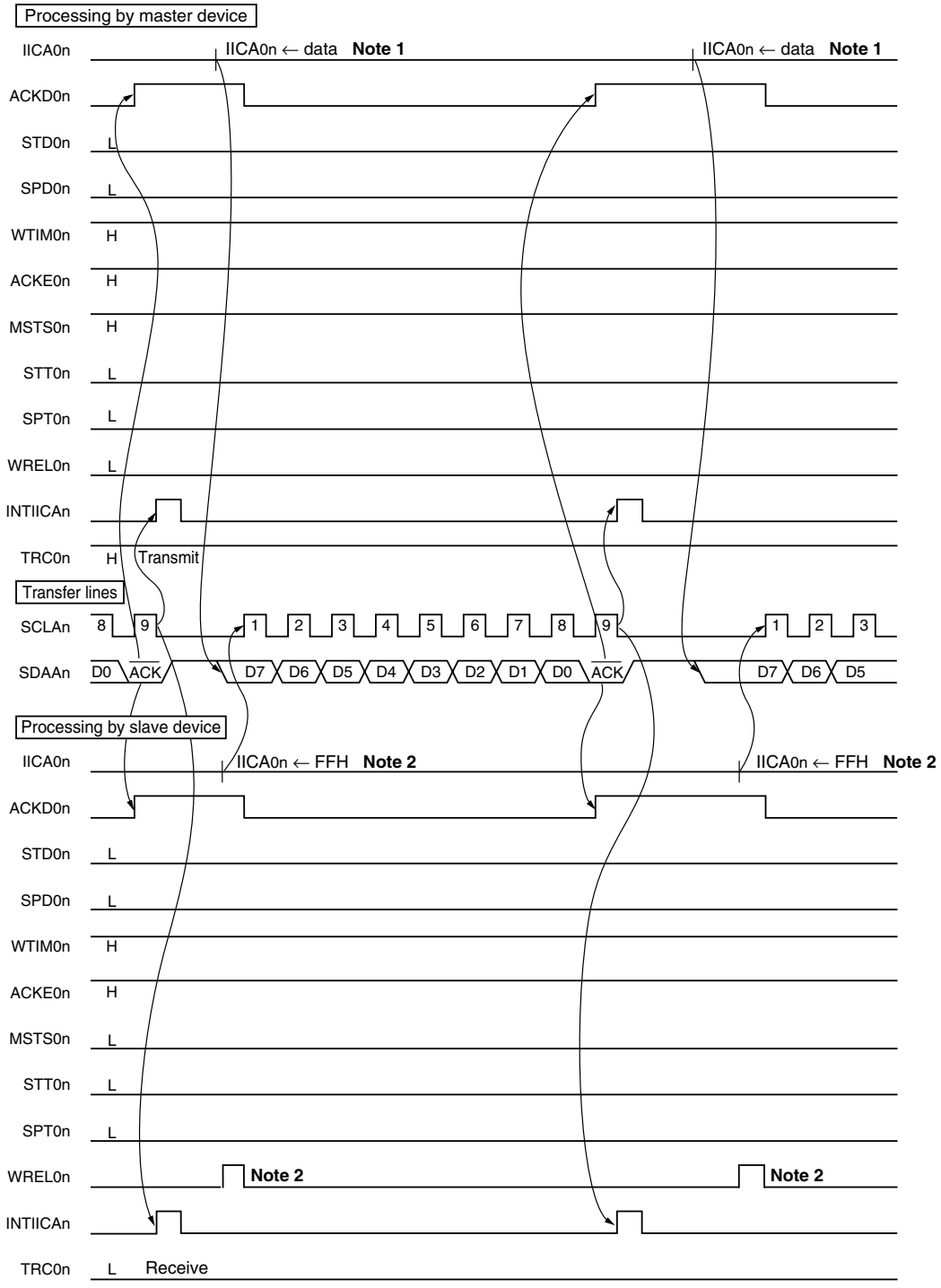
Notes 1. Write data to IICA0n, not setting WREL0n, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA0n or set WREL0n.

Remark Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

Figure 16-31. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(2) Data



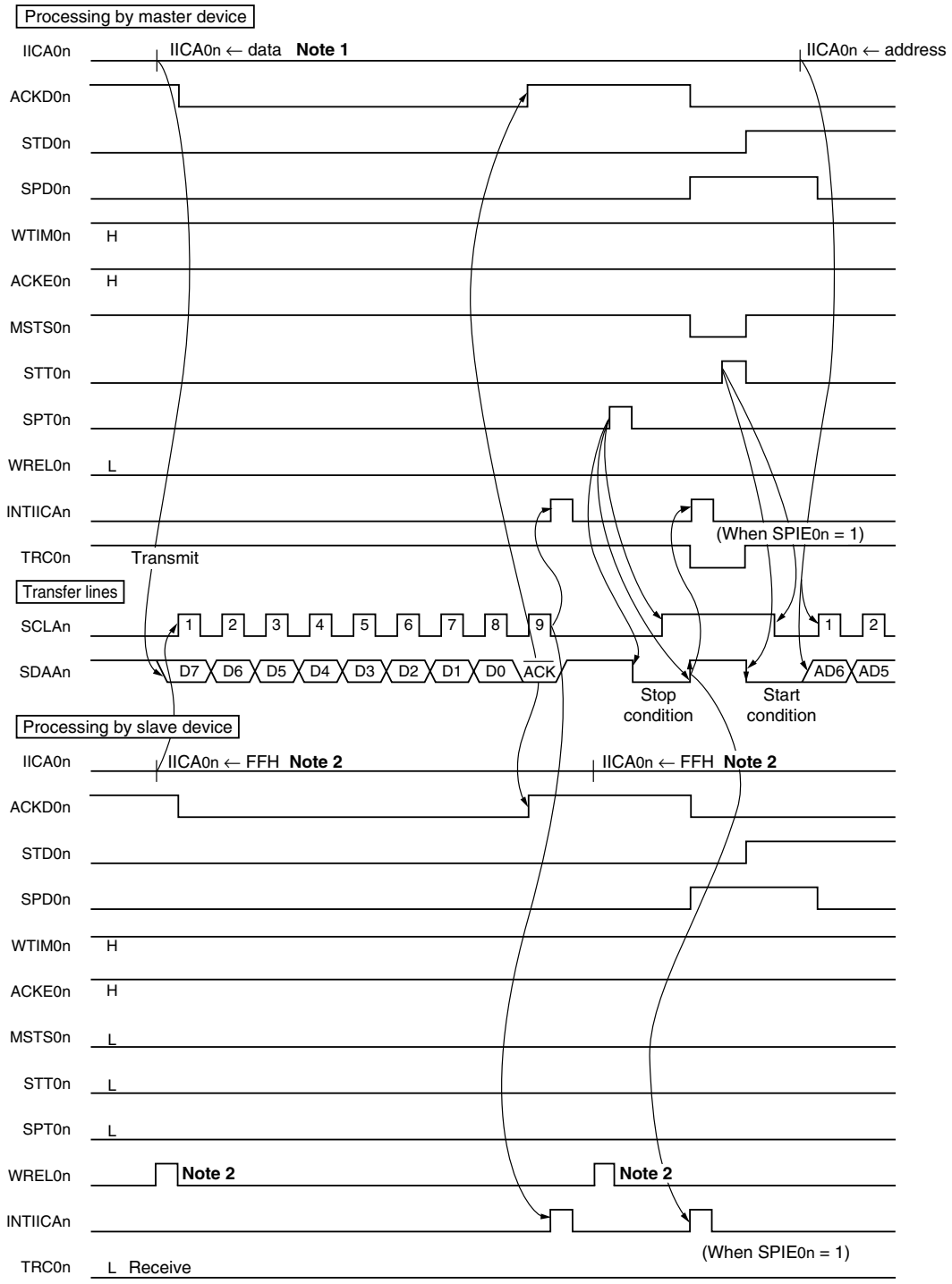
Notes 1. Write data to IICA0n, not setting WREL0n, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write "FFH" to IICA0n or set WREL0n.

Remark Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

Figure 16-31. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition



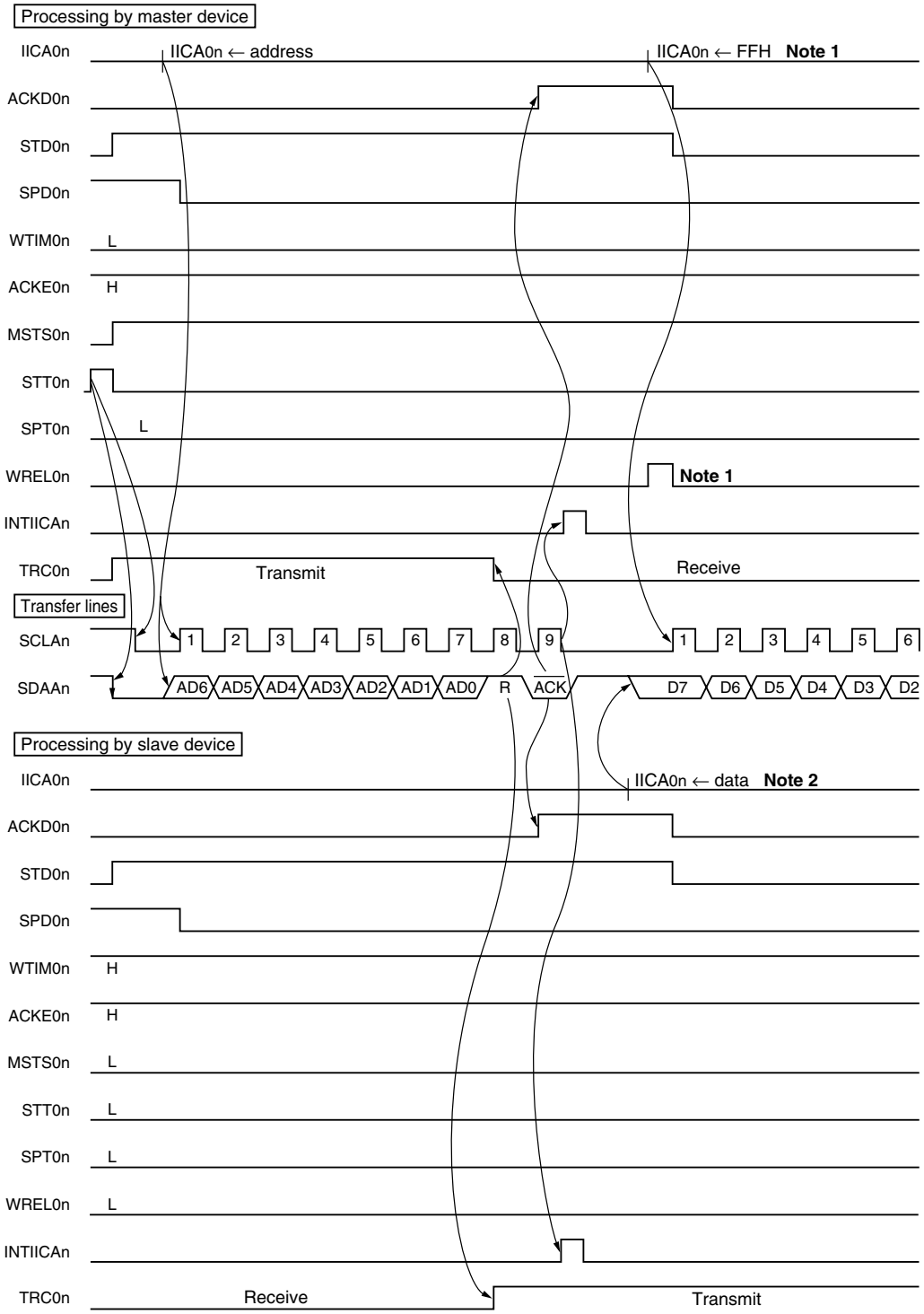
Notes 1. Write data to IICA0n, not setting WREL0n, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write "FFH" to IICA0n or set WREL0n.

Remark Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

Figure 16-32. 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



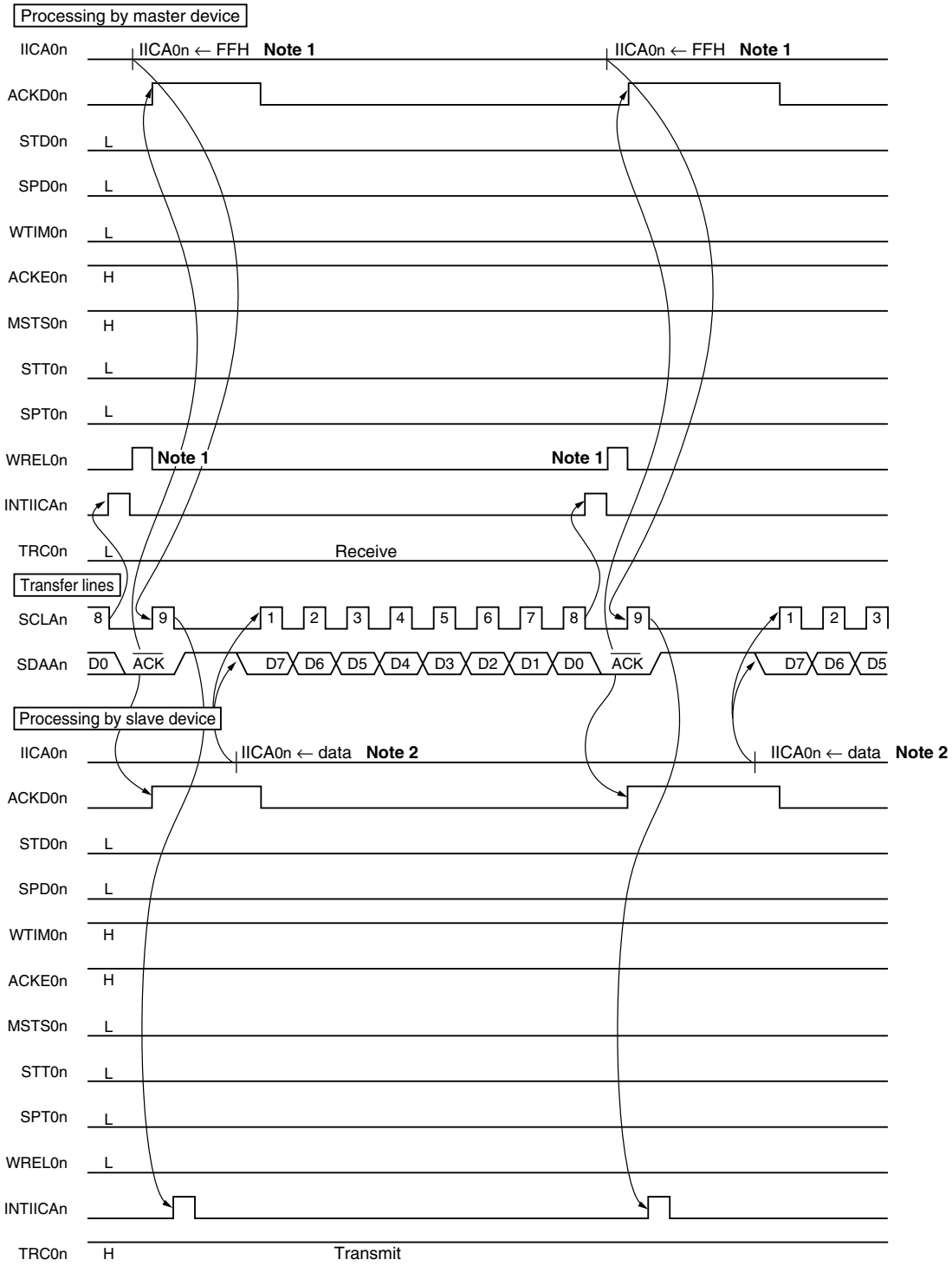
Notes 1. To cancel master wait, write “FFH” to IICA0n or set WREL0n.

2. Write data to IICA0n, not setting WREL0n, in order to cancel a wait state during slave transmission.

Remark Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

Figure 16-32. 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



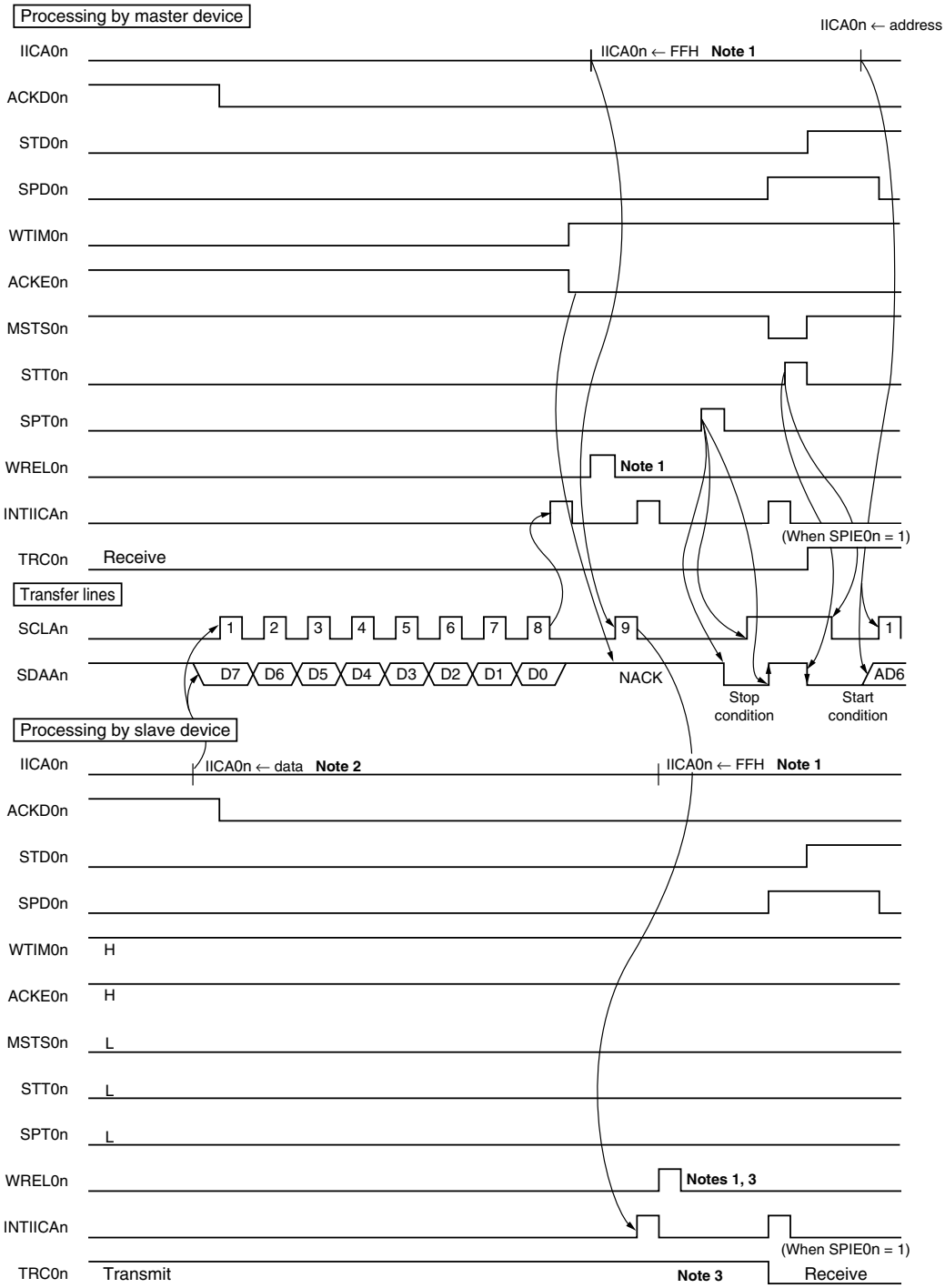
Notes 1. To cancel master wait, write “FFH” to IICA0n or set WREL0n.

2. Write data to IICA0n, not setting WREL0n, in order to cancel a wait state during slave transmission.

Remark Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

Figure 16-32. 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 IICA0n or set WREL0n.

2. Write data to IICA0n, not setting WREL0n, in order to cancel a wait state during slave transmission.

3. If a wait state during slave transmission is canceled by setting WREL0n, TRC0n will be cleared.

Remark Serial interface IICA00: n = 0, Serial interface IICA01: n = 1,
Serial interface IICA02: n = 2

CHAPTER 17 CEC TRANSMISSION/RECEPTION CIRCUIT

Caution If CEC transmission/reception circuit is used, INTP4 and INTP5 cannot be used.

17.1 Functions of CEC Transmission/Reception Circuit

The CEC transmission/reception circuit can generate and receive CEC signals conforming to the CEC (Consumer Electronics Control) standard, as well as automatically detect communication statuses by hardware. CEC transmission/reception can be easily controlled by using these functions.

- Serial communication conforming to the CEC standard can be performed.
- The operating clock can be selected from the main system clock and subsystem clock.
- The low-level width/bit width of the start bit and data bit can be set to different values for transmission and reception.
- Errors and communication statuses can be detected by hardware.
- Signal-free time can be counted.

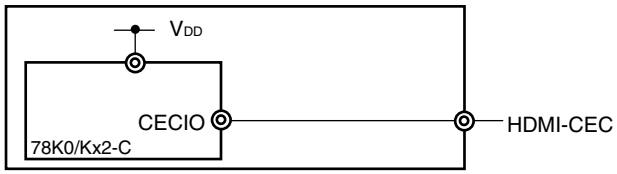
<R> **(1) About the setting of each mode**

(a) TYPE1: When using the internal pull-up resistors and diodes

- Using P72
- Using the power supply at V_{DD} = 3.3 V±10%
- Not needing external circuits to the microcontroller.

P72	PM72	PF72	PIM72	PU72
1	0	1	1	1

PF72 = 1: CECIO mode
PIM72 = 1: CEC input buffer
PU72 = 1: Connects pull-up resistor + diode

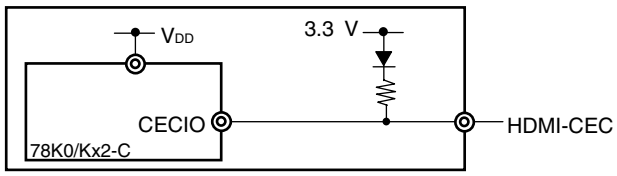


(b) TYPE2: When using the circuit connecting to external pull-up resistors and diodes

- Using P72
- Using the power supply at V_{DD} = 3.3 V±10%
- Needing to connect external pull-up resistors and diodes to the microcontroller

P72	PM72	PF72	PIM72	PU72
1	0	1	1	0

PF72 = 1: CECIO mode
PIM72 = 1: CEC input buffer
PU72 = 0: Do not connected pull-up resistor + diode

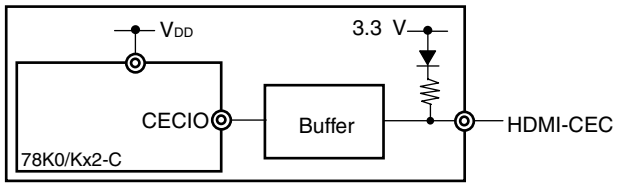


(c) TYPE3: When using the CEC function out of range of V_{DD} = 3.3 V±10%

- Using P72
- Requiring the external circuit to convert the CECIO signal voltage to that on CEC standard value
- Requiring the external circuit to satisfy the electrical characteristics of normal input buffer, before the signal being inputted to the CECIO pin
- This type can be used at V_{DD} = 1.8 to 5.5 V

P72	PM72	PF72	PIM72	PU72
1	0	1	0	0

PF72 = 1: CECIO mode
PIM72 = 0: Normal input buffer
PU72 = 0: Do not connected pull-up resistor + diode



(d) TYPE4: When using the CEC function with connecting to CEC input and output separately

- Using P40 and P41
- Being used when the CEC functions are used separately divided into CEC input and output
- Requiring the external circuit to satisfy the electrical characteristics of normal input buffer, before the signal being inputted to the CECIN pin
- Requiring the external circuit to convert the CECOUT signal voltage to that on CEC standard value
- This type can be used at V_{DD} = 1.8 to 5.5 V

P41	PM41	P40	PM40	PF40	PF41
–	1	1	0	1	1

PF40 = 1: CECIN mode
PF41 = 1: CECOUT mode

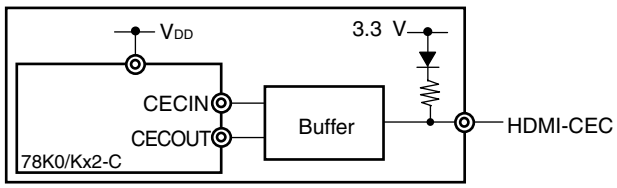
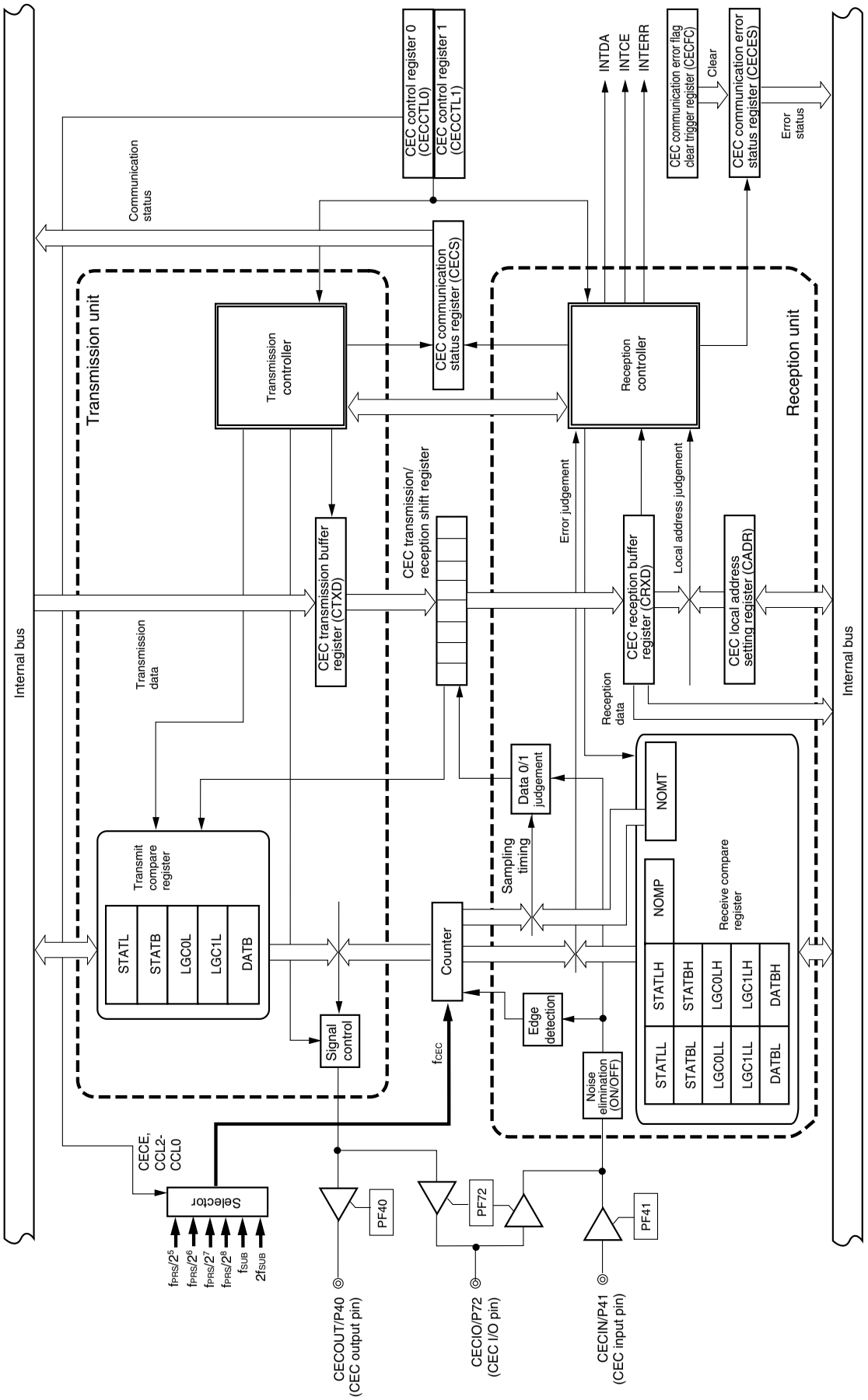


Figure 17-1. Block Diagram of CEC transmission/reception circuit



17.2 Configuration of CEC Transmission/Reception Circuit

The CEC transmission/reception circuit includes the following hardware.

Table 17-1. Configuration of CEC Transmission/Reception Circuit

Item	Configuration
Control registers	CEC control register 0 (CECCTL0) CEC control register 1 (CECCTL1) CEC communication status register (CECS) CEC communication error status register (CECES) CEC communication error flag clear trigger register (CECFC) CEC local address setting register (CADR) Port function register 4 (PF4) Port function register 7 (PF7) Port mode register 4 (PM4) Port mode register 7 (PM7)
Compare registers[Notes 1, 2, 3]	CEC reception buffer register (CRXD) CEC transmission buffer register (CTXD) CEC reception start bit minimum low width setting register (STATLL) CEC reception start bit maximum low width setting register (STATLH) CEC reception start bit minimum bit width setting register (STATBL) CEC reception start bit maximum bit width setting register (STATBH) CEC reception logical 0 minimum low width setting register (LGC0LL) CEC reception logical 0 maximum low width setting register (LGC0LH) CEC reception logical 1 minimum low width setting register (LGC1LL) CEC reception logical 1 maximum low width setting register (LGC1LH) CEC reception data bit minimum bit width setting register (DATBL) CEC reception data bit maximum bit width setting register (DATBH) CEC transmission start bit low width setting register (STATL) CEC transmission start bit width setting register (STATB) CEC transmission logical 0 low width setting register (LGC0L) CEC transmission logical 1 low width setting register (LGC1L) CEC transmission data bit width setting register (DATB) CEC data bit reference width setting register (NOMP) CEC reception data sampling time setting register (NOMT)

Notes 1. 0 can be written to the registers, but the CEC transmission/reception circuit will not operate normally.

2. Set up all registers other than buffer registers even when only receiving or transmitting data.

3. Some of the register setting values must have a specific relationship in size. Set the registers such that the following relationships are observed.

- STATL < STATB
- LGC1L < LGC0L < DATB
- STATLL < STATLH
- STATBL < STATBH
- LGC0LL < LGC0LH
- LGC1LL < LGC1LH
- DATBL < DATBH

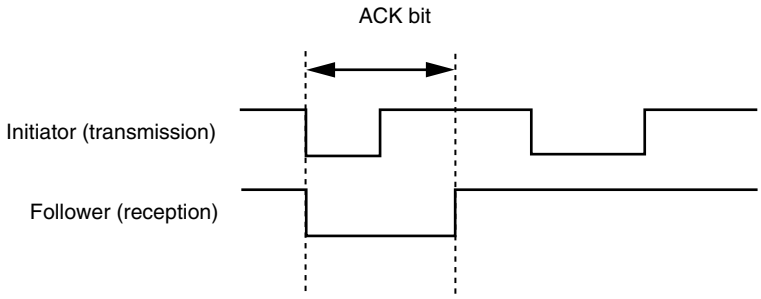
17.3 Term Description

- Initiator: Device that transmits or is transmitting CEC messages
- Follower: Device that receives or is receiving CEC messages
- Message: All data from the start bit to the operand
- Initiator address: Source address
- Destination address: Destination address
- Direct address communication (direct address message): Communication with one follower
- Broadcast communication (direct address message): Communication with multiple followers
- Arbitration: Prioritizing the devices that output a low level to the CEC line when multiple initiators exist
- Arbitration loss: State in which competing devices are prioritized. At this time, the local station stops transmitting.
- Bus free: State in which no communication is performed and transmission can be performed
- Bus busy: During communication
- Error handling: Outputting an error pulse (low level with a width of the bit width × 1.5) and transitioning to the communication standby state when a bit width shorter than the bit width of the data bit is received

The logic levels received at the ACK bit timing are as follows.

- ACK: Outputs logical 0.
- NACK: Outputs logical 1.

Example: If the initiator outputs logical 1 and the follower outputs logical 0 during an ACK bit period, the initiator transmits a NACK and the follower transmits an ACK.



17.4 Registers Controlling Transmission/Reception Circuit

The CEC transmission/reception circuit is controlled by the following registers.

- CEC local address setting register (CADR)
- CEC transmission buffer register (CTXD)
- CEC reception buffer register (CRXD)
- CEC control register 0 (CECCTL0)
- CEC control register 1 (CECCTL1)
- CEC communication status register (CECS)
- CEC communication error status register (CECES)
- CEC communication error flag clear trigger register (CECFC)
- Port function register 4 (PF4)
- Port function register 7 (PF7)

(1) CEC local address setting register (CADR)

CADR is a 16-bit register that sets local addresses. It is valid only during a reception, the ADR00 to ADR14 bits correspond to CEC logical addresses 0 to 14, and up to 15 local addresses can be set. To specify address 15 as the CEC local address, clear the ADR00 to ADR14 bits to 0. A broadcast address always operates as a local address.

For example, when using addresses 0 as local addresses, 1 is set to the ADR00 bits.

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

Reset signal generation clears this register to 0000H.

Figure 17-2. Format of CEC Local Address Setting Register (CADR) (1/2)

Address: FF2AH, FF2BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CADR	0	ADR 14	ADR 13	ADR 12	ADR 11	ADR 10	ADR 09	ADR 08	ADR 07	ADR 06	ADR 05	ADR 04	ADR 03	ADR 02	ADR 01	ADR 00

ADR00	Address 0 (TV)
0	Does not set as local address.
1	Sets as local address.

ADR01	Address 1 (recording device 1)
0	Does not set as local address.
1	Sets as local address.

ADR02	Address 2 (recording device 2)
0	Does not set as local address.
1	Sets as local address.

ADR03	Address 3 (tuner 1)
0	Does not set as local address.
1	Sets as local address.

ADR04	Address 4 (playback device 1)
0	Does not set as local address.
1	Sets as local address.

Figure 17-2. Format of CEC Local Address Setting Register (CADR) (2/2)

Address: FF2AH, FF2BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CADR	0	ADR 14	ADR 13	ADR 12	ADR 11	ADR 10	ADR 09	ADR 08	ADR 07	ADR 06	ADR 05	ADR 04	ADR 03	ADR 02	ADR 01	ADR 00

ADR05	Address 5 (audio system)
0	Does not set as local address.
1	Sets as local address.

ADR06	Address 6 (tuner 2)
0	Does not set as local address.
1	Sets as local address.

ADR07	Address 7 (tuner 3)
0	Does not set as local address.
1	Sets as local address.

ADR08	Address 8 (playback device 2)
0	Does not set as local address.
1	Sets as local address.

ADR09	Address 9 (recording device 3)
0	Does not set as local address.
1	Sets as local address.

ADR10	Address 10 (tuner 4)
0	Does not set as local address.
1	Sets as local address.

ADR11	Address 11 (playback device 3)
0	Does not set as local address.
1	Sets as local address.

ADR12	Address 12 (reserved)
0	Does not set as local address.
1	Sets as local address.

ADR13	Address 13 (reserved)
0	Does not set as local address.
1	Sets as local address.

ADR14	Address 14 (specific use)
0	Does not set as local address.
1	Sets as local address.

Caution To specify address 15 (unregistered) as the CEC local address, clear the ADR00 to ADR14 bits to 0.

(2) CEC transmission buffer register (CTXD)

CTXD is an 8-bit register that sets transmit data. It sequentially transmits eight bits of data, starting from bit 7. A transmission/reception interrupt request signal (INTDA) is generated at the start timing of the header block and data block. Successive transmission can be performed by writing the next data to CTXD before the transmission ends after INTDA was generated.

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

Reset signal generation clears this register to 00H.

Figure 17-3. Format of CEC Transmission Buffer Register (CTXD)

Address: FF51H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTXD	CTXD							

Caution If an underrun error occurs (UERR = 1), transmission is not continued. An error interrupt is generated and the transmission wait state is entered.

(3) CEC reception buffer register (CRXD)

CRXD is an 8-bit register that retains receive data.

Receive data can be read by reading this register.

For every byte of data received, new data will be transferred from the CEC reception shift register.

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

Reset signal generation clears this register to 00H.

Figure 17-4. Format of CEC Reception Buffer Register (CRXD)

Address: FF52H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CRXD	CRXD							

Caution If an overrun error occurs (OERR = 1), the data of the reception buffer register will be overwritten.

(4) CEC control register 0 (CECCTL0)

CECCTL0 selects enabling operation, starting transmission, and the operating clock.

CECCTL0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-5. Format of CEC Control Register 0 (CECCTL0) (1/2)

Address: FF42H After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
CECCTL0	CECE	ACKTEN	CCL2	CCL1	CCL0	TXTRG	CECRXEN	EOM

CECE	CEC operation enable flag
0	Stops CEC operation. * Does not reset control register even if CECE = 0 is set.
1	Enables CEC transmit/receive operation.

ACKTEN	Enabling ACK bit timing error (bit width) check
0	Does not detect ACK bit timing errors (bit width).
1	Detects ACK bit timing errors (bit width). Notes 1

CCL2	CCL1	CCL0	Source clock (f_{CEC}) selection Notes 2, 3
0	0	0	$f_{PRS}/2^5$
0	0	1	$f_{PRS}/2^6$
0	1	0	$f_{PRS}/2^7$
0	1	1	$f_{PRS}/2^8$
0	0	0	f_{SUB} (32.768 kHz)
0	0	1	$f_{SUB} \times 2$ (32.768 kHz × 2)
Other than above			Setting prohibited

Notes 1. Timing errors are detected for the bit width (the values specified for DATBL and DATBH), in addition to the low-level width of the ACK bit (the values specified for LGC0LL, LGC0LH, LGC1LL, and LGC1LH). However, the maximum bit width (DATBH) is not checked for the ACK bit of the last frame (EOM = 1), even if ACKTEN is 1.

2. Rewritable only when CECE = 0.

3. Set the source clock frequency of the CEC counter to a frequency from 7.8125 to 78.125 kHz. Examples of CEC counter source clock settings are shown below.

CEC Operating Clock	CEC Counter Source Clock (f_{CEC})					
	When f_{PRS} = 20 MHz	When f_{PRS} = 16 MHz	When f_{PRS} = 12 MHz	When f_{PRS} = 10 MHz	When f_{PRS} = 8 MHz	When f_{PRS} = 2 MHz
$f_{PRS}/2^5$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	62.5 kHz
$f_{PRS}/2^6$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	31.25 kHz
$f_{PRS}/2^7$	Setting prohibited	Setting prohibited	Setting prohibited	78.125 kHz	62.5 kHz	15.625 kHz
$f_{PRS}/2^8$	78.125 kHz	62.5 kHz	46.875 kHz	39.0625 kHz	31.25 kHz	7.8125 kHz
f_{SUB}	32.768 kHz (when operating at f_{SUB} = 32.768 kHz)					
$f_{SUB} \times 2$	65.536 kHz (when operating at f_{SUB} = 32.768 kHz)					

Remarks 1. f_{CEC}: CEC counter source clock (selected by bits 5 to 3 (CCL2 to CCL0))

2. f_{PRS}: CPU/peripheral hardware clock oscillation frequency

3. f_{SUB}: Subsystem clock oscillation frequency

Figure 17-5. Format of CEC Control Register 0 (CECCTL0) (2/2)

Address: FF42H After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
CECCTL0	CECE	ACKTEN	CCL2	CCL1	CCL0	TXTRG	CECRXEN	EOM

TXTRG	Transmission start trigger bit[Notes 1, 2]
0	Does not start CEC transmission. (0 is always read. Writing 0 has no meaning.)
1	Starts CEC transmission when CECE = 1.

CECRXEN	Reception rejection control bit[Notes 3, 4, 5]		
1	Enables continuing reception or reports normal reception (normally selected).		
	Reception status		ACK/NACK timing output
	During direct address reception (to local station)	Normal reception	ACK
		Timing error occurrence	NACK
	During broadcast address reception	Normal reception	NACK
		Timing error occurrence	ACK
	During direct address reception (to another station)	Not participating in communication (high impedance)	
0	Stops continuing reception or reports abnormal reception.		
	Reception status		ACK/NACK timing output
	During direct address reception (to local station)	Normal reception	NACK
		Timing error occurrence	NACK
	During broadcast address reception	Normal reception	ACK
		Timing error occurrence	ACK
	During direct address reception (to another station)	Not participating in communication (high impedance)	

EOM	EOM setting bit
0	Continues transmission.
1	Last frame

Notes 1. TXTRG is a trigger bit from which 0 is always read. If the subsystem clock multiplication clock is selected (CCL3 to CCL0 = 1001B), the clock starts operating a maximum of two clocks later. A transmission starts when a maximum of two clocks elapse after 1 is set to this bit.

2. Writing to the TXTRG bit is prohibited until the CEC transmission ends. Transmission completion can be confirmed by checking that the shift register after the trigger bit is set is 0.
Set TXTRG to 1 when the bus is free (BUSST = 0). Transmission starts no more than two f_{CEC} clock cycles after TXTRG is set to 1.

3. Rewriting the CECRXEN bit during a communication (BUSST = 1) is prohibited.

4. If CECRXEN = 0 is set, reception rejection is reported at the next ACK/NACK timing and the communication standby state is entered.

5. Set CECRXEN = 1 after determining the local address (setting the CADR register). Set CECRXEN = 0, therefore, after checking that address match flag ADRF of the CEC communication status register (CECS) is 1.

(5) CEC control register 1 (CECCTL1)

CECCTL1 selects a digital filter, data interrupt generation, a start bit error interrupt, and whether to generate a communication complete interrupt and when to generate it.

CECCTL1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-6. Format of CEC Control Register 1 (CECCTL1) (1/2)

Address: FF47H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CECCTL1	CDFC	CINTMK	BLERRD	STERRD	CESEL1	CESEL0	SFT1	SFT0

CDFC	Digital filter select bit[Notes 1, 2]
0	Does not use a digital filter.
1	Uses a digital filter and eliminates noise of one cycle of f_{CEC}.

CINTMK	CEC data interrupt (INTDA) generation select register[Notes 1, 3]	
0	Destination address matches local address.	Generates data interrupt (INTDA).
	Destination address does not match local address.	Does not generate data interrupt (INTDA).
1	Destination address matches local address.	Generates data interrupt (INTDA).
	Destination address does not match local address.	

Notes 1. Rewritable only when CECE = 0.

2. Examples of the noise elimination width settings when the digital filter is used are shown below.

CEC Operating Clock (f_{CEC})	Noise Elimination Width by Digital Filter					
	When f_{PRS} = 20 MHz	When f_{PRS} = 16 MHz	When f_{PRS} = 12 MHz	When f_{PRS} = 10 MHz	When f_{PRS} = 8 MHz	When f_{PRS} = 2 MHz
$f_{CLK}/2^5$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	16 μs
$f_{CLK}/2^6$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	32 μs
$f_{CLK}/2^7$	Setting prohibited	Setting prohibited	Setting prohibited	12.8 μs	16 μs	64 μs
$f_{CLK}/2^8$	12.8 μs	16 μs	21.3 μs	25.6 μs	32 μs	128 μs
f_{SUB}	30.5 μs (when operating at f_{SUB} = 32.768 kHz)					
$f_{SUB} \times 2$	15.2 μs (when operating at f_{SUB} = 32.768 kHz)					

3. Whether to generate a data interrupt (INTDA) of the header block when the destination address and local address do not match during a reception can be selected by setting the CINTMK bit. See **17.7.5 CEC reception** for details.

Remarks 1. f_{CEC}: CEC counter source clock (selected by bits 5 to 3 (CCL2 to CCL0) of the CECCTL0 register)

2. f_{PRS}: CPU/peripheral hardware clock oscillation frequency

3. f_{SUB}: Subsystem clock oscillation frequency

Figure 17-6. Format of CEC Control Register 1 (CECCTL1) (2/2)

Address: FF47H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CECCTL1	CDFC	CINTMK	BLERRD	STERRD	CESEL1	CESEL0	SFT1	SFT0

BLERRD	Bus lock detection select bit[Notes 1, 2]
0	Does not detect sticking of receive data to high or low level.
1	Detects sticking of receive data to high or low level.

STERRD	Start bit error detection select bit[Notes 1, 3]
0	Does not detect timing errors during start bit reception.
1	Detects timing errors during start bit reception (recommended).

CESEL1	CESEL0	Communication complete interrupt (INTCE) generation timing[Note 1]
0	0	Detects EOM = 1 and generates communication complete interrupt once after ACK transmission/reception completion and another time after signal-free time set by SFT1 and SFT0 is detected.
0	1	Detects EOM = 1 and generates communication complete interrupt after ACK transmission/reception completion.
1	0	Generates communication complete interrupt after signal-free time set by SFT1 and SFT0 is detected.
1	1	Setting prohibited

SFT1	SFT0	Data bit width of signal-free time[Note 4]
0	0	Signal-free time of 3-data bit width
0	1	Signal-free time of 5-data bit width
1	0	Signal-free time of 7-data bit width
1	1	Does not detect signal-free time.

Notes 1. Rewritable only when CECE = 0.

2. The bus lock status of the CEC line can be detected by setting BLERRD to 1. If the next falling edge is not input for a period 2.5 times the 1-data bit width set with the NOMP register in a state in which the falling edge of the CEC line is awaited (excluding the communication standby state), an error interrupt (INTERR) is generated and a bus lock error flag (BLERR) is set. Afterward, the communication standby state is entered.

3. This sets whether to detect timing errors during start bit reception.
Timing errors of the start bit can be detected according to the set value of the STATLL, STATLH, STATBL, or STATBH register by setting STERRD = 1. If a timing error occurred, the start bit for which the error occurred is determined to be disabled and the communication standby state is entered. If STERRD is 0, no timing error is detected. All pulses are determined to be start bits.

4. Rewritable only when SFTST of the CECS register is 0.
The bit width for the signal-free time is specified using the NOMP register.

(6) CEC communication status register (CECS)

CECS indicates the CEC communication status.

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

Clearing CECE to 0 or generating a reset signal clears this register to 00H.

Figure 17-7. Format of CEC Communication Status Register (CECS) (1/2)

Address: FFA6H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CECS	SFTST	0	0	ITCEF	EOMF	TXST	BUSST	ADRF

SFTST	Signal-free time rewrite disable report flag
0	Enables rewriting SFT1 and SFT0. SFTST is cleared to 0 in the following cases. • When CECE = 0 • When the SFT1 and SFT0 rewrite disable period (three operating clocks f_{CEC}, maximum) has elapsed
1	Disables rewriting SFT1 and SFT0. SFTST is set to 1 in the following case. • When a write access to the SFT1 and SFT0 bits (CECCTL1 register) is performed

ITCEF	INTCE generation source flag
0	Generates INTCE if the signal-free time has been counted.
1	Generates INTCE if communication ends or an error is detected.
This setting is enabled only if CESEL1 and CESEL0 are cleared to 0. By checking ITCEF after INTCE is generated, it can be determined which source was the generation source. (For details, see **Figure 17-8**.)	

EOMF	EOM flag
0	The EOM bit received immediately before is logically 0.
1	The EOM bit received immediately before is logically 1.

TXST	Transmission status flag
0	During the communication standby state or reception (A follower is running.)
1	During transmission (The initiator is running.)

Figure 17-7. Format of CEC Communication Status Register (CECS) (2/2)

Address: FFA6H After reset: 00H R

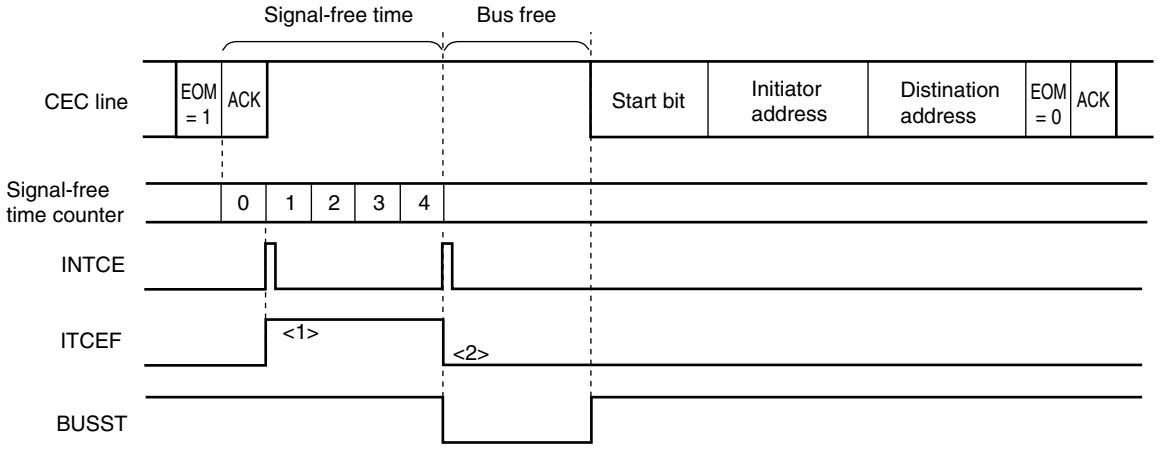
Symbol	7	6	5	4	3	2	1	0
CECS	SFTST	0	0	ITCEF	EOMF	TXST	BUSST	ADRF

BUSST	Bus busy detection flag
0	Bus-free state BUSST is reset to 0 in the following cases. • BUSST is reset to 0 regardless of the bus state when CECE = 0. • When the signal-free time set by the SFT1 and SFT0 bits has elapsed after the communication has ended (see **Figure 17-9**)
1	Bus busy state BUSST is set to 1 in the following cases. • When a fall of the CEC bus is detected (see **Figure 17-10**) • When CECE = 1 is set during a reception or standby period (see **Figure 17-11**)

ADRF	Address match detection flag
0	During a communication between other stations, while communication is stopped, or while the local station is transmitting ADRF is reset to 0 in the following cases. • When CECE = 0 • When reception is completed
1	During local reception ADRF is set to 1 in the following cases. • When the local address and reception destination address match • When a broadcast address is received

Figure 17-8. Checking INTCE Generation Source Using ITCEF When CESEL0 and CESEL1 Are 0

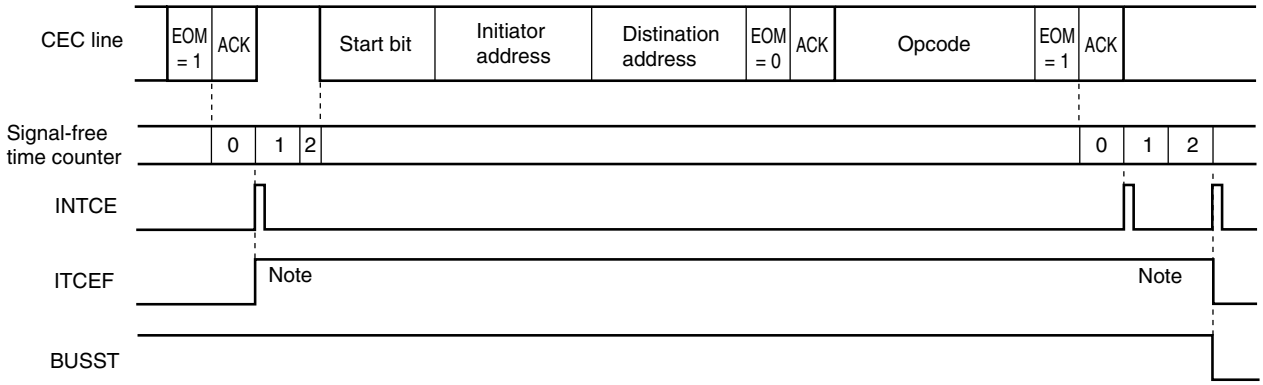
(1) Reception started after the signal-free time was counted



If ITCEF is 1, INTCE is generated after ACK is received (<1>).

If ITCEF is 0, INTCE is generated after the signal-free time is counted (<2>).

(2) Reception started during the signal-free time



Note ITCEF is 1 until INTCE is generated after the signal-free time is counted.

Figure 17-9. Timing When Signal-Free Time Set by SFT1 and SFT0 Bits Has Elapsed After Communication End

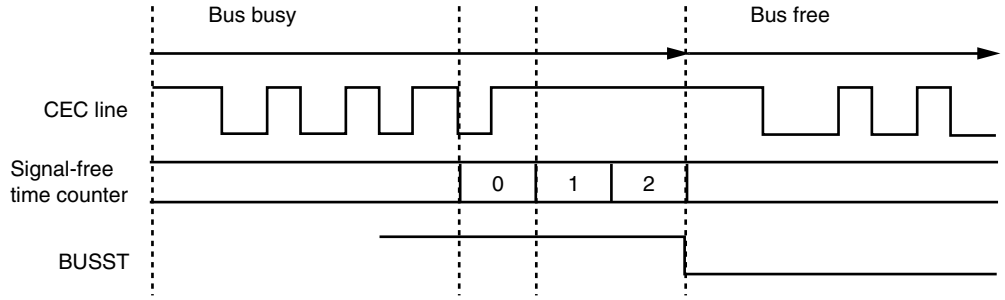


Figure 17-10. CEC Line Fall Detection Timing

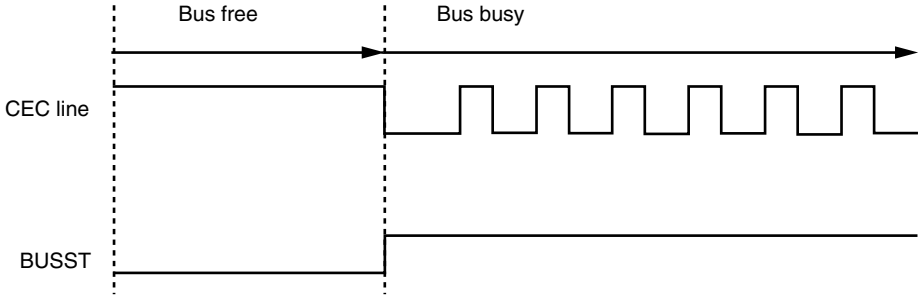
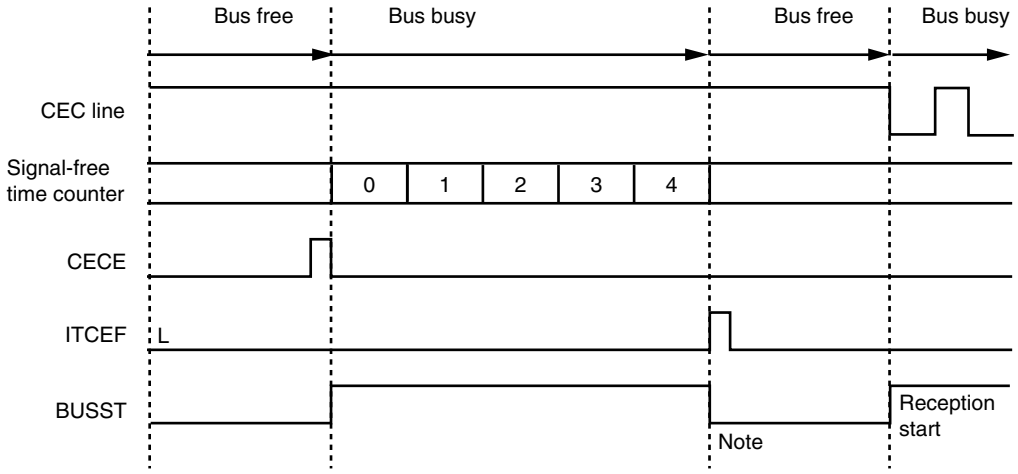


Figure 17-11. Timing When CECE = 1 Is Set During Reception or Standby Period



Note Transmission errors are not detected while transmitting the start bit and ACK bit.

(7) CEC communication error status register (CECES)

CECES indicates whether a bus lock error, arbitration loss, transmission error, timing error, ACK error, underrun error, or overrun error has detected.

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

Clearing CECE to 0 or generating a reset signal clears this register to 00H.

Figure 17-12. Format of CEC Communication Error Status Register (CECES) (1/2)

Address: FFA5H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CECES	0	BLERR	AERR	TXERR	TERR	ACKERR	UERR	OERR

BLERR	Bus lock error detection flag
0	No bus lock error has occurred. BLERR is cleared to 0 in the following case. • When BLCTRG = 1 is set
1	A bus lock error has occurred. BLERR is set to 1 in the following case. • When the CEC reception signal is fixed to low or high level midway through a frame An error occurs if the next falling edge is not input for a time 2.5 times the 1-data bit width set by the NOMP register after the falling edge of the CEC reception signal.

AERR	Arbitration loss detection flag
0	No arbitration loss has occurred or the communication is being stopped. AERR is reset to 0 in the following case. • When ACTRG = 1
1	Arbitration loss AERR is set to 1 in the following case. • When an arbitration loss occurs between start bit transmission and source address transmission

TXERR	Transmission error detection flag[Note]
0	No transmission error has occurred. TXERR is reset to 0 in the following case. • When TXCTRG = 1
1	A transmission error has occurred. TXERR is set to 1 in the following case. • When the logic of the transmit data and receive data are compared and do not match when the initiator is operating

Note Transmission errors are not detected while transmitting the start bit and ACK bit.

Figure 17-12. Format of CEC Communication Error Status Register (CECES) (2/2)

Address: FFA5H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CECES	0	BLERR	AERR	TXERR	TERR	ACKERR	UERR	OERR

TERR	Timing error detection flag
0	No timing error has occurred. TERR is reset to 0 in the following case. • When TCTRG = 1
1	A timing error has occurred. TERR is set to 1 in the following case. • When a violation is detected in the timing check of the received data

<R>

ACKERR	ACK error detection flag[Note]
0	No ACK error has occurred. ACKERR is reset to 0 in the following case. • When ACKTRG = 1
1	An ACK error has occurred. ACKERR is set to 1 in the following cases. • When a NACK (logical 1) is received during a direct address communication • When an ACK (logical 0) is received during a broadcast communication • When an NACK (logical 1) is received during a logical address allocation transmission

UERR	Underrun error detection flag
0	No underrun error has occurred. UERR is reset to 0 in the following case. • When UECTRG = 1
1	An underrun error has occurred. UERR is set to 1 in the following case. • When transmit data is not written to the transmission buffer register (CTXD) after a data interrupt (INTDA) is generated and before the next interrupt (INTDA) is generated

OERR	Overrun error detection flag
0	No overrun error has occurred. OERR is reset to 0 in the following case. • When OCTRG = 1
1	An overrun error has occurred. OERR is set to 1 in the following case. • When the next receive operation is completed before the receive data stored in the reception buffer register (CRXD) is read

<R> **Note** It can run at initiator but cannot run at follower.

(9) CEC communication error flag clear trigger register (CECFC)

CECFC clears error flags written to the communication error status register (CECES). Only the set bits can be cleared by setting "1" to each flag.

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

Reset signal generation clears this register to 00H.

Figure 17-13. Format of CEC Communication Error Flag Clear Trigger Register (CECFC)

Address: FFA7H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
CECFC	0	BLCTRG	ACTRG	TXCTRG	TCTRG	ACKCTRG	UCTRG	OCTRG

BLCTRG	Bus lock error clear trigger
0	Does not clear bus lock error flag.
1	Clears bus lock error flag.

ACTRG	Arbitration loss flag clear trigger
0	Does not clear arbitration loss flag.
1	Clears arbitration loss flag.

TXCTRG	Transmission error flag clear trigger
0	Does not clear transmission error flag.
1	Clears transmission error flag.

TCTRG	Timing error flag clear trigger
0	Does not clear timing error flag.
1	Clears timing error flag.

ACKCTRG	ACK error flag clear trigger
0	Does not clear ACK error flag.
1	Clears ACK error flag.

UCTRG	Underrun error flag clear trigger
0	Does not clear underrun error flag.
1	Clears underrun error flag.

OCTRG	Overrun error flag clear trigger
0	Does not clear overrun error flag.
1	Clears overrun error flag.

(10) Port function register 4 (PF4)

PF4 sets whether to use the P40/CECOUT/RTC1HZ pin of port 4 in I/O port mode or real-time counter correction clock output/CECOUT mode.

PF4 sets whether to use the P41/CECIN/RTCCL pin of port 11 in I/O port mode or real-time counter clock output/CECIN mode.

PF4 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-14. Format of Port Function Register 4 (PF4)

Address: FF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF4	0	0	0	0	0	0	PF41	PF40

PF40	P40 pin operating mode selection
0	I/O port mode or real-time counter correction clock output
1	CECOUT mode

PF41	P41 pin operating mode selection
0	I/O port mode or real-time counter clock output
1	CECIN mode

Caution Using the CECIO and CECIN/CECOUT pins at the same time is prohibited.
When PF72 = 1, do not set PF40 and PF41 to 1.
When PF40 = 1 or PF41 = 1, do not set PF7 to 1.

(11) Port function register 7 (PF7)

PF7 sets whether to use the P72/CECIO/KR2 pin of port 7 in I/O port mode or key interrupt input/CECIO mode.

A normal input buffer/CEC input buffer can be specified by setting the PF7 register and port input mode register 7 (PIM7).

Whether to use a diode connection can be set by setting the PF7 register and pull-up resistor option register 7 (PU7).

PF7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-15 Format of Port Function Register 7 (PF7)

Address: FF36H After reset: 00 R/W

Symbol	7	6	5	4	3	2	1	0
PF7	0	0	0	0	0	PF72	0	0

PF72	P72 pin operating mode selection
0	I/O port mode or key interrupt input
1	CECIO mode

Figure 17-16 Relationship Between PF7 Register and PIM7 and PU7 Registers

PF72	PIM72	Input buffer selection
0	0	Normal input buffer
0	1	
1	0	
1	1	CEC input buffer

PF72	PU72	Whether to use a diode connection
0	0	Does not connect pull-up resistor + diode.
0	1	
1	0	
1	1	Connects pull-up resistor + diode.

Cautions 1. Using the CECIO and CECIN/CECOUT pins at the same time is prohibited.
When PF72 = 1, do not set PF40 and PF41 to 1.
When PF40 = 1 or PF41 = 1, do not set PF72 to 1.
2. When PF72 = 1 is set, set PM72 = 0 and P72 = 1.

17.5 Start Bit and Data Bit Registers

17.5.1 Specifying low-level width and bit width of CEC transmission data

The low-level width and bit width of the transmit data are set by using the following registers.

(1) CEC transmission start bit low width setting register (STATL)

STATL is a 9-bit register that sets the low-level width of the start bit during a transmission.

STATL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-17. Format of CEC Transmission Start Bit Low Width Setting Register (STATL)

Address: FA06H, FA07H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATL	0	0	0	0	0	0	0	STATL8 to STATL0								

Remark Low-level width = (Set values of STATL8 to STATL0 + 1) × Clock cycle of f_{CEC}

(2) CEC transmission start bit width setting register (STATB)

STATB is a 9-bit register that sets the bit width of the start bit during a transmission.

STATB can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-18. Format of CEC Transmission Start Bit Width Setting Register (STATB)

Address: FA04H, FA05H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATB	0	0	0	0	0	0	0	STATB8 to STATB0								

Remark Bit width = (Set values of STATB8 to STATB0 + 1) × Clock cycle of f_{CEC}

(3) CEC transmission logical 0 low width setting register (LGC0L)

LGC0L is a 9-bit register that sets the low-level width of logical 0 during a transmission.

LGC0L can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-19. Format of CEC Transmission Logical 0 Low Width Setting Register (LGC0L)

Address: FA08H, FA09H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC0L	0	0	0	0	0	0	0	LGC0L8 to LGC0L0								

Remark Low-level width = (Set values of LGC0L8 to LGC0L0 + 1) × Clock cycle of f_{CEC}

(4) CEC transmission logical 1 low width setting register (LGC1L)

LGC1L is a 9-bit register that sets the low-level width of logical 1 during a transmission.

LGC1L can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-20. Format of CEC Transmission Logical 1 Low Width Setting Register (LGC1L)

Address: FA0AH, FA0BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC1L	0	0	0	0	0	0	0	LGC1L8 to LGC1L0								

Remark Low-level width = (Set values of LGC1L8 to LGC1L0 + 1) × Clock cycle of f_{CEC}

(5) CEC transmission data bit width setting register (DATB)

DATB is a 9-bit register that sets the bit width of the data bit during a transmission.

DATB can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-21. Format of CEC Transmission Data Bit Width Setting Register (DATB)

Address: FA0CH, FA0DH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATB	0	0	0	0	0	0	0	DATB8 to DATB0								

Remark 1-data bit width = (Set values of DATB8 to DATB0 + 1) × Clock cycle of f_{CEC}

17.5.2 Checking CEC reception data timing

The timing at which errors in the low-level width and bit width of the receive data are judged is set by the following registers.

(1) CEC reception start bit minimum low width setting register (STATLL)

STATLL is a 9-bit register that detects the minimum value of the low-level width of the start bit during a reception.

STATLL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-22. Format of CEC Reception Start Bit Minimum Low Width Setting Register (STATLL)

Address: FA10H, FA11H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATLL	0	0	0	0	0	0	0	STATLL8 to STATLL0								

Remark Low-level width = (Set values of STATLL8 to STATLL0 + 1) × Clock cycle of f_{CEC}

(2) CEC reception start bit maximum low width setting register (STATLH)

STATLH is a 9-bit register that detects the maximum value of the low-level width of the start bit during a reception.

STATLH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-23. Format of CEC Reception Start Bit Maximum Low Width Setting Register (STATLH)

Address: FA12H, FA13H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATLH	0	0	0	0	0	0	0	STATLH8 to STATLH0								

Remark Low-level width = (Set values of STATLH8 to STATLH0 + 1) × Clock cycle of f_{CEC}

(3) CEC reception start bit minimum bit width setting register (STATBL)

STATBL is a 9-bit register that sets the minimum value of the bit width of the start bit during a reception.

STATBL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-24. Format of CEC Reception Start Bit Minimum Bit Width Setting Register (STATBL)

Address: FA14H, FA15H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATBL	0	0	0	0	0	0	0	STATBL8 to STATBL0								

Remark Bit width = (Set values of STATBL8 to STATBL0 + 1) × Clock cycle of f_{CEC}

(4) CEC reception start bit maximum bit width setting register (STATBH)

STATBH sets the maximum value of the bit width of the start bit during a reception.

STATBH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-25. Format of CEC Reception Start Bit Maximum Bit Width Setting Register (STATBH)

Address: FA16H, FA17H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATBH	0	0	0	0	0	0	0	STATBH8 to STATBH0								

Remark Bit width = (Set values of STATBH8 to STATBH0 + 1) × Clock cycle of f_{CEC}

(5) CEC reception logical 0 minimum low width setting register (LGC0LL)

LGC0LL is a 9-bit register that sets the minimum value of the low-level width of logical 0 during a reception.

LGC0LL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-26. Format of CEC Reception Logical 0 Minimum Low Width Setting Register (LGC0LL)

Address: FA18H, FA19H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC0LL	0	0	0	0	0	0	0	LGC0LL8 to LGC0LL0								

Remark Low-level width = (Set values of LGC0LL8 to LGC0LL0 + 1) × Clock cycle of f_{CEC}

(6) CEC reception logical 0 maximum low width setting register (LGC0LH)

LGC0LH is a 9-bit register that sets the maximum value of the low-level width of logical 0 during a reception.

LGC0LH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-27. Format of CEC Reception Logical 0 Maximum Low Width Setting Register (LGC0LH)

Address: FA1AH, FA1BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC0LH	0	0	0	0	0	0	0	LGC0LH8 to LGC0LH0								

Remark Low-level width = (Set values of LGC0LH8 to LGC0LH0 + 1) × Clock cycle of f_{CEC}

(7) CEC reception logical 1 minimum low width setting register (LGC1LL)

LGC1LL is a 9-bit register that sets the minimum value of the low-level width of logical 1 during a reception.

LGC1LL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-28. Format of CEC Reception Logical 1 Minimum Low Width Setting Register (LGC1LL)

Address: FA1CH, FA1DH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC1LL	0	0	0	0	0	0	0	LGC1LL8 to LGC1LL0								

Remark Low-level width = (Set values of LGC1LL8 to LGC1LL0 + 1) × Clock cycle of f_{CEC}

(8) CEC reception logical 1 maximum low width setting register (LGC1LH)

LGC1LH is a 9-bit register that sets the maximum value of the low-level width of logical 1 during a reception.

LGC1LH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-29. Format of CEC Reception Logical 1 Maximum Low Width Setting Register (LGC1LH)

Address: FA1EH, FA1FH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC1LH	0	0	0	0	0	0	0	LGC1LH8 to LGC1LH0								

Remark Low-level width = (Set values of LGC1LH8 to LGC1LH0 + 1) × Clock cycle of f_{CEC}

(9) CEC reception data bit minimum bit width setting register (DATBL)

DATBL is a 9-bit register that sets the minimum value of the bit width of the data bit during a reception.

DATBL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-30. Format of CEC Reception Data Bit Minimum Bit Width Setting Register (DATBL)

Address: FA20H, FA21H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATBL	0	0	0	0	0	0	0	DATBL8 to DATBL0								

Remark Bit width = (Set values of DATBL8 to DATBL0 + 1) × Clock cycle of f_{CEC}

(10) CEC reception data bit maximum bit width setting register (DATBH)

DATBH is a 9-bit register that sets the maximum value of the bit width of the data bit during a reception.

DATBH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-31. Format of CEC Reception Data Bit Maximum Bit Width Setting Register (DATBH)

Address: FA22H, FA23H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATBH	0	0	0	0	0	0	0	DATBH8 to DATBH0								

Remark Bit width = (Set values of DATBH8 to DATBH0 + 1) × Clock cycle of f_{CEC}

17.5.3 Determining CEC reception data logic (1 or 0)

The timing at which receive data is judged to be 1 or 0 is set by the following register.

(1) CEC reception data sampling time setting register (NOMT)

NOMT is a 9-bit register that determines the sampling time of data during a reception.

NOMT can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 17-32. Format of CEC Reception Data Sampling Time Setting Register (NOMT)

Address: FA0EH, FA0FH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOMT	0	0	0	0	0	0	0	NOMT8 to NOMT0								

Remark Sampling time = (Set values of NOMT8 to NOMT0 + 1) × Clock cycle of f_{CEC}

Set this register within a period of LGC1LH < NOMT < LGC0LL.

17.5.4 Specifying bit width for detecting errors, signal-free time, and bus locking

The bit width used when detecting errors, the signal-free time, and bus locking is specified using the following register:

(1) CEC data bit reference width setting register (NOMP)

NOMP is a 9-bit register for specifying the bit width.

The bit width is used when counting the number of bits when detecting errors, the signal-free time, and bus locking.

NOMP can be set by a 16-bit memory manipulation instruction. It can be rewritten only if CECE is 0.

Reset signal generation clears this register to 0000H.

Figure 17-33. Format of CEC Data Bit Reference Wide Setting Register (NOMP)

Address: FA24H, FA25H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOMP	0	0	0	0	0	0	0	NOMT8 to NOMT0								

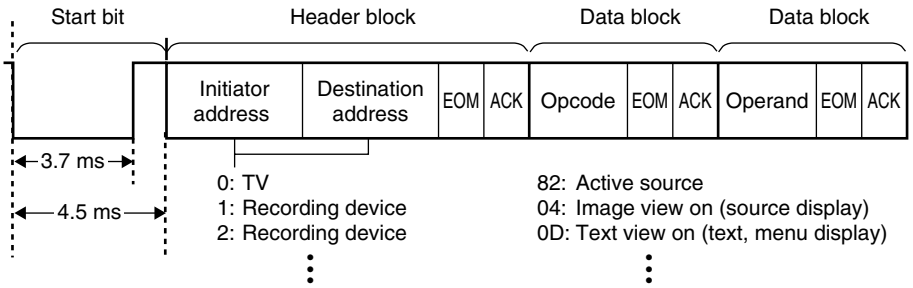
Remark Bit width = (Set values of NOMP8 to NOMP0 + 1) × Clock cycle of f_{CEC}

17.6 Operation of CEC Transmission/Reception Circuit

17.6.1 CEC transmission/reception data format

Figure 17-34 shows the basic CEC communication format. A CEC data frame consists of a start bit, a header block, data block 1 (opcode), and data block 2 (operand). The three blocks other than the start bit are configured of 10 bits.

Figure 17-34. Format Example



Start bit: Bit indicating the start of a message

Header block (1 block): Block indicating the source and destination addresses. Arbitration for the source address is performed at this block and the initiator with the smaller address obtains the transmission right.

Data block (blocks 0 to 15): Block consisting of an opcode and operand. The data length of the operand is determined by the opcode.

17.6.2 Communication types

CEC transmission/reception takes place in the state of a direct address message or broadcast address message. In a CEC communication, the transmitting side transmits a start bit and message (data) and the receiving side receives the message and returns a desired acknowledge signal to the transmitting side. CEC transmission/reception is configured of a start bit and a data bit and performs all transmissions/receptions of CEC.

Remark Direct address message: When the receiving party of a transmission is the local device
Broadcast address message: When the receiving party of a transmission is a device other than the local device

17.6.3 Bit timing

Figure 17-35 shows an example of the pulse format of a start bit. Whether the start bit is valid is judged at the low-level period (a) and bit period (b).

Figure 17-35. Start Bit Format

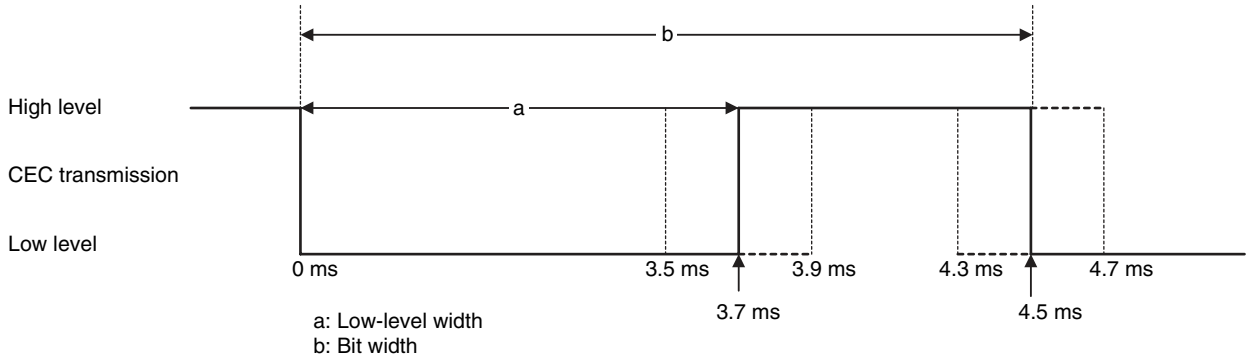
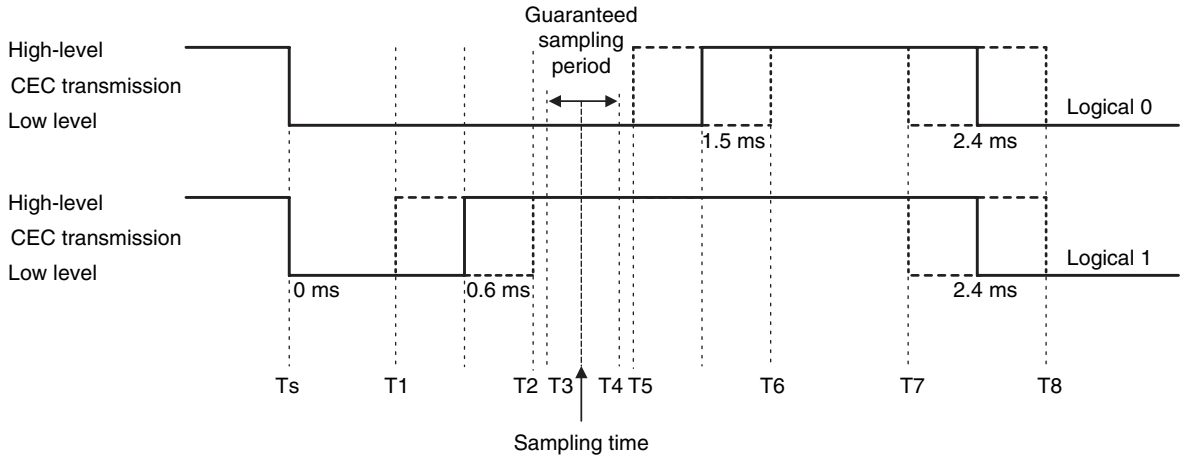


Figure 17-36 shows an example of the pulse format of a data bit timing. A data bit is sampled at a sampling timing set with the CEC reception data sampling time setting register (NOMT). If the result of sampling is low level, the pulse format is logical 0. If the result of sampling is high level, the pulse format is logical 1. The last change from high to low level of a data bit is the start of the next data bit. Consequently, the last data bit remains at high level.

Figure 17-36. Data Bit Format



Ts	0 ms	The bit start event
T1	0.4 ms	The earliest time for a low-high transition when indicating a logical 1
T2	0.8 ms	The latest time for a low-high transition when indicating a logical 1
T3	0.85 ms	The earliest time it is safe to sample the signal line to determine its state
T4	1.25 ms	The latest time it is safe to sample the signal line to determine its state
T5	1.3 ms	The earliest time a device is permitted return to a high impedance state(logical 0)
T6	1.7 ms	The latest time a device is permitted return to a high impedance state(logical 0)
T7	2.05 ms	The earliest time for the start of a following bit
	2.4 ms	The nominal data bit period
T8	2.75 ms	The latest time for the start of a following bit

17.6.4 Header/data block

All data blocks are configured of 10 bits and have the same structure. Figure 17-37 shows the configuration of header and data blocks. An information bit has a different meaning for a header block and data block, and indicates the data, opcode, and address. EOM (End of Message) and ACK (Acknowledge) are control bits and have the same meanings for a header block and data block.

Figure 17-37. Header Block and Data Block Format

Header/data block									
7	6	5	4	3	2	1	0	–	–
Information bit								EOM	ACK

A header block consists of an initiator logical address, destination logical address, EOM (End of Message), and ACK (Acknowledge). The EOM of a header block is used for “ping” with another device (checking whether the power of another device is turned on). “ping” can be checked by setting EOM = 1 and transmitting only the header block (transmitting a message without data blocks). In the case of direct address transmission, the power of the device to which the header block is transmitted is turned on if an ACK is returned.

17.6.5 EOM (End of Message)

An EOM indicates whether the transmitted block is the last block of a message. It is added to an information bit and output.

Caution EOM bit = 0: When one block or multiple blocks follow
EOM bit = 1: When the transmitted block is the last block

17.6.6 ACK (Acknowledge)

The meaning of an ACK depends on whether the receiving party of a transmission is a direct address message or broadcast message. The result of comparing the received data and CEC line data is transmitted to the transmitting side as an ACK or NACK.

Direct communication: An ACK is transmitted if the comparison result is normal.
Broadcast communication: An NACK is transmitted if the comparison result is normal.

Cautions 1. ACK = 0 is the normal value for a direct address message.

<1> If no error exists in the header block and the local address is used, the ACK bit is 0.
<2> If no error exists in the data blocks, the ACK bit is 0.
<3> If an error exists in the header block or another address is used, the ACK bit is 1.
<4> If an error exists in the data blocks, the ACK bit is 1.

An NACK (ACK = 1) is the normal value for a broadcast message.
<1> If one or more followers have abandoned the message, the ACK bit is 0.
<2> If all followers have not abandoned the message, the ACK bit is 1.

2. The initiator always outputs logical 1 at the ACK bit timing. Consequently, a follower determines the logic level of the ACK bit.

17.7 CEC Communication Functions

17.7.1 Communication bit width adjustment function

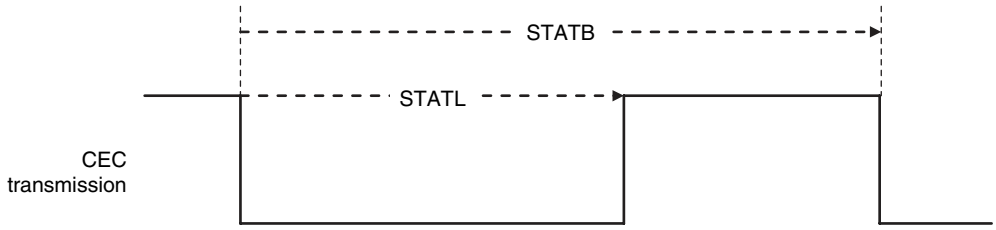
This function can be used to set the low-level width and bit width of the start bit and data bit during a transmission.

The relationships between the various width setting registers (see **17.5.1**) and the bit timing are shown below.

<Start bit>

The STATL register is used to set the low-level width and the STATB register is used to set the bit width of the start bit.

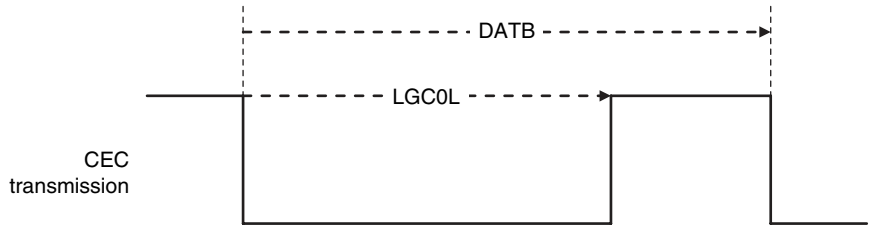
Figure 17-38. Start Bit Output Waveform



<Data bit (logical 0)>

The LGC0L register is used to set the low-level width and the DATB register is used to set the bit width of the data bit of logical 0.

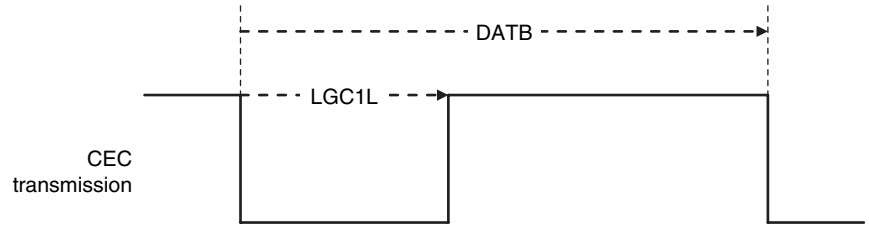
Figure 17-39. Data Bit (Logical 0) Output Waveform



<Data bit (logical 1)>

The LGC1L register is used to set the low-level width and the DATB register is used to set the bit width of the data bit of logical 1.

Figure 17-40. Data Bit (Logical 1) Output Waveform



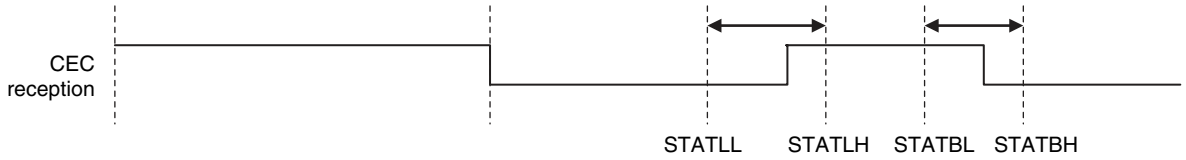
17.7.2 Receive bit timing check function

The CEC transmission/reception circuit has a timing check function that judges whether the low-level width and bit width of the start bit and data bit during a reception are within the set range. The timing check time can be set by using the various timing judgment registers (see **17.5.2**). The relationships between the timing judgment registers and bit timing are shown below.

<Start bit>

The STATLL register is used to set the minimum low-level width and the STATLH register is used to set the maximum low-level width of the start bit. The STATBL register is used to set the minimum value and the STATBH register is used to set the maximum value of the start bit width.

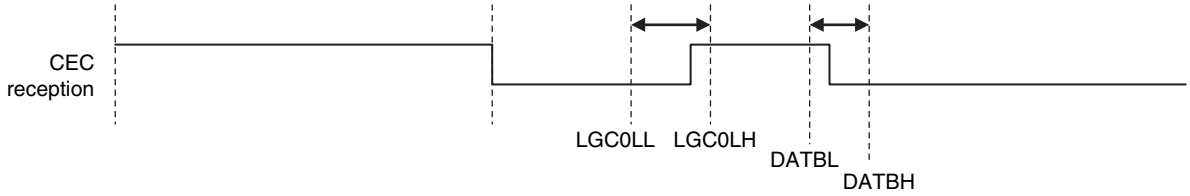
Figure 17-41. Start Bit Reception Timing



<Data bit (logical 0)>

The LGC0LL register is used to set the minimum low-level width and the LGC0LH register is used to set the maximum low-level width of the data bit (logical 0). The DATBL register is used to set the minimum value and the DATBH register is used to set the maximum value of the bit width.

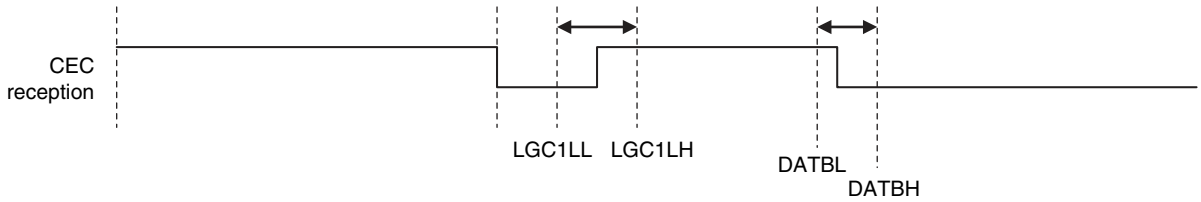
Figure 17-42. Data Bit (Logical 0) Reception Timing



<Data bit (logical 1)>

The LGC1LL register is used to set the minimum low-level width and the LGC1LH register is used to set the maximum low-level width of the data bit (logical 1). The DATBL register is used to set the minimum value and the DATBH register is used to set the maximum value of the bit width.

Figure 17-43. Data Bit (Logical 1) Reception Timing

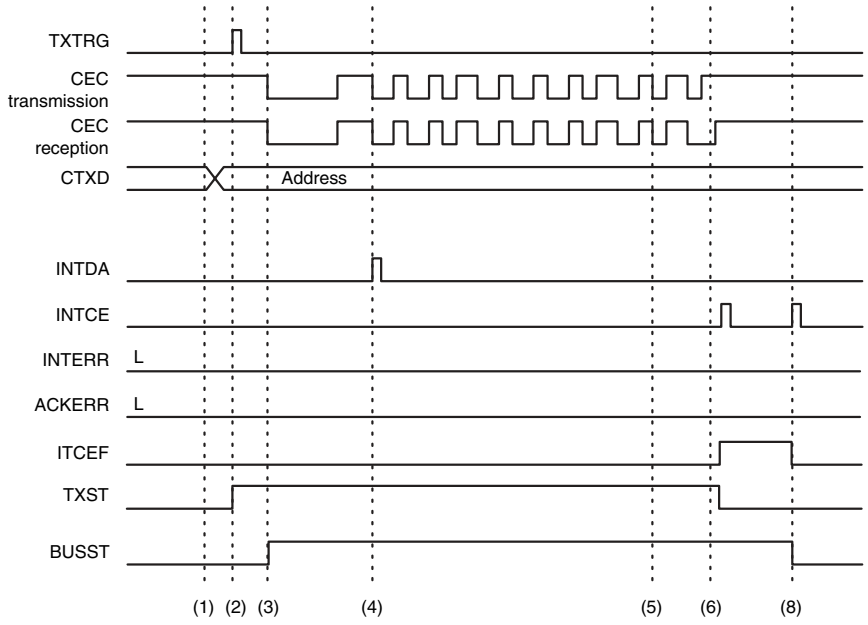


17.7.3 Initial CEC communication settings

The initial CEC communication setting flow is explained below. The logical address acquisition flow is executed by setting the various control registers and using direct address transmission after a reset. In a logical address acquisition transmission, EOM = 1 is set because the same address is set for the source and destination addresses and only the header block is transmitted. Furthermore, to prevent a false address match from occurring before the local address is determined, CECRXEN = 0 must be set until the CADR setting. Figure 17-44 shows the logical allocation timing diagram and Figure 17-45 the operation procedure and an explanation of the operation.

Figure 17-44. Logical Allocation (CECSEL0, CECSEL1 = 0, 0)

- The address is used by another device



- The address is not used by another device

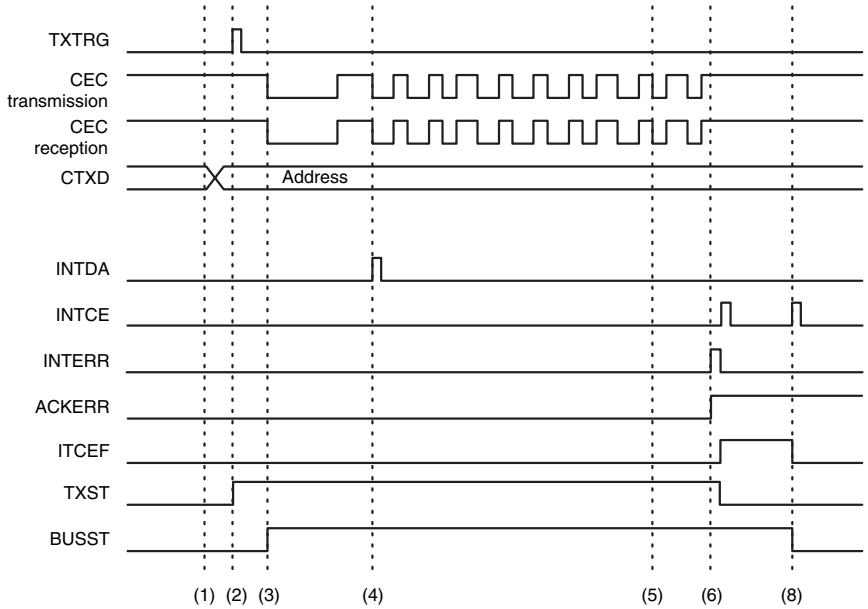


Figure 17-45. Initial CEC Communication Setting Procedure (1/2)

	Software Manipulation	Hardware State
Initial CEC setting	[Operating clock (f_{CEC}) setting] Set CCL2 to CCL0. [Reception rejection control setting] Set CECRXEN to 0. [Setting for reporting address mismatch] Set CINTMK. [Noise elimination selection] Set CDFC. (Specify whether to use the noise filter.) [Start bit low-level/bit width setting] Set the low/bit width to STATL/STATB. [Logical 0/1 low-level/bit width setting] Set the low/bit width to LGC0L/LGC1L/DATB. [Sampling time setting] Specify the time to sample the received data for NOMT. [Bit width setting] Specify the bit width for NOMP. [Timing check register settings] Set the low-level width timing check time of the start bit to STATLL/STATLH. Set the bit width timing check time of the start bit to STATBL/STATBH. Set the low-level width timing check time of the data bit (logical 0) to LGC0LL/LGC0LH. Set the low-level width timing check time of the data bit (logical 1) to LGC1LL/LGC1LH. Set the bit width timing check time of the data bit to DATBL/DATBH. [Bus lock detection setting] Set up BLERRD. (Select whether to detect bus locking.) [Start bit timing error detection setting] Set STERRD. (Select whether to detect timing errors of the start bit.) [Communication complete interrupt setting] Set CESEL1 and CESEL0. [Signal-free time setting] Set SFT1 and SFT0. (Set the signal-free time detection time.) [Operating clock (f_{CEC}) supply] Set CECE to 1. →	The operating clock is stopped. The operating clock (f_{CEC}) is supplied. Transmission can be performed. The signal-free time is started. BUSST becomes 1. BUSST becomes 0 and the communication standby state is entered after counting up to the set values of SFT1 and SFT0.

Figure 17-45. Initial CEC Communication Setting Procedure (2/2)

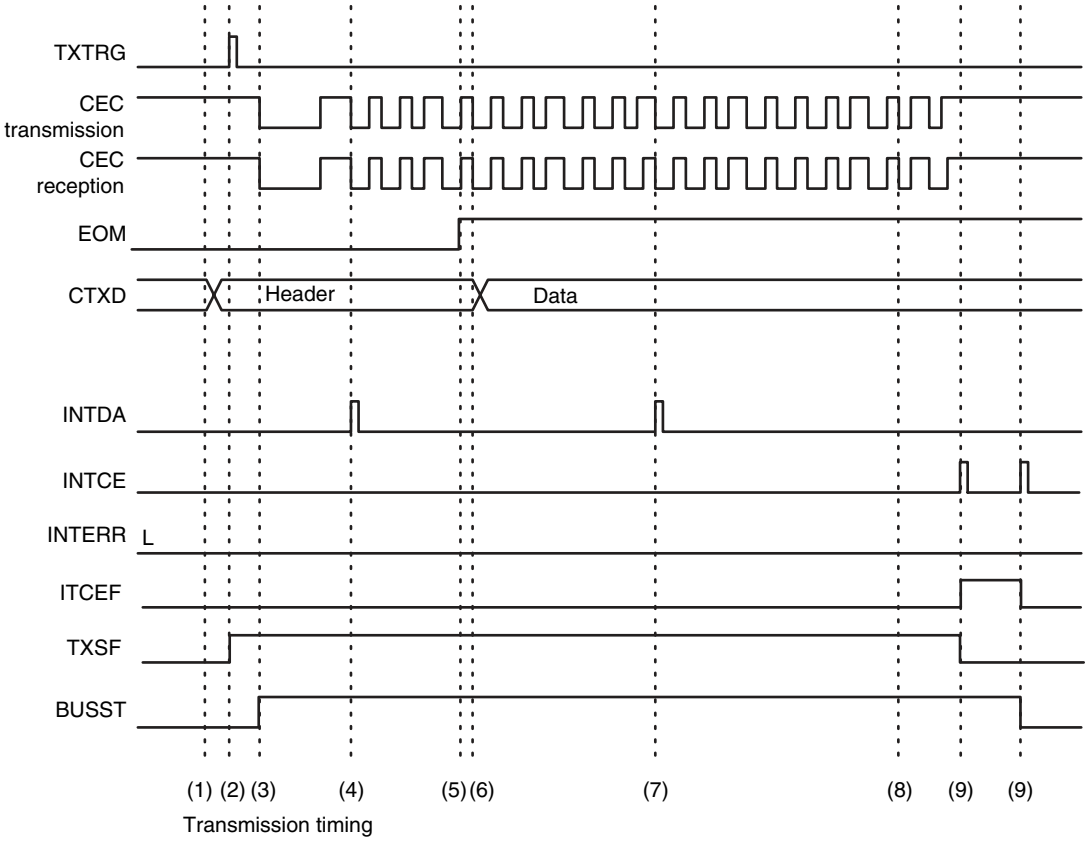
	Software Manipulation	Hardware State
Logical address allocation	[EOM setting] Set EOM to 1. [Transmit data setting] (1) Set the transmit data (logical address) to CTXD. [Bus-free state check] Check that BUSST is 0. [Starting transmit operation] (2) Set TXTRG to 1. →	Transmission is started. The start bit is output (3).
	Do not write the next data, because only the header block is transmitted. ←	The values set to the CTXD register are sequentially output at the same time as INTDA is output when the header block output is started (4).
		1 is output from the EOM bit (5).
	[Local address setting]	
		The ACK bit is received.
	• ACK Change the transmitted address and then retransmit the address, because it is used by another station. ←	When logical 0 is received, INTERR is not output and the ACKERR flag is not set (6).
	• NACK Use the transmitted address as the local address, because the transmitted address is not used by another station (CADR setting). ←	When logical 1 is received, INTERR is output and the ACKERR flag is set (7).
		INTCE is output according to the CESEL1, CESEL0, SFT1, and SFT0 bit settings (8).
	[Reception rejection control setting] Set CECRXEN to 1. →	The communication standby state is entered.

17.7.4 CEC transmission

CEC transmission is explained below. This section assumes that the settings in **17.7.3 Initial CEC communication settings** are completed.

A CEC transmission performs a receive operation even during transmission and performs an arbitration check, a data check, and a timing check. The value of the receive data register (CRXD) during a transmit operation, however, is not guaranteed.

Figure 17-46. Basic Transmission Timing (Direct Address Transmission) (CESEL0, CESEL1 = 0, 0)



Caution A new start bit detected during transmission (the time from when a start bit is detected until the EOM of the last frame is received) is ignored and detected as a timing error. Communication is not restarted.

(1) CEC transmission manipulation procedure

Figure 17-47. CEC Transmission Manipulation Procedure

	Software Manipulation	Hardware State
CEC transmit operation	[Signal-free time setting] Set SFT1 and SFT0. (Set the signal-free time detection time.) [EOM setting] (1) Set EOM (EOM = 0). [Transmit data setting] (1) Set the transmit data to CTXD. [Bus-free state check] Check that BUSST is 0. [Starting transmit operation] (2) Set TXTRG to 1. →	Transmission is started. The start bit is output (3).
	[EOM setting] (5) Set the EOM of the next frame (EOM = 1) before the next frame starts (7).	← The values set to the CTXD register are sequentially output at the same time as INTDA is output when the header block output is started (4).
	[Transmit data setting] (6) Set the transmit data to CTXD.	Outputting the data of the second frame is started (7). 1 is output at the EOM bit position because the last frame is reached (8). INTCE is output according to the CESEL1, CESEL0, SFT1, and SFT0 bit settings (9). The communication standby state is entered.

(2) Broadcast transmission

When FH is set to the destination address of the header block transmit data (CTXD), the hardware recognizes the current transmission as a broadcast transmission and operates. Normally, the communication is judged as being successfully performed when logical 0 is received at the ACK bit timing; however, in broadcast communication, the communication is judged as being successfully performed when logical 1 is received at the ACK bit timing.

The hardware judges whether the communication is a direct communication or broadcast communication by looking at the transmit data of the header block, and it automatically determines whether the reception of logical 0 or logical 1 has been successfully performed.

(3) CEC transmission interrupt

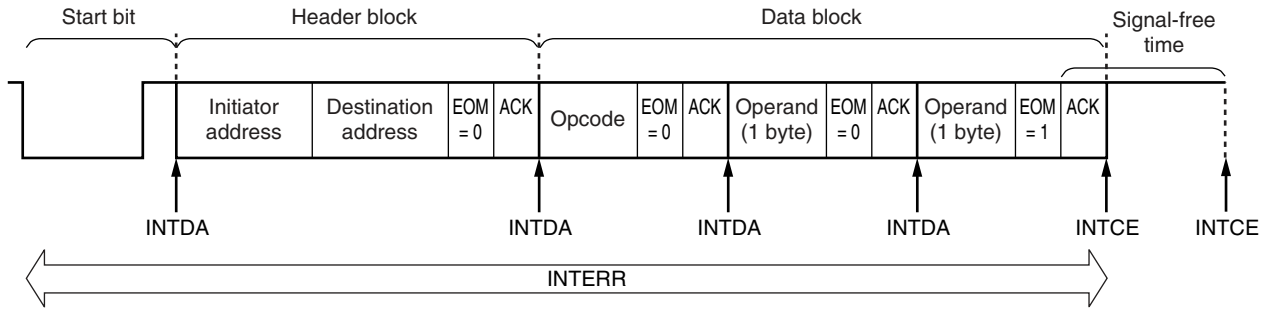
The hardware has three interrupt functions, namely a data interrupt (INTDA), a communication complete interrupt (INTCE), and an error interrupt (INTERR).

A data interrupt (INTDA) occurs at the start of each block.

A communication complete interrupt (INTCE) can be generated if ACK reception for a data block for which EOM is set to 1 ends, if the signal-free time specified using SFT1 and SFT0 elapses, or if both conditions occur, depending on the settings of CESEL1 and CESEL0.

An error interrupt (INTERR) is generated if a timing error, ACK error, underrun error, transmission error, or bus lock error is detected during any period of time during communication.

Figure 17-48. Interrupt Generation Timing



Caution If the falling edge of the CEC line is detected when receiving the ACK bit by setting EOM to 1 (before receiving the ACK bit ends), an irregular operation is performed as shown in Table 17-2 according to that timing.

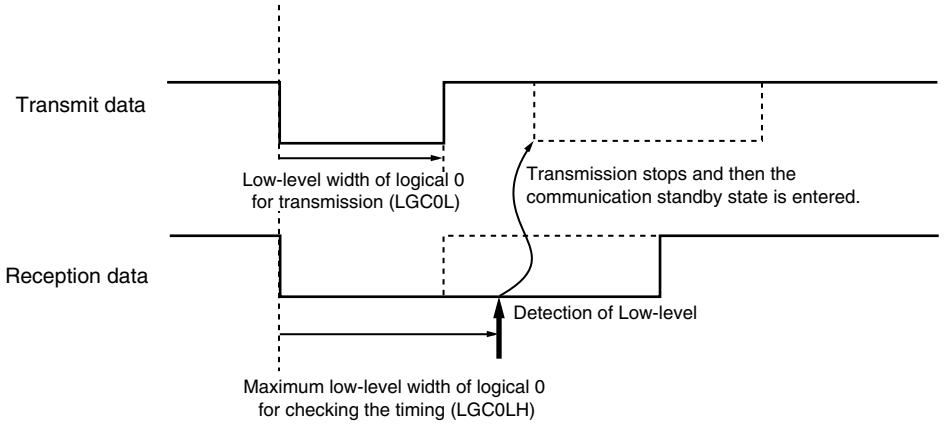
Table 17-2. Operation If Falling Edge of CEC Line Is Detected before Receiving ACK Bit Ends

CEC Line Falling Timing	Values Specified for CESEL1 and CESEL0 Bits of CECCTL1 Register	INTCE Generation	Handling of ACK Bit	Operation after CEC Line Falls
After the minimum data bit value (DATBL ≤ counter)	CESEL1 and CESEL0 are 0 and 0, or 0 and 1, respectively.	INTCE is generated once when the CEC line falls.	Handling the ACK bit is enabled because it has the correct width. (ACK or NACK is correctly determined.)	The start of the next communication is recognized and then determining whether to receive the start bit starts.
	CESEL1 and CESEL0 are 1 and 0, respectively.	INTCE is not generated.		
Before the minimum data bit value (counter < DATBL)	CESEL1 and CESEL0 are 0 and 0, or 0 and 1, respectively.	INTCE is generated once when the CEC line falls.	ACK cannot be correctly determined because it has the incorrect width. (If ACKTEN is set to 1, a timing error occurs.)	
	CESEL1 and CESEL0 are 1 and 0, respectively.	INTCE is not generated.		

(4) Receiving error handling pulse

If the received data is at low level when the maximum low-level width of logical 0 is reached while the initiator runs, an error handling pulse is judged to be received, a timing error occurs, transmission stops, and then the communication standby state is entered.

Figure 17-49. Interrupt Generation Timing



17.7.5 CEC reception

CEC reception is explained below. This section assumes that the settings in **17.7.3 Initial CEC communication settings** are completed. During reception, the data is received at the sampling timing set by the CEC reception data sampling time setting register (NOMT) and stored in CRXD.

The receive operation differs depending on the CECRXEN bit setting value, CINTMK bit setting value, communication type (direct address communication or broadcast communication), and whether the reception address and local address match.

The correspondences between various conditions and the operations are shown in the following table.

Table 17-3. Corresponding Operation During CEC Reception

CEC RXEN	Communication Type	Address Match/ Mismatch	CINTMK Bit	BUSST Operation	INTDA Output	INTCE Output	INTERR Output	Error Flag Operation	Error Detection (Other than Short Bit Width Detection)	Error Detection (Short Bit Width Detection)	Error Handling Output	Bus Lock Detection Note 1	ACK/NACK Output	Signal-Free Time Count
0	–	–	–	√	×	×	×	×	×	×	×	√Note 2	×	×
1	Start bit	–	–	√	×	√Note 3	×	×	√Note 4	√Note 4	×	√	×	√
	Header	Mismatch	0	√	×	√Note 3	√	√	√	√	√	√	√	√
			1	√	√	√	√	√	√	√	√	√	√	√
		Match	–	√	√	√	√	√	√	√	√	√	√	√
	Direct (data)	Mismatch	0	√	×	×	×	×	×	√	√	√	×	√
			1	√	√	√	√	√	√	√	√	√	×	√
		Match	–	√	√	√	√	√	√	√	√	√	√	√
	Broadcast (data)	–	–	√	√	√	√	√	√	√	√	√	√	√

Notes 1. Bus lock errors are detected by setting BLERRD.

2. A bus lock error is detected but a flag is not set.

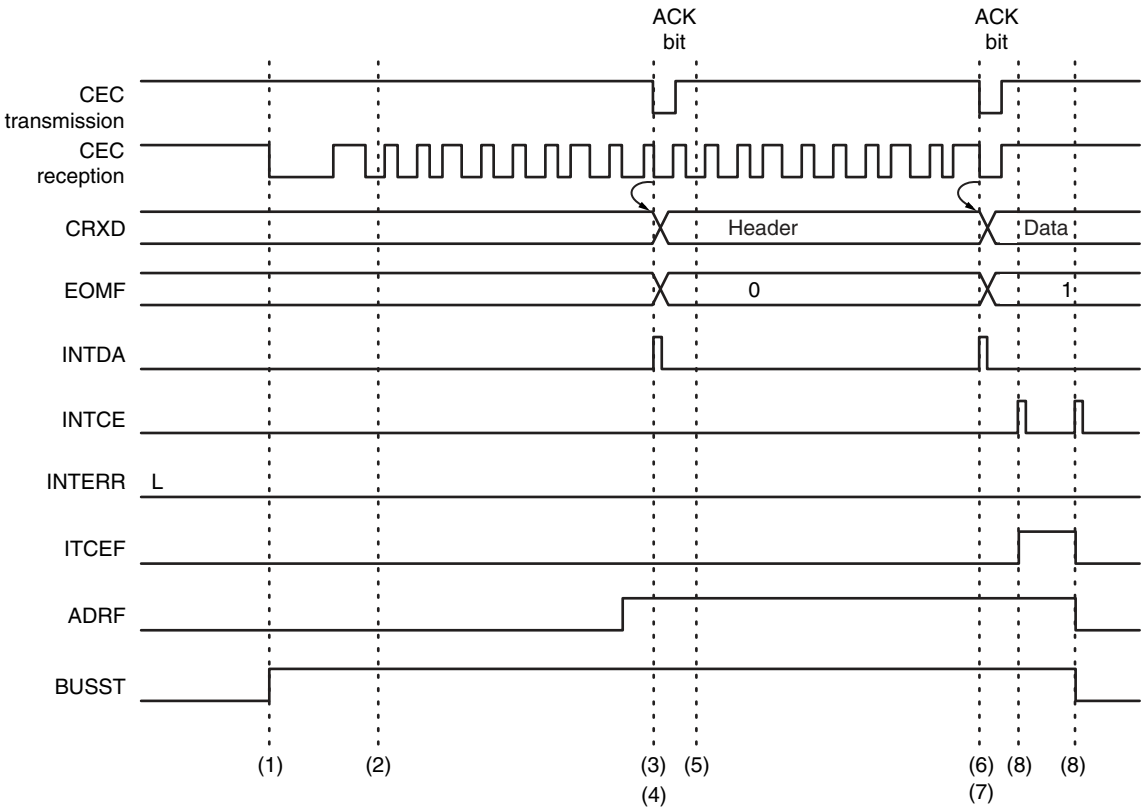
3. Only, INTCE output is generated if an error is detected.

4. This is supported only if detecting timing errors for the start bit (STERRD = 1). An error is detected but a flag is not set.

Remark √: Supported, ×: Not supported, –: Don't care

(1) CEC reception manipulation procedure

Figure 17-50. Basic Reception Timing (1)
(Direct Address Reception, CESEL0, CESEL1 = 0, 0)



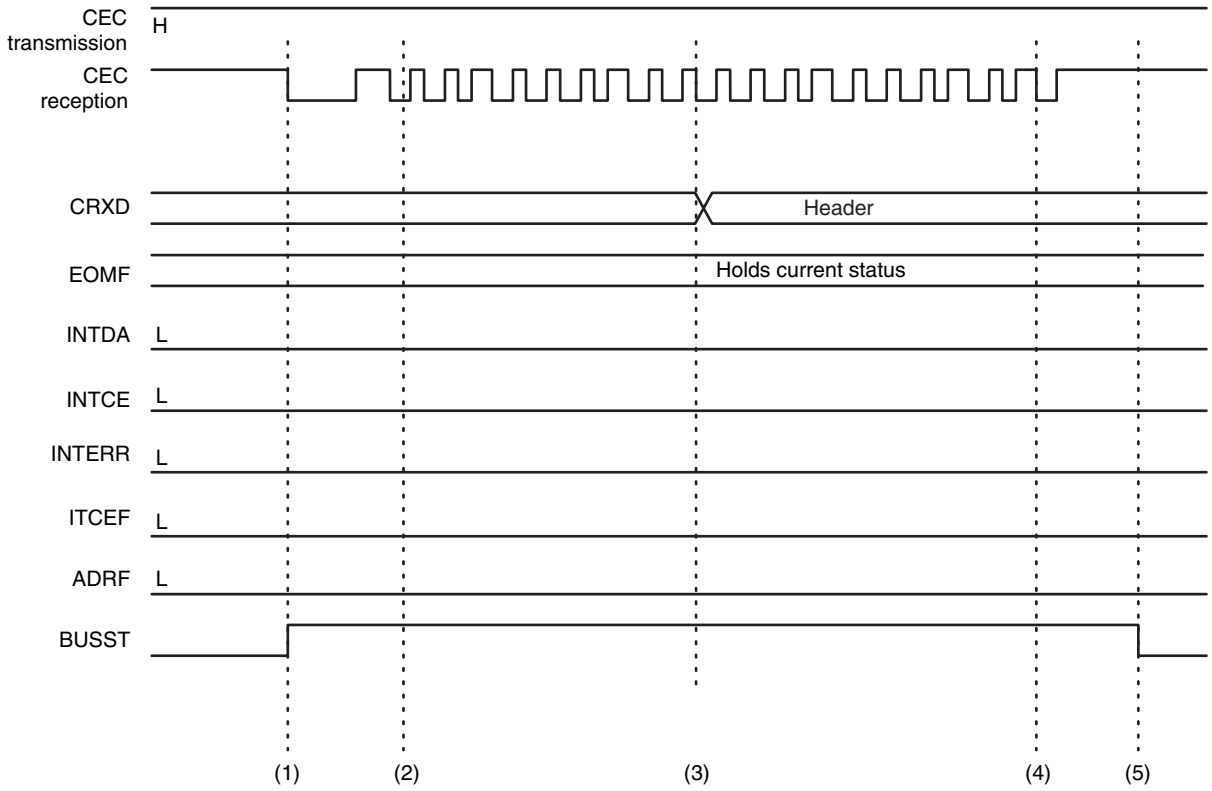
Caution A new start bit detected during transmission (the time from when a start bit is detected until the EOM of the last frame is received) is ignored and detected as a timing error.

Remark (1) to (8) in Figure 17-50 correspond to (1) to (8) in Figure 17-51.

Figure 17-51. CEC Reception Manipulation Procedure (1)

	Software Manipulation	Hardware State
CEC receive operation		[Start bit detection] The falling edge of the CEC reception signal is detected and a receive operation is started (1). [Sampling] The data is sampled at the NOMT setting time and sequentially stored in the shift register (2). [Address match interrupt] INTDA is generated because the address received at the header block has matched with the local address (3).
	Prepare for receiving a standby release or the like in response to INTDA generation. Use EOMF to check whether to continue transmission or whether it is the last frame.	[ACK bit transmission] Logical 0 is transmitted at the ACK bit timing because the reception was successful (4). [Continuing reception] The data of the second frame is continuously received (5). [Receive data interrupt] When receiving 8-bit data is completed, the data is transferred to CRXD and INTDA is generated (6).
	Read the receive data from CRXD in response to INTDA generation. Use EOMF to check whether to continue transmission or whether it is the last frame.	[ACK bit transmission] Logical 0 is transmitted at the ACK bit timing because the reception was successful (7). [Reception completion] The reception is judged to be completed because EOM = 1 is received and INTCE is output according to the CESEL1, CESEL0, SFT1, and SFT0 bit settings (8).

Figure 17-52. Basic Reception Timing (2)
(CECRXEN = 1, Direct Address, Address Mismatch, CINTMK = 0)



Remark (1) to (5) in Figure 17-52 correspond to (1) to (5) in Figure 17-53.

Figure 17-53. CEC Reception Manipulation Procedure (2)

	Software Manipulation	Hardware State
CEC receive operation		[Start bit detection] The falling edge of the CEC reception signal is detected and a receive operation is started. Set BUSST flag (1). [Sampling] The data is sampled at the NOMT setting time and sequentially stored in the shift register (2). [Address match interrupt] INTDA is not generated and neither ACK nor NACK is returned because the address received at the header block does not match the local address (3). However, monitoring CECRXD continues in order to check the bit length and detect communication completion. [ACK bit transmission] Neither ACK nor NACK is returned because communication is performed between other stations (4). [Reception completion] Communication between other stations is judged to be completed because EOM = 1 is received, the signal-free time is counted according to the SFT1 and SFT0 bit settings, and then BUSST is cleared to 0 (5).

(2) Broadcast reception

The reception flow and timing check period are the same as those of direct address reception. If the destination address transmitted by the initiator is FH, the communication operates as a broadcast reception.

The differences from direct address reception are as follows.

- Logical 1 is transmitted at the ACK bit timing in a normal operation.
- If reception has failed or CECRXEN = 0 has been set, logical 0 is transmitted at the ACK bit timing.

(3) CEC reception interrupt

Three interrupt functions, namely a data interrupt (INTDA), communication complete interrupt (INTCE), and error interrupt (INTERR) are provided.

A data interrupt (INTDA) is output at the following timings during reception (follower).

- When the address received at the header block of a direct address communication has matched the local address
- When address reception has been completed at the header block of a direct address communication when CINTMK = 1 is set
- When address reception of a broadcast communication has been completed at the header block
- When data reception has been completed at the data block and the receive data has been stored in the CRXD register

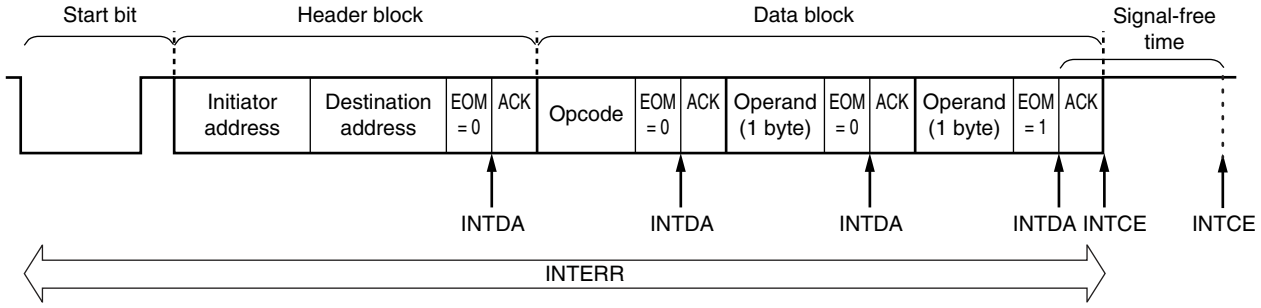
Communication complete interrupt INTCE is output at the following timings during reception (follower).

- CESEL1, CESEL0 = 0, 0
  INTCE is output if receiving the ACK bit of the last frame (EOM = 1) ends, if the signal-free time has been counted, or if the falling edge of the CEC line is detected starting in the high-level period of the ACK bit of the last frame while the signal-free time is counted.
- CESEL1, CESEL0 = 0, 1
  INTCE is output if receiving the ACK bit of the last frame (EOM = 1) ends or if the falling edge of the CEC line is detected starting in the high-level period of the ACK bit of the last frame while the signal-free time is counted.
- CESEL1, CESEL0 = 1, 0
  INTCE is output if the signal-free time has been counted.

An error interrupt (INTERR) is output at the following timings during reception (follower).

- When a timing error is detected
- When an overrun error is detected
- When a bus lock error is detected when BLERRD = 1 is set

Figure 17-54. Basic Reception Interrupt Timing



17.7.6 Error detection function

The hardware detects the following seven errors.

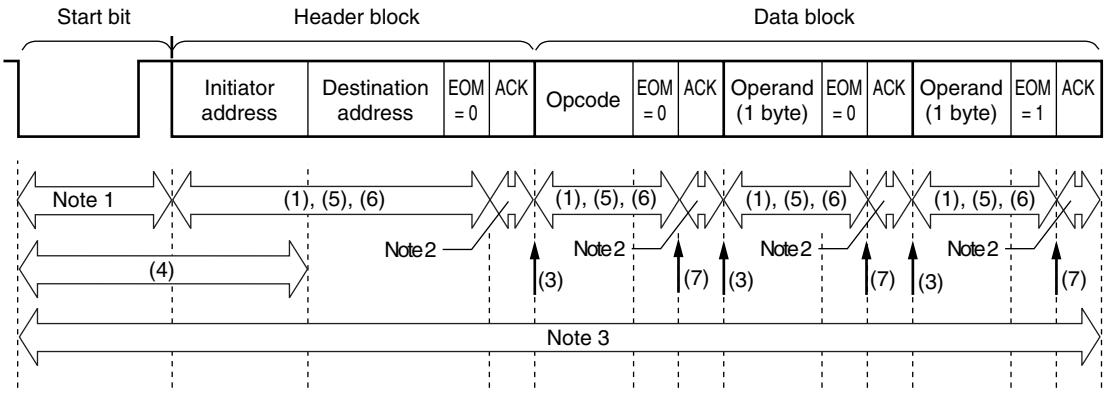
The errors that can be detected for the initiator and follower are shown below.

Table 17-4. Errors That Can Be Detected for Initiator and Follower

Error	Initiator	Follower
(1) Transmission error	√	×
(2) ACK error	√	×
(3) Underrun error	√	×
(4) Arbitration error	√	×
(5) Timing error (low-level width)	√	√
(6) Timing error (bit width)	√	√
(7) Overrun error	×	√
(8) Bus lock error	√	√

Remark √: Detected, ×: Not detected

Figure 17-55. Error Detection Period



Notes 1. (5) and (6) if detecting timing errors for the start bit (STERRD = 1).

2. (6) if detecting timing errors for the ACK bit (ACKTEN = 1), in addition to (2) and (5).

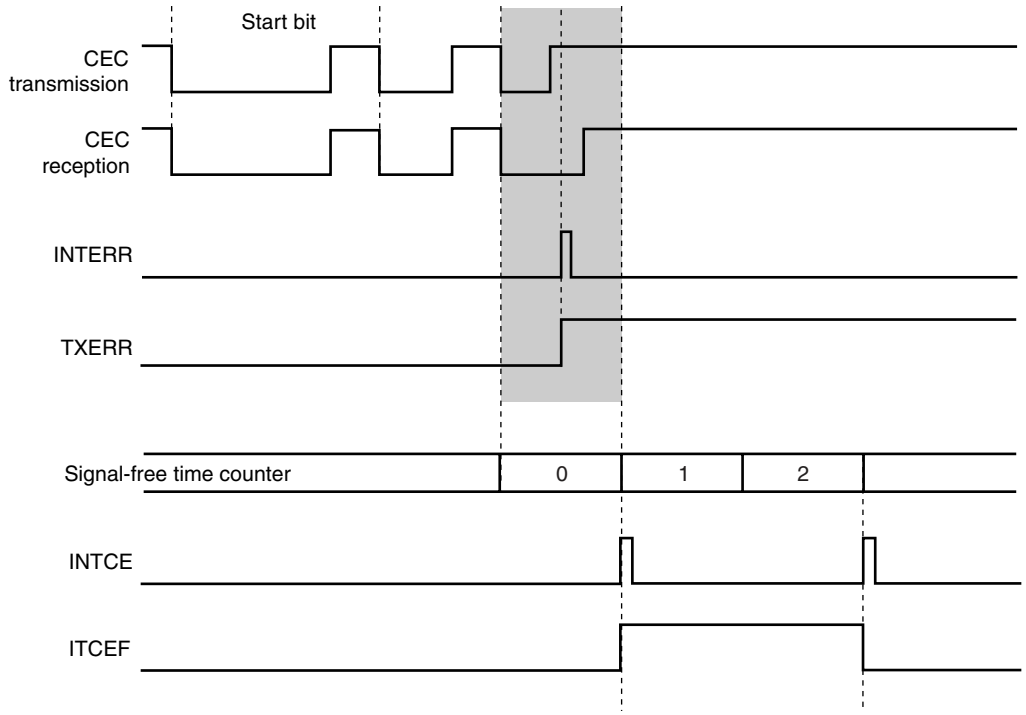
3. (8) if detecting bus lock errors (BLERRD = 1).

The details of each error are explained below.

(1) Transmission error

During initiator operation, the logic of the data the initiator has transmitted is compared with that of the CEC line receive data and a transmission error occurs when they vary. Errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). Errors are judged during the data bit period of the frame that includes the EOM bit. After an error has been detected, an error interrupt (INTERR) is generated, the transmission error flag (TXERR) is set, and the transmission is stopped at that bit. INTCE is generated at the end of bit at which transmission stopped and after the signal-free time has been counted, according to the values specified for CESEL1 and CESEL0.

Figure 17-56. Transmission Error Detection Waveform (When Three Bits Are Set as Signal-Free Time)



When a transmission error is detected, the transmit operation is stopped at the bit where the error was detected, regardless of the set value of the EOM bit of the CECCTL0 register.

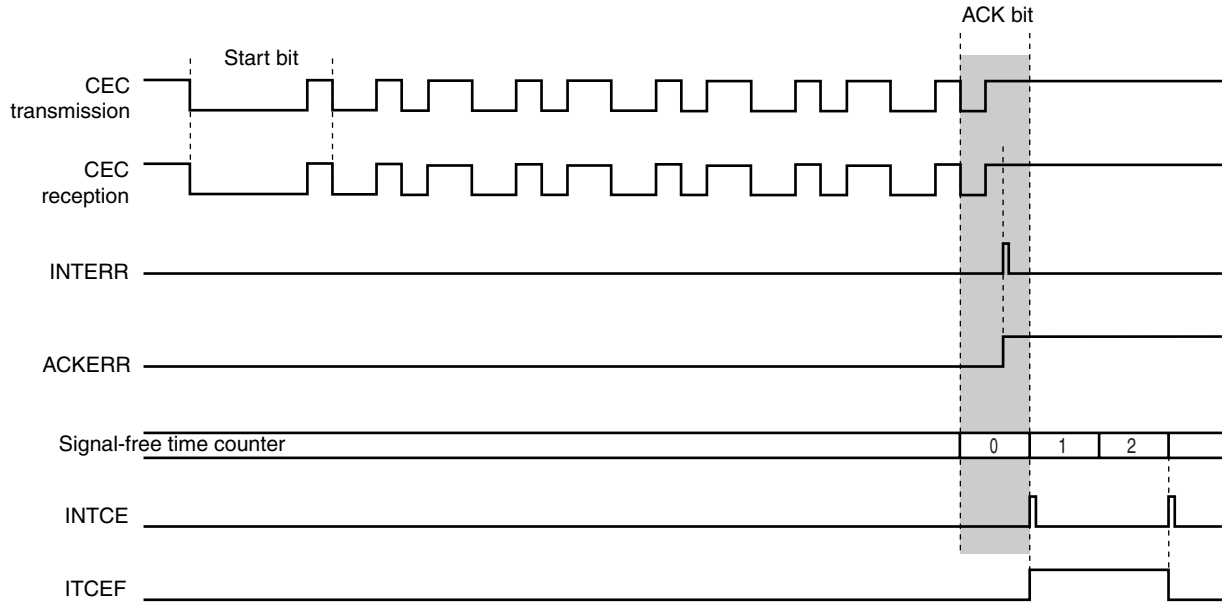
If EOM = 0 is received even though the initiator is transmitting EOM = 1, the transmission is judged as a transmission error and stopped. The follower judges that the transmission be continued, because EOM = 0. If BLERRD is set to 1, whether the received data is staying at high or low level can be detected.

(2) ACK error

During direct address transmission, an ACK error occurs when the initiator receives logical 1 at the ACK bit timing. During broadcast transmission, an ACK error occurs when the initiator receives logical 0 at the ACK bit timing.

Errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). After an error is detected, an error interrupt (INTERR) is generated and the ACK error flag (ACKERR) is set. After the end of the ACK bit, the communication standby state is entered and the signal-free time is counted. INTCE occurs once or twice depending on the set values of CESEL1 and CESEL0.

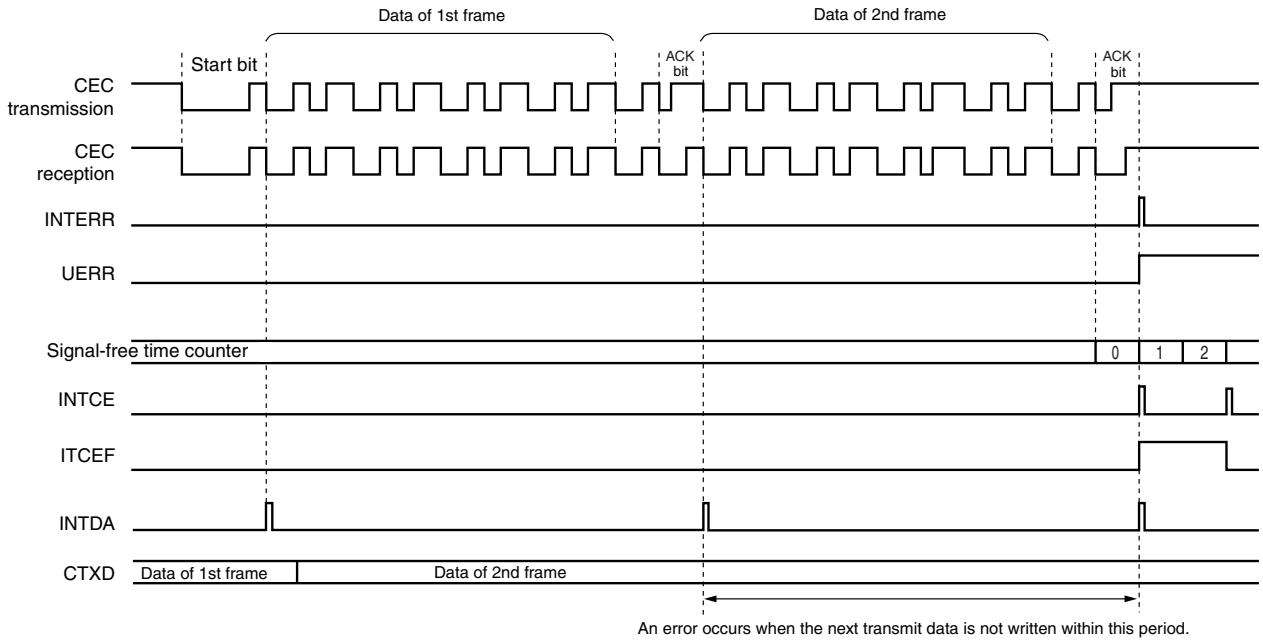
Figure 17-57. ACK Error During Direct Address Communication (When Three Bits Are Set as Signal-Free Time)



(3) Underrun error

An underrun error occurs when no data is set to the transmission buffer when transmitting the next data is started. An error interrupt (INTERR) is generated, the underrun error flag (UERR) is set, the transmission is aborted, and the communication standby state is entered. INTCE occurs once or twice depending on the set values of CESEL1 and CESEL0.

Figure 17-58. Underrun Error Timing



(4) Arbitration error

If logical 0 is received in response to logical 1 transmission during the period from the transmission start flag being set to the source address being transmitted, an arbitration error occurs. Errors between setting the transmission start trigger and outputting the start bit are judged when low level is output to the CEC transmission signal. While the source address is being transmitted, errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). After an error is detected, an error interrupt (INTERR) is generated and the arbitration error flag (AERR) is set. At this time, the transmission is aborted, but the receive operation is continued. Multiple error flags may be detected, as shown in Figure 17-59, until the source address detection period is entered.

Figure 17-59. Arbitration Timing

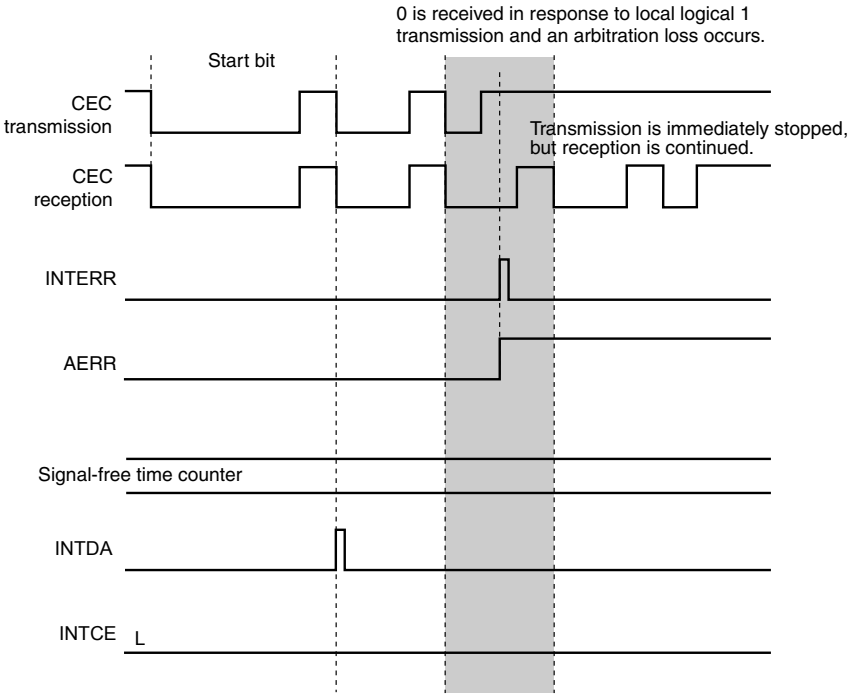
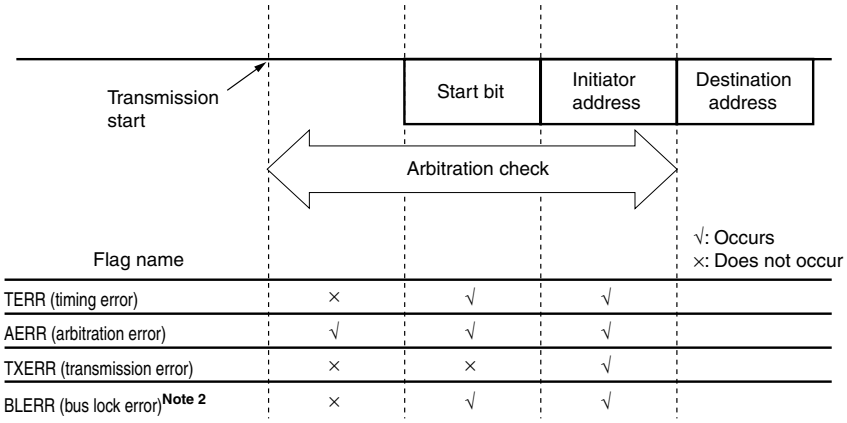


Figure 17-60. Relationships Between Arbitration Error and Other Errors



Notes 1. If STERRD is 1, the start bit timing is checked, but no error interrupt is generated.

2. If BLERRD is 1, a bus lock error is detected.

[Detailed explanation of arbitration error]

Details about the arbitration check performed from the time when the transmission start flag is set until the initiator address output period are provided below.

(a) Arbitration check by setting transmission start flag

The arbitration check is performed when two clocks have elapsed after the transmission start flag is set. If an arbitration loss is judged, an error interrupt (INTERR) is generated, the arbitration loss flag (AERR) is set, and the mode is immediately switched to reception mode.

(b) Start bit output period

If low level is detected at the reception line when the transmission start flag is set and the start bit is actually output, the arbitration loss flag is set and the mode is switched to reception mode. If STERRD = 1 and if the rising edge of the reception line is detected beyond the maximum low-level width of the start bit set by STATLH, the arbitration loss flag is set and the mode is switched to reception mode.

(c) Initiator address output period

After transmitting of the start bit is completed, a logic check is performed at the same time as starting to transmit the Initiator address. If an address earlier than the local address is detected, an error interrupt (INTERR) is generated, the timing error flag (TERR) and arbitration loss flag (AERR) are set, and the mode is immediately switched to reception mode.

(5) Timing error

Timing errors of the CEC reception signal are checked during whether initiator or follower operation. A timing error occurs if the CEC reception signal is outside the range of the compare register set. Low-level width timing errors are detected when the rising edge is detected. Timing errors for the minimum bit width are detected when the falling edge is detected, and timing errors for the maximum bit width are detected if there is no falling edge even though the maximum bit width has been exceeded. Whether to check the ACK bit timing can be selected using the ACKTEN bit of the CECCTL0 register. However, even if ACKTEN is set to 1, the maximum bit width is not checked only for the ACK bit of the last data block (when EOM is 1). The minimum bit width is checked. If a timing error with a short bit width is detected during follower operation, a low-level pulse (error handling pulse) that has a bit width 1.5 times the bit width specified using the NOMP register is transmitted. If a timing error with a short bit width is detected during initiator operation, transmission immediately stops and then the communication standby state is entered.

An error handling pulse is not transmitted if a start bit timing error is detected.

If a timing error other than one that has a short bit width is detected during initiator operation, transmission immediately stops. Reception is continued and a NACK is transmitted at the ACK bit timing during follower operation.

The generation of INTCE depends on the set values of CESEL1 and CESEL0.

If a timing error for the minimum bit width of the last ACK bit is detected, INTCE is output at the same time as INTERR.

Figure 17-61. Timing Error When Bit Width Is Short During Initiator Operation

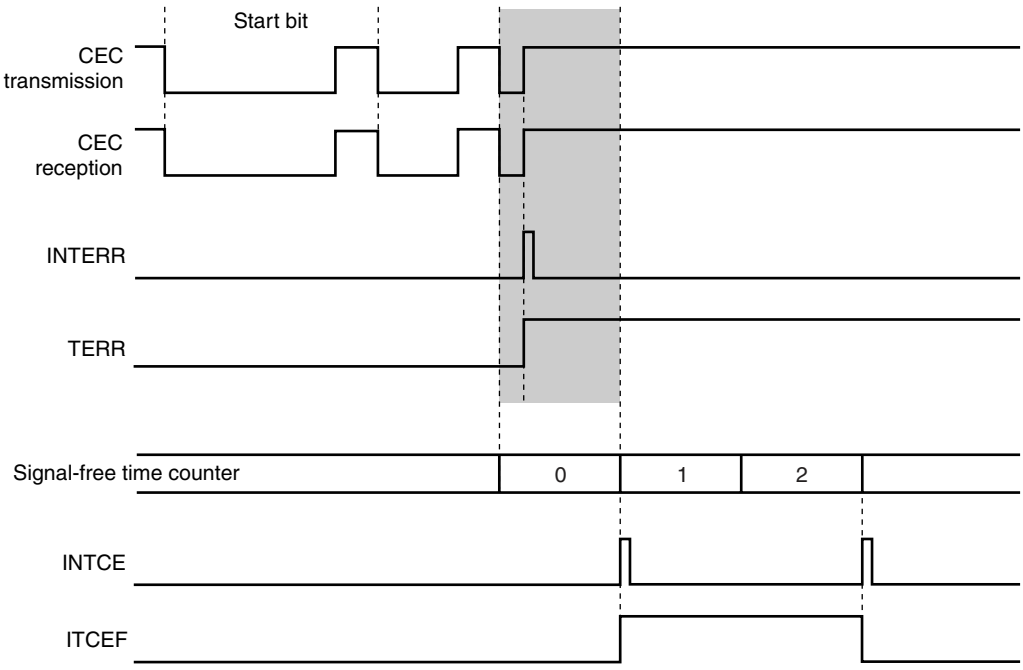


Figure 17-62. Timing Error When Bit Width Is Short During Follower Operation

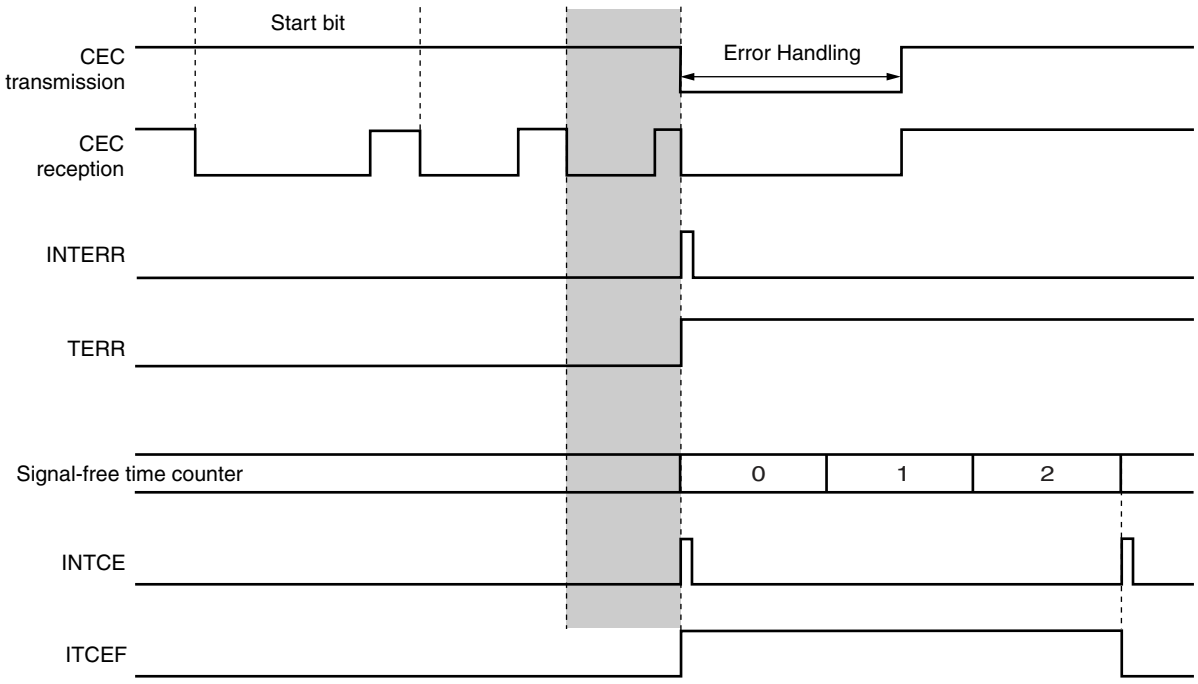
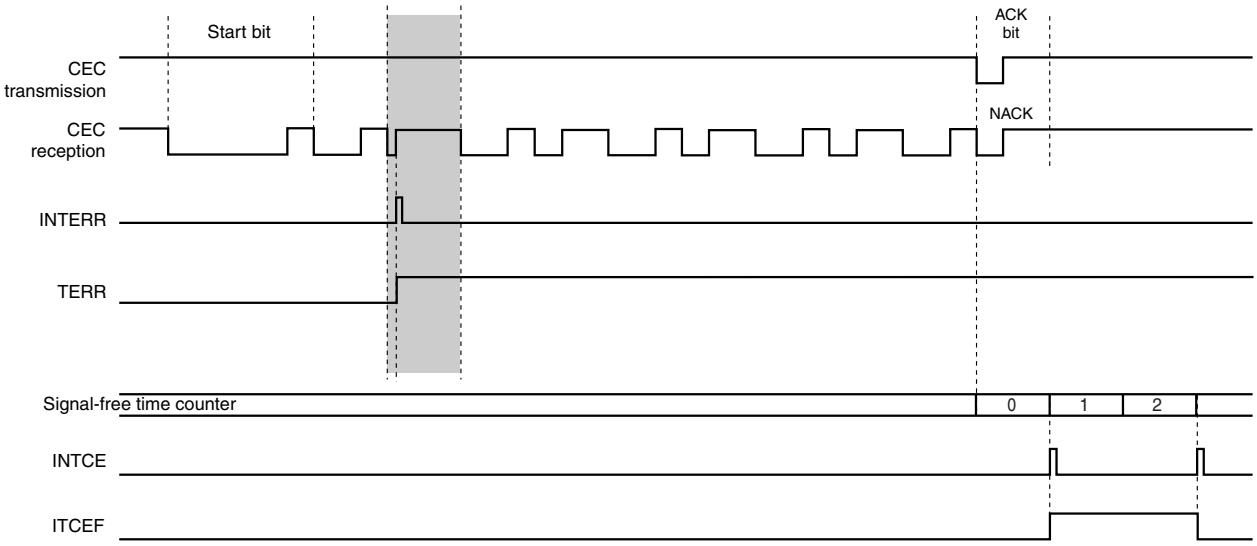


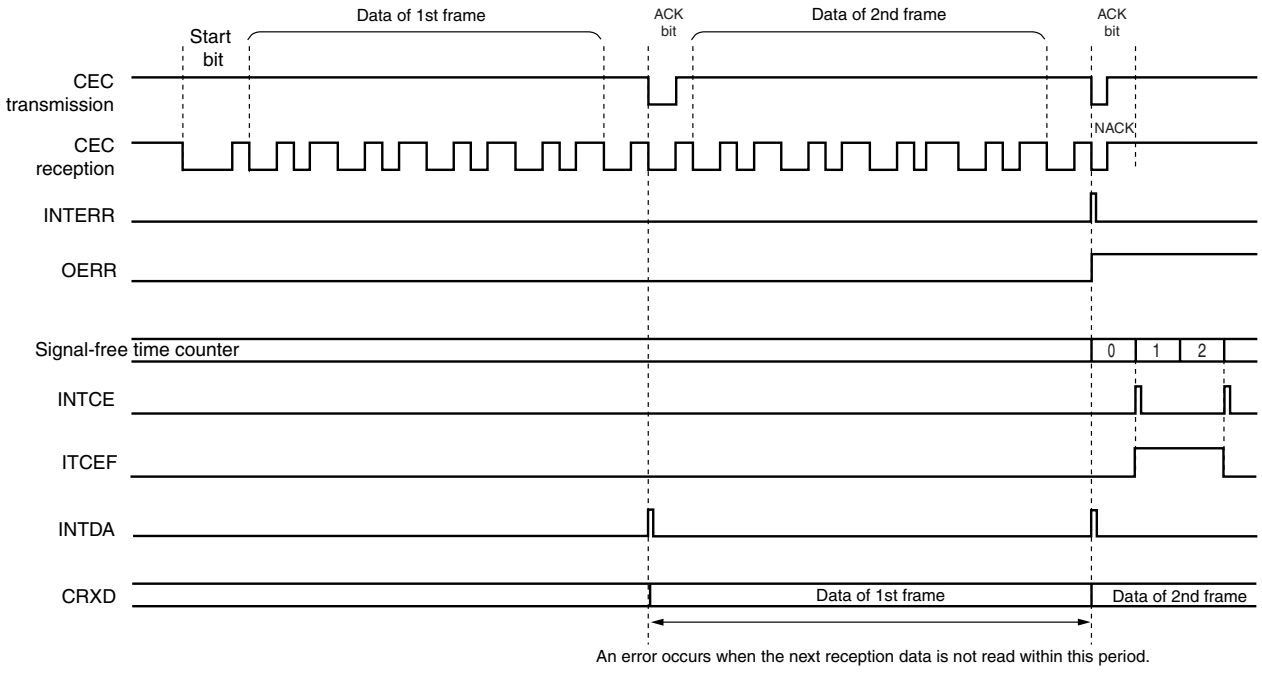
Figure 17-63. Timing Error When Bit Width Is Short During Follower Operation



(6) Overrun error

If receiving the next data is completed before reading data from the reception buffer register (CRXD) during follower operation, an overrun error occurs. An error interrupt (INTERR) is generated, the overrun error flag (OERR) is set, and the CRXD buffer value is overwritten by a new value. Afterward, logical 1 is returned during direct address communication and logical 0 is returned during broadcast communication at the ACK transmission timing of the block in which an overrun error occurred. The failure of reception is reported to the initiator, and the reception standby state is entered. INTCE operates according to the setting of CESEL1 and CESEL0.

Figure 17-64. Overrun Error (When Three Bits Are Set as Signal-Free Time)



(7) Bus lock error

A bus lock error occurs if the CEC reception signal stays at high or low level for a period corresponding to 2.5 times the data bit width specified using NOMP[Note]. Whether an input signal is stuck to high/low level can be detected by setting BLERRD to 1. If an error is detected, an error interrupt (INTERR) is generated, the bus lock error flag (BLERR) is set, the communication standby state is entered, and the signal-free time is counted. INTCE operates according to the setting of CESEL1 and CESEL0.

Note The above applies after the falling edge is detected and communication is started.

Caution If bus lock errors are not to be detected, the communication may not be terminated by an intended operation. In that case, the retransmission timing is lost and abnormal communication is continued. Consequently, set BLERRD and set, so that bus lock errors are detected during a receive operation.

Figure 17-65. During Initiator Operation (When Three Bits Are Set as Signal-Free Time)

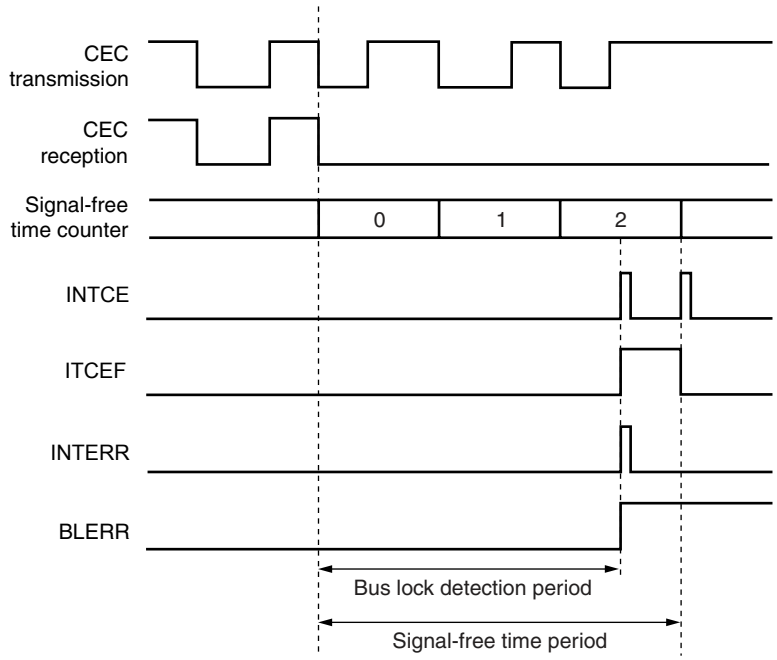
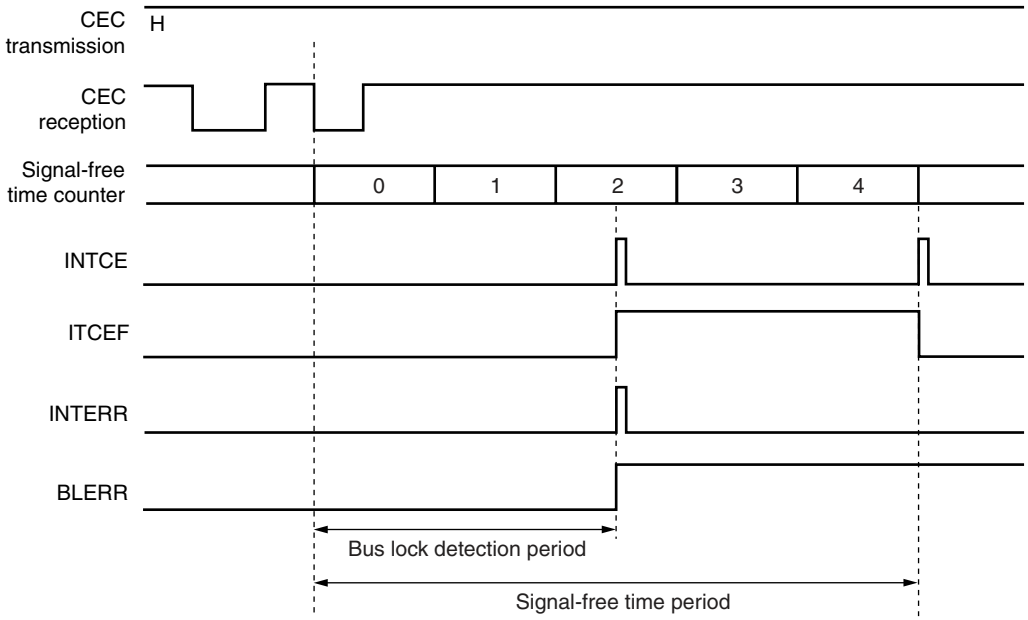


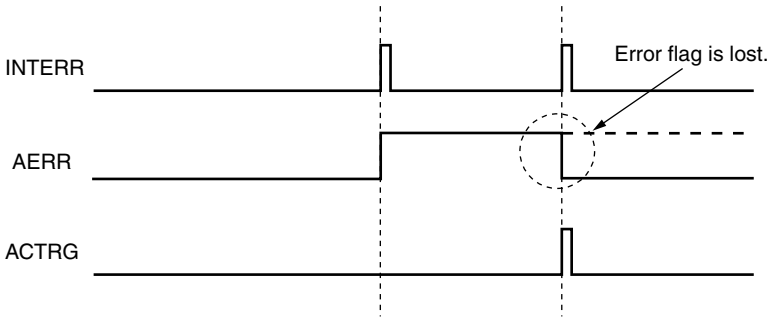
Figure 17-66. During Follower Operation (When Five Bits Are Set as Signal-Free Time)



17.7.7 Clearing error flag

An error flag set to the CEC communication error status register (CECES) can be cleared by setting 1 to the corresponding bit of the CEC communication error flag clear trigger register (CECFC). Figure 17-67 shows the case when an arbitration error occurs. An arbitration error flag can be cleared by setting 20H to CECFC. An error flag that has been set once will be cleared by using the clear trigger register or setting CECE = 0.

Figure 17-67. When Two Identical Errors Have Occurred and Conflict with Clear Trigger



Caution If clearing an error flag and setting an error flag to be cleared conflict, the second error flag may not be set.

17.7.8 Signal-free time

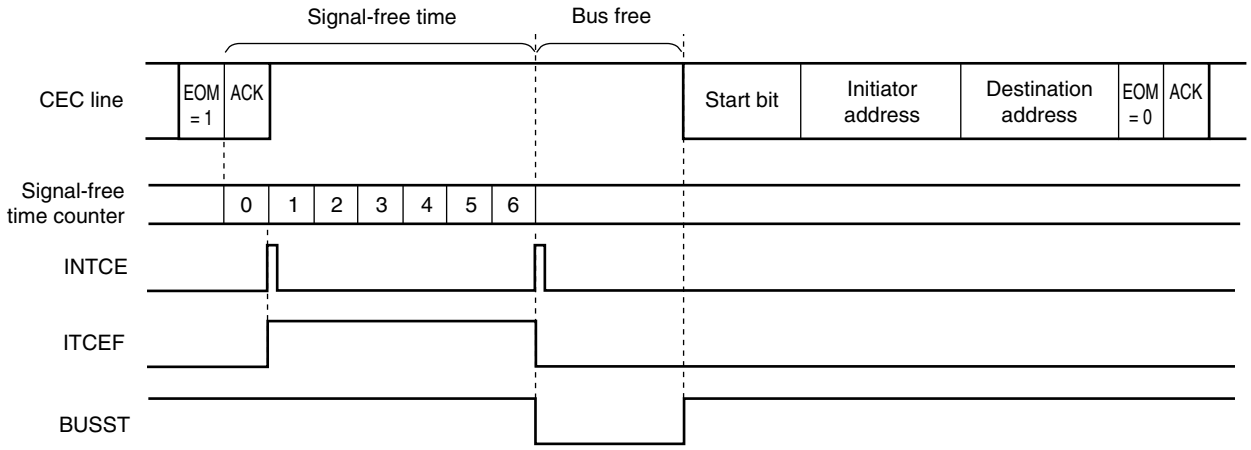
The end of the signal-free time is reported by generating a communication complete interrupt by detecting a match with the specified time (3, 5, or 7 bits of the bit width specified using the NOMP register). The number of bits of the signal-free time is specified using SFT1 and SFT0. When a communication complete interrupt is generated can be selected by setting up CESEL1 and CESEL0. Counting the signal-free time always starts when the falling edge of the received data is detected. During normal communication, counting the signal-free time starts after the falling edge of the ACK bit is detected, if EOM is 1.

If the communication standby state is forcibly entered due to the occurrence of an error, counting is started as soon as the communication standby state is entered.

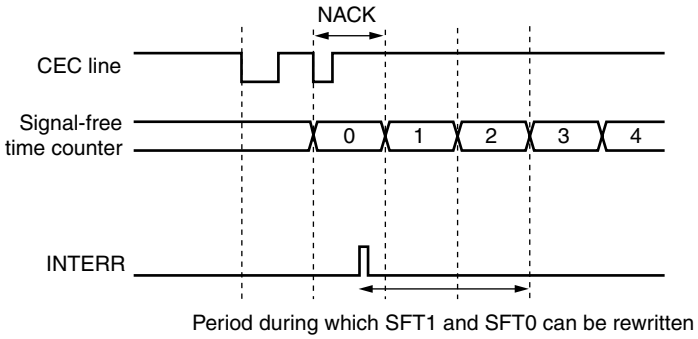
If an error handling pulse (a low-level pulse that has a bit width 1.5 times the bit width specified using the DATB register) is acknowledged, the signal-free time is counted starting at the falling edge of the error handling pulse signal.

Figure 17-68 shows an operation example in which the detection of a 7-data bit width signal-free time that is set to CESEL1 = 1, CESEL0 = 0, SFT1 = 1, and SFT0 = 0 is set.

Figure 17-68. Signal-Free Time Operation



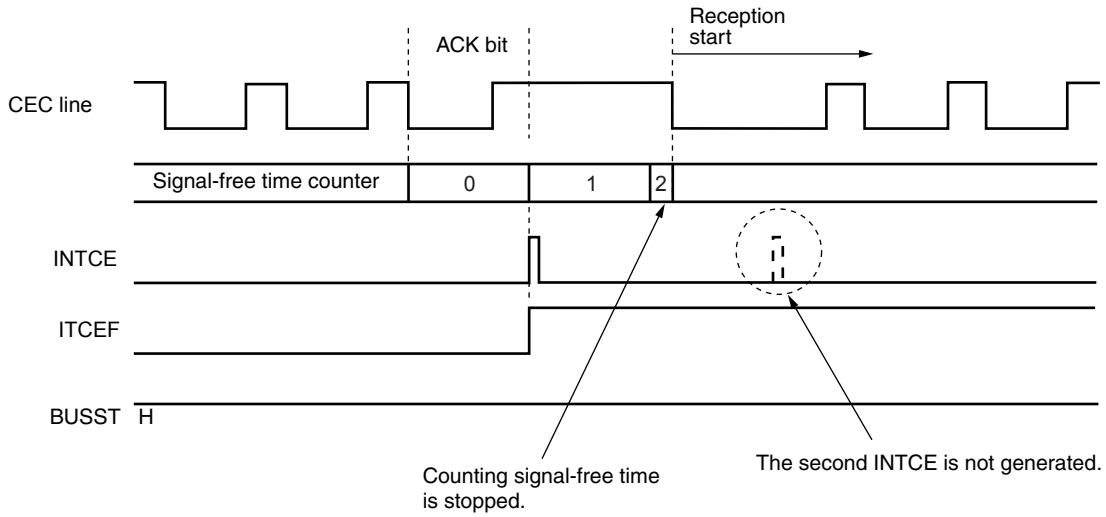
Caution Rewriting the values of the SFT1 and SFT0 registers to the number of bits smaller than the current number of bits while counting the signal-free time must be completed until the rewritten bit count values match. If rewriting is not performed in time, the counter overflows and the signal-free time period continues until the number of bits match again. An example in which the data bit width is changed from five to three bits is shown below.



<Starting receive operation during signal-free time>

If a falling edge of the CEC reception signal is detected while counting the signal-free time, a receive operation is started. INTCE is not output even if the output of INTCE after counting the signal-free time is set, because the count operation of the signal-free time counter is stopped at this time.

Similarly, if 1 is written to the transmission trigger while counting the signal-free time, a transmission is started and the count operation of the signal-free time counter is stopped.



CHAPTER 18 REMOTE CONTROLLER RECEIVER

Caution If remote controller receiver is used, clock output (PCL) and INTP6 cannot be used.

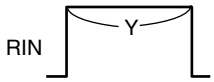
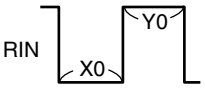
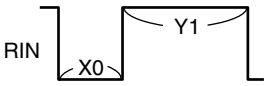
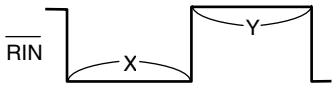
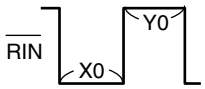
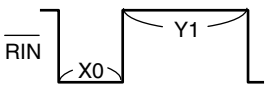
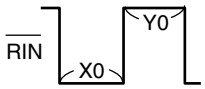
18.1 Remote Controller Receiver Functions

The remote controller receiver uses the following three remote controller reception modes.

- Type A reception mode … Guide pulse (half clock) provided
- Type B reception mode … Guide pulse (1 clock) provided
- Type C reception mode … Guide pulse not provided

Remark Guide pulse (reader code): Remote control signal header section

Table 18-1. Format of Each Reception Mode (Waveform to Be Input to RMINP)

Reception mode	Guide pulse	Data 0	Data 1	Example of Format
Type A				See **18. 5. 1**
Type B				See **18. 5. 3**
Type C	Not provided			See **18. 5. 5**

Remarks 1. The waveform of data 1 is longer than that of data 0 (Y0 < Y1, Z0 < Z1).

2. X, X0: low-level width
Y, Y0, Y1: high-level width
Z, Z0, Z1: cycle

3. RIN: Input signal from the remote control receive data input (RIN) pin
$\overline{\text{RIN}}$: Inverted RIN signal
RMINP: See **Figure 18-1**.

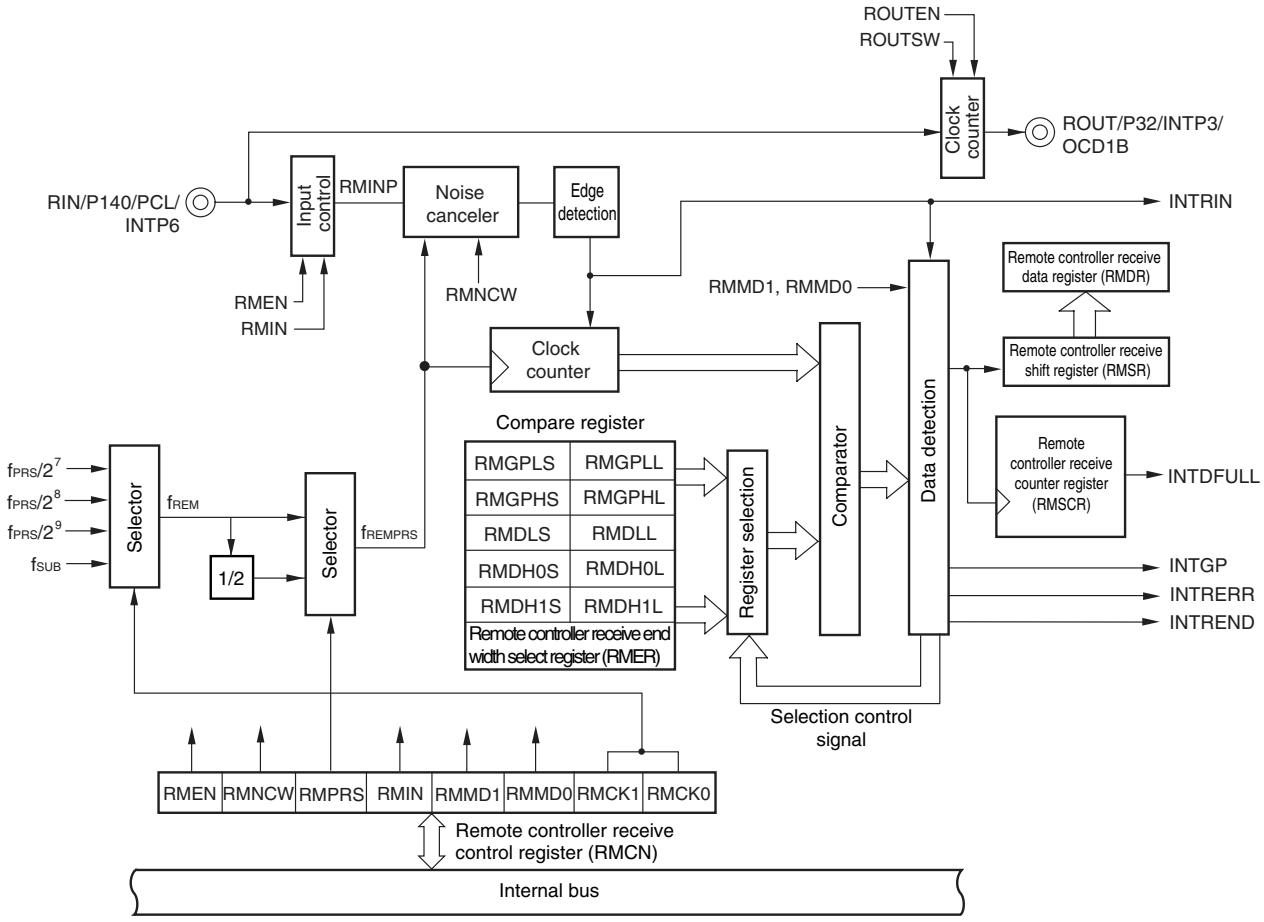
18.2 Remote Controller Receiver Configuration

The remote controller receiver includes the following hardware.

Table 18-1. Remote Controller Receiver Configuration

Item	Configuration
Registers	Remote controller receive data register (RMDR) Remote controller receive counter register (RMSCR) Remote controller receive shift register (RMSR) Remote controller receive GPLS compare register (RMGPLS) Remote controller receive GPLL compare register (RMGPLL) Remote controller receive GPHS compare register (RMGPHS) Remote controller receive GPHL compare register (RMGPHL) Remote controller receive DLS compare register (RMDLS) Remote controller receive DLL compare register (RMDLL) Remote controller receive DH0S compare register (RMDH0S) Remote controller receive DH0L compare register (RMDH0L) Remote controller receive DH1S compare register (RMDH1S) Remote controller receive DH1L compare register (RMDH1L) Remote controller receive end width select register (RMER)
Common registers	Remote controller receive interrupt status register (INTS) Remote controller receive interrupt status clear register (INTC) Remote controller receive control register (RMCN) Remote controller receive data output control register (RMSW) Port mode registers 3, 14 (PM3, PM14) Port registers 3, 14 (P3, P14)

Figure 18-1. Block Diagram of Remote Controller Receiver



(1) Remote controller receive shift register (RMSR)

This is an 8-bit register for reception of remote controller data.

Data is stored in bit 7 first. Each time new data is stored, the stored data is shifted to the lower bits. Therefore, the latest data is stored in bit 7, and the first data is stored in bit 0.

When data 0 is normally received, “0” will be stored in bit 7, and when data 1 is normally received, “1” will be stored in bit 7.

RMSR is read with an 8-bit memory manipulation instruction.

Reset signal generation sets RMSR to 00H.

Also, RMSR is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- INTDFULL is generated.
- RMSR is read after INTREND has been generated.

Figure 18-2. Format of Remote Controller Receive Shift Register (RMSR)

Address: FF6FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RMSR								

Cautions 1. Reading RMSR is disabled during remote controller reception. Complete reception, then read RMSR. When the reading operation is complete, RMSR is cleared. Therefore, values once read are not guaranteed.

2. If eight bits of remote control signals are received by RMSR, the contents of RMSR will be transferred to remote control receive data register (RMDR) and INTDFULL will be generated. At this time, RMSR will be cleared.

3. After INTREND is generated, be sure to read RMSCR before reading RMSR.
RMSCR and RMSR will be automatically cleared after RMSR is read.
After INTREND is generated, the next data cannot be received until RMSR is read.

(2) Remote controller receive data register (RMDR)

This register holds the remote controller reception data. When the remote controller receive shift register (RMSR) overflows, the data in RMSR is transferred to RMDR. Bit 7 stores the last data, and bit 0 stores the first data. INTDFULL is generated at the same time as data is transferred from RMSR to RMDR.

RMDR is read with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDR to 00H.

When the remote controller operation is disabled (RMEN = 0), RMDR is cleared to 00H.

Figure 18-3. Format of Remote Controller Receive Data Register (RMDR)

Address: FF54H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RMDR								

Caution When INTDFULL has been generated, read RMDRn before the next 8-bit data is received. If the next INTDFULL is generated before the read operation is complete, RMDR is overwritten.

(3) Remote controller receive counter register (RMSCR)

This is a 3-bit counter register used to indicate the number of valid bits remaining in the remote controller receive shift register (RMSR) when remote controller reception is complete (INTREND is generated). Reading the values of this register allows confirmation of the number of bits, even if the received data is in a format other than an integral multiple of 8 bits.

RMSCR is read with an 8-bit memory manipulation instruction.

Reset signal generation sets RMSCR to 00H.

It is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- RMSR is read after INTREND has been generated.

Figure 18-4. Format of Remote Controller Receive Counter Register (RMSCR)

Address: FF4EH After reset: 00H R

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

Caution When INTREND has been generated, immediately read RMSCR before reading RMSR. If reading occurs at another timing, the value is not guaranteed.

Example of relationship between receive bit number and RMSCR register

Receive bit number	RMSCR register value	Receive bit number	RMSCR register value
3	3 (3 – 8 × 0)	8	0 (8 – 8 × 1)
15	7 (15 – 8 × 1)	16	0 (16 – 8 × 2)

Table 18-5. Operation Examples of RMSR, RMSCR, and RMDR Registers When Receiving 1010101011111111B (16 Bits)

	RMSR								RMSCR	RMDR
	7	6	5	4	3	2	1	0		
After reset	0	0	0	0	0	0	0	0	00H	00000000B
Receiving 1 bit	1	0	0	0	0	0	0	0	01H	00000000B
Receiving 2 bits	0	1	0	0	0	0	0	0	02H	00000000B
Receiving 3 bits	1	0	1	0	0	0	0	0	03H	00000000B
…	…	…	…	…	…	…	…	…	…	…
Receiving 7 bits	1	0	1	0	1	0	1	0	07H	00000000B
Receiving 8 bits	0	1	0	1	0	1	0	1	00H	00000000B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDRn transfer	0	0	0	0	0	0	0	0	00H	01010101B
Receiving 9 bits	1	0	0	0	0	0	0	0	01H	01010101B
Receiving 10 bits	1	1	0	0	0	0	0	0	02H	01010101B
…	…	…	…	…	…	…	…	…	…	…
Receiving 16 bits	1	1	1	1	1	1	1	1	00H	01010101B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDRn transfer	0	0	0	0	0	0	0	0	00H	11111111B

18.3 Registers to Control Remote Controller Receiver

The remote controller receiver is controlled by the following register.

- Remote controller receive interrupt status register (INTS)
- Remote controller receive interrupt status clear register (INTC)
- Remote controller receive control register (RMCN)
- Remote controller receive data output control register (RMSW)
- Port mode registers 3, 14 (PM3, PM14)
- Port registers 3, 14 (P3, P14)

(1) Remote controller receive interrupt status register (INTS)

This register is used to identify which interrupt request among the remote control receive interrupts (INTRERR, INTGP, INTREND, INTDFULL) has occurred.

INTS is set with a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets INTS to 00H.

Figure 18-6. Format of Remote Controller Receive Interrupt Status Register (INTS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTS	0	0	0	0	INTS DFULL	INTS REND	INTS GP	INTS RERR	FFF9H	00H	R

INTS DFULL	Interrupt request by reading of 8-bit shift data
0	Interrupt request by reading of 8-bit shift data has not occurred
1	Interrupt request by reading of 8-bit shift data has occurred

INTS REND	Request by data reception completion interrupt
0	Request by data reception completion interrupt has not occurred
1	Request by data reception completion interrupt has occurred

INTS GP	Guide pulse detection interrupt
0	Guide pulse detection interrupt request has not occurred
1	Guide pulse detection interrupt request has occurred

INTS RERR	Interrupt request by remote control receive error
0	Interrupt request by remote control receive error has not occurred
1	Interrupt request by remote control receive error has occurred

Caution The INTS register will not be cleared even if it is read. Use the INTC register to clear the INTS register.

(2) Remote controller receive interrupt status clear register (INTC)

This register is used to control the remote controller receive interrupt status register (INTS).

INTC is set with a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets INTC to 00H.

Figure 18-7. Format of Remote Controller Receive Interrupt Status Clear Register (INTC)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTC	0	0	0	0	INTC DFULL	INTC REND	INTC GP	INTC RERR	FFFAH	00H	R/W

INTC DFULL	Interrupt identification bit control by reading of 8-bit shift data
0	INTSDFULL bit not changed
1	INTSDFULL bit cleared

INTC REND	Data reception completion Interrupt identification bit control
0	INTSREND bit not changed
1	INTSREND bit cleared

INTC GP	Guide pulse detection interrupt identification bit control
0	INTSGP bit not changed
1	INTSGP bit cleared

INTC RERR	Interrupt identification bit control by remote control receive error
0	INTSRERR bit not changed
1	INTSRERR bit cleared

(3) Remote controller receive control register (RMCN)

This register is used to enable/disable remote controller reception and to set the noise elimination width, clock internal division, input invert signal, and source clock.

RMCN is set with a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets RMCN to 00H.

Figure 18-8. Format of Remote Controller Receive Control Register (RMCN)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RMCN	RMEN	NCW	PRSEN	RMIN	RMMD1	RMMD0	RMCK1	RMCK0	FF3FH	00H	R/W

RMEN	Control of remote controller receive operation
0	Disable remote controller reception
1	Enable remote controller reception

NCW	Noise elimination width control signal
0	Eliminate noise less than $1/f_{REMPRS}$
1	Eliminate noise less than $2/f_{REMPRS}$

PRSEN	Internal clock division control signal
0	Clock not divided internally ($f_{REMPRS} = f_{REM}$)
1	Clock internally divided into two ($f_{REMPRS} = f_{REM}/2$)

RMIN	Remote controller input invert control
0	Does not invert signal of remote control receive data input pin.
1	Inverts signal of remote control receive data input pin.

RMMD1	RMMD0	Remote controller reception mode
0	0	Type A reception mode (Guide pulse (half clock) provided)
0	1	Type B reception mode (Guide pulse (1 clock) provided)
1	0	Type C reception mode (Guide pulse not provided)
1	1	Setting prohibited

RMCK1	RMCK0	Selection of source clock (f_{REM}) of remote controller counter				
		Selection clock	f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	$f_{PRS}/2^6$	39.063 kHz	78.125 kHz	156.250 kHz	312.500 kHz
0	1	$f_{PRS}/2^7$	15.625 kHz	39.063 kHz	78.125 kHz	156..250 kHz
1	0	$f_{PRS}/2^8$	7.813 kHz	19.531 kHz	39.063 kHz	78.125 kHz
1	1	f_{SUB}	32.768 kHz			

Caution To change the values of NCW, PRSEN, RMIN, RMMD1, RMMD0, RMCK1, and RMCK0, disable remote controller reception (RMEN = 0) first.

Remark 1. f_{REM}: Source clock of remote controller counter (selected by bits 0 and 1 (RMCK0 and RMCK1))

Remarks 2. f_{REMPRS}: Operation clock inside remote controller receiver

3. f_{PRS}: Peripheral hardware clock frequency

4. f_{SUB}: Oscillation frequency of subsystem clock

(4) Remote controller receive data control register (RMSW)

This register is output control register for the output from ROUT pin of the remote controller reception data input from the RIN pin without eliminating noise and without decoding.

RMSW is set with a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets RMSW to 00H.

Figure 18-9. Format of Remote Controller Receive Data Control Register (RMSW)

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
RMSW	0	0	0	0	0	0	ROUTSW	ROUTEN	FFB7H	00H	R/W

ROUTSW	Control of ROUT pin output operation
0	Disable ROUT pin output
1	Enable ROUT pin output

ROUTEN	Selection of through-output/invert-output
0	The RIN pin input signal is through-output from the ROUT pin.
1	The RIN pin input signal is invert-output from the ROUT pin.

(5) Port mode registers 3 and 14 (PM3, PM14)

These registers set port 3 and 14 input/output in 1-bit units.

When using the P32/ROUT/INTP3/OCD1B pin for remote controller receive data output, clear PM32 to 0. The output latches of P32 to 1.

When using the P140/PCL/INTP6/RIN pin for emote controller receive data input, set PM140 to 1. The output latches of P140 at this time may be 0 or 1.

PM3 and PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 8-10. Format of Port Mode Registers 3, 14 (PM3, PM14)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

Address: FF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	PM141[Note]	PM140

PMmn	P4n pin I/O mode selection (m = 3, 14; n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark 78K0/KE2-C only

18.4 Compare Registers of Remote Controller Receiver

This register is used to set timing judgment of remote controller reception signal.

(1) Remote controller receive GPLS compare register (RMGPLS) (Type B reception mode only)

This register is used to detect the low level of a remote controller guide pulse (short side).

RMGPLS is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPLS to 00H.

Figure 18-11. Format of Remote Controller Receive GPLS Compare Register (RMGPLS)

Address: FA44H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMGPLS								

Caution Write this register while RMEN = 0.

(2) Remote controller receive GPLL compare register (RMGPLL) (Type B reception mode only)

This register is used to detect the low level of a remote controller guide pulse (long side).

RMGPLL is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPLL to 00H.

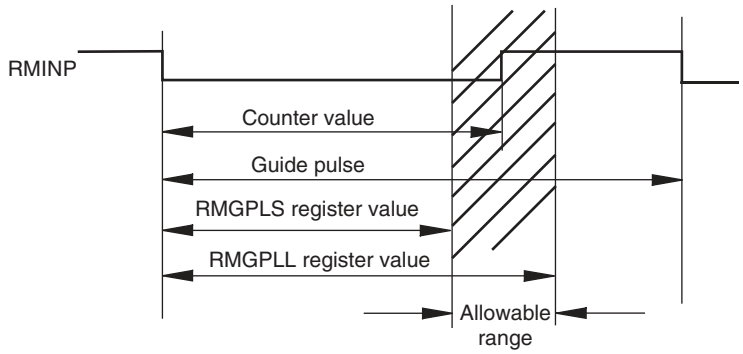
Figure 18-12. Format of Remote Controller Receive GPLL Compare Register (RMGPLL)

Address: FA45H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMGPLL								

Caution Write this register while RMEN = 0.

<Allowable range of guide pulse low-level width>



If RMGPLS ≤ counter value < RMGPLL is satisfied, it is assumed that the low level of the guide pulse has been successfully received.

If the guide pulse low level is received normally, the guide pulse high-level width will be measured.

(3) Remote controller receive GPHS compare register (RMGPHS) (Type A or B reception mode only)

This register is used to detect the high level of a remote controller guide pulse (short side).

RMGPHS is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPHS to 00H.

Figure 18-13. Format of Remote Controller Receive GPHS Compare Register (RMGPHS)

Address: FA46H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMGPHS								

Caution Write this register while RMEN = 0.

(4) Remote controller receive GPHL compare register (RMGPHL) (Type A or B reception mode only)

This register is used to detect the high level of a remote controller guide pulse (long side).

RMGPHL is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPHL to 00H.

Figure 18-14. Format of Remote Controller Receive GPHL Compare Register (RMGPHL)

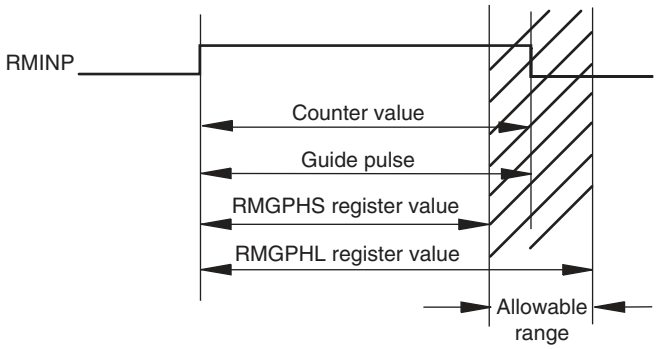
Address: FA47H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMGPHL								

Caution Write this register while RMEN = 0.

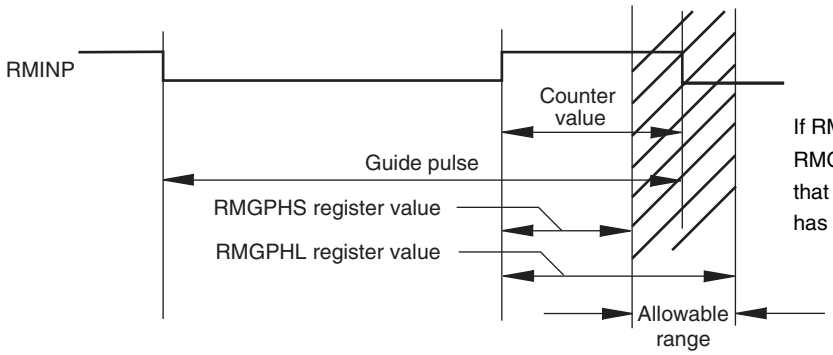
<Allowable range of guide pulse high-level width>

(a) Type A reception mode



If RMGPHS ≤ counter value < RMGPHL is satisfied, it is assumed that the high level of the guide pulse has been successfully received.

(b) Type B reception mode



If RMGPHS ≤ counter value < RMGPHL is satisfied, it is assumed that the high level of the guide pulse has been successfully received.

If the guide pulse high level is received normally, an INTGP interrupt signal will be output and the low-level width of the next data will be measured.

(5) Remote controller receive DLS compare register (RMDLS)

This register is used to detect the low level of a remote controller data (short side).

RMDLS is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDLS to 00H.

Figure 18-15. Format of Remote Controller Receive DLS Compare Register (RMDLS)

Address: FA48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDLS								

Caution Write this register while RMEN = 0.

(6) Remote controller receive DLL compare register (RMDLL)

This register is used to detect the low level of a remote controller data (long side).

RMDLL is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDLL to 00H.

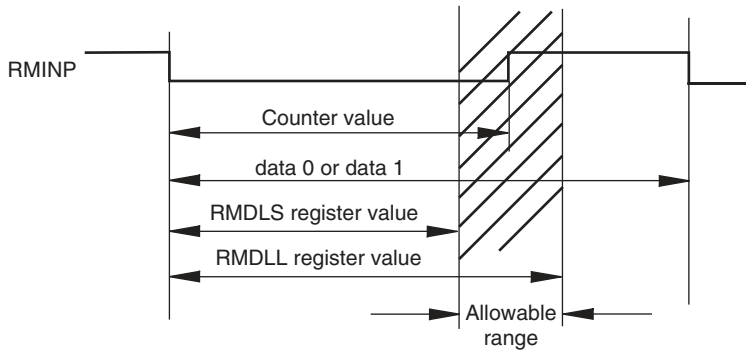
Figure 18-16. Format of Remote Controller Receive DLL Compare Register (RMDLL)

Address: FA49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDLL								

Caution Write this register while RMEN = 0.

<Allowable range of data 0 or data 1 low-level width>



If RMDLS ≤ counter value < RMDLL is satisfied, it is assumed that the low level of the data 0 or data1 has been successfully received.

If the data low-level is received normally, the data high-level width will be measured.

(7) Remote controller receive DH0S compare register (RMDH0S)

This register is used to detect the high level of a remote controller data 0 (short side).

RMDH0S is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH0S to 00H.

Figure 18-17. Format of Remote Controller Receive DH0S Compare Register (RMDH0S)

Address: FA4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDH0S								

Caution Write this register while RMEN = 0.

(8) Remote controller receive DH0L compare register (RMDH0L)

This register is used to detect the high level of a remote controller data 0 (long side).

RMDH0L is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH0L to 00H.

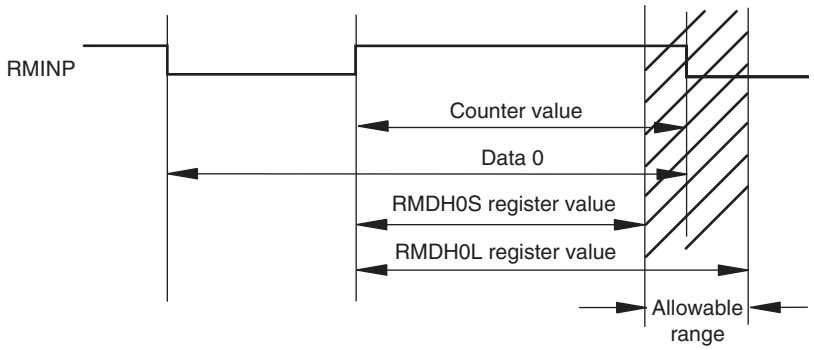
Figure 18-18. Format of Remote Controller Receive DH0L Compare Register (RMDH0L)

Address: FA48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDH0L								

Caution Write this register while RMEN = 0.

<Allowable range of data 0 high-level width>



If RMDH0S ≤ counter value < RMDH0L is satisfied, it is assumed that the high level of data 0 has been successfully received, and therefore RMSR receives the data.

If data 0 is received normally, the shift register will be right-shifted, “0” will be stored in bit 7, and the low-level width of the next data will be measured.

Caution Be sure to set the high-level width of data 0 shorter than that of data 1.

(9) Remote controller receive DH1S compare register (RMDH1S)

This register is used to detect the high level of remote controller data 1 (short side).

RMDH1S is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH1S to 00H.

Figure 18-19. Format of Remote Controller Receive DH1S Compare Register (RMDH1S)

Address: FA4CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDH1S								

Caution Write this register while RMEN = 0.

(10) Remote controller receive DH1L compare register (RMDH1L)

This register is used to detect the high level of remote controller data 1 (long side).

RMDH1L is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH1L to 00H.

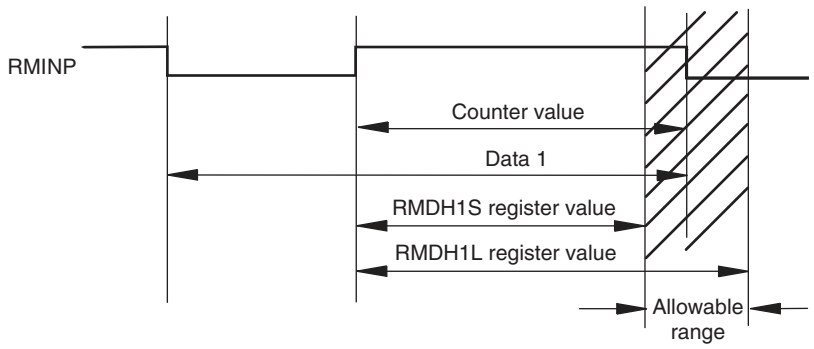
Figure 18-20. Format of Remote Controller Receive DH1L Compare Register (RMDH1L)

Address: FA4DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDH1L								

Caution Write this register while RMEN = 0.

<Allowable range of data 1 high-level width>



If RMDH1S ≤ counter value < RMDH1L is satisfied, it is assumed that the high level of data 1 has been successfully received, and therefore RMSR receives the data.

If data 1 is received normally, the shift register will be right-shifted, “1” will be stored in bit 7, and the low-level width of the next data will be measured.

Caution Be sure to set the high-level width of data 1 longer than that of data 0.

(11) Remote controller receive end width select register (RMER)

This register determines the interval between the timing at which the INTREND signal is output.

The conditions for entering the end state differ for each reception mode.

RMER is set by 8-bit memory manipulation instructions.

Reset signal generation sets RMER to 00H.

Figure 18-21. Format of Remote Controller Receive End width Select Register (RMER)

Address: FA4EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMER								

Caution Write a value while RMEN = 0.

$$RMER = \left[\frac{T_{WE} \times (1 - a/100)}{1/f_{REMPRS}} \right]_{INT} - 1$$

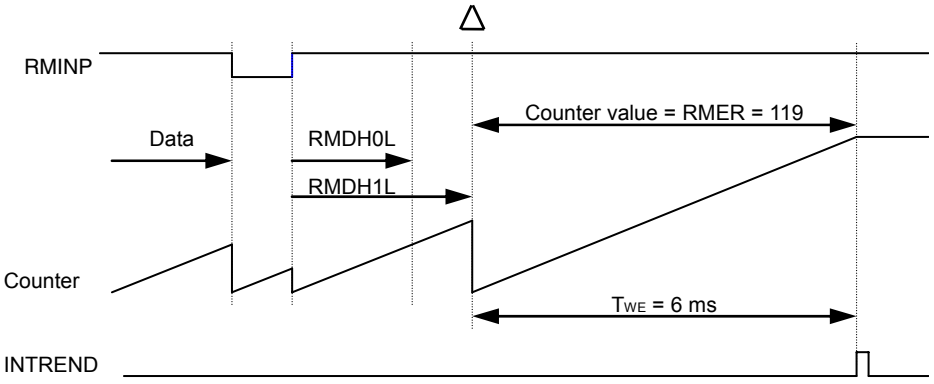
$1/f_{REMPRS}$: Width of internal operation clock cycle after division control by RMPRS

a: Tolerance (%)

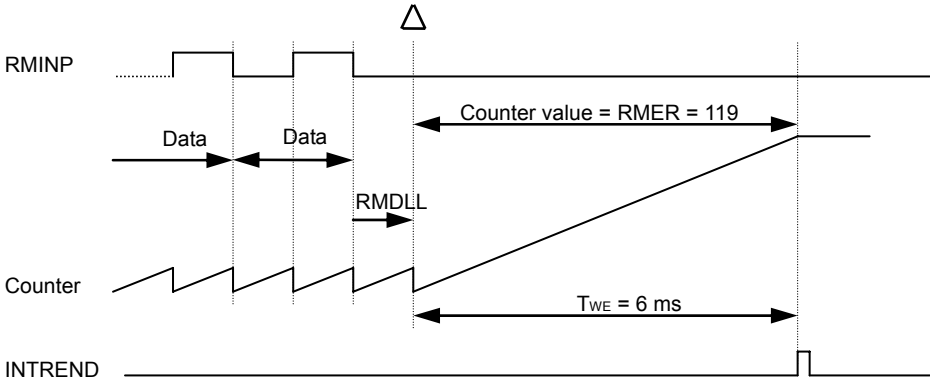
$[\ ]_{INT}$: Round down the fractional portion of the value produced by the formula in the brackets.

T_{WE}: End width of RMINP input

Example 1: With type B or C, if the provisional end time is 6 ms and the internal operation clock cycle is 50 μs, the RMER register value will be {6 ms/50 μs} – 1 = 119.
A length of (RMDH1L + 1 + RMER + 1) x 50 μs, however, will be required as the high-level width of RMINP. If this cannot be met, an error interrupt (INTRERR) will be generated.



Example 2: With type A, if the provisional end time is 6 ms and the internal operation clock cycle is 50 μs, the RMER register value will be {6 ms/50 μs} – 1 = 119.
A length of (RMDLL + 1 + RMER + 1) x 50 μs, however, will be required as the low-level width of RIN. If this cannot be met, an error interrupt (INTRERR) will be generated.



18.5 Operation of Remote Controller Receiver

The following remote controller reception mode is used for this remote controller receiver.

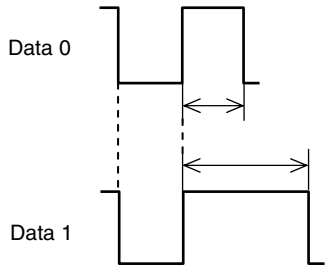
Reception mode	Guide Pulse Low-Level Period	Guide Pulse High-Level Period	Input Waveform to RMINP Note	Detection method
Type A	Not used	Used	RIN	High/low level width
Type B	Used	Used	$\overline{\text{RIN}}$	
Type C	Not used	Not used		

Note Refer to **Figure 18-1**.

The data waveforms are as follows.

- In type A, type B, and type C reception mode,

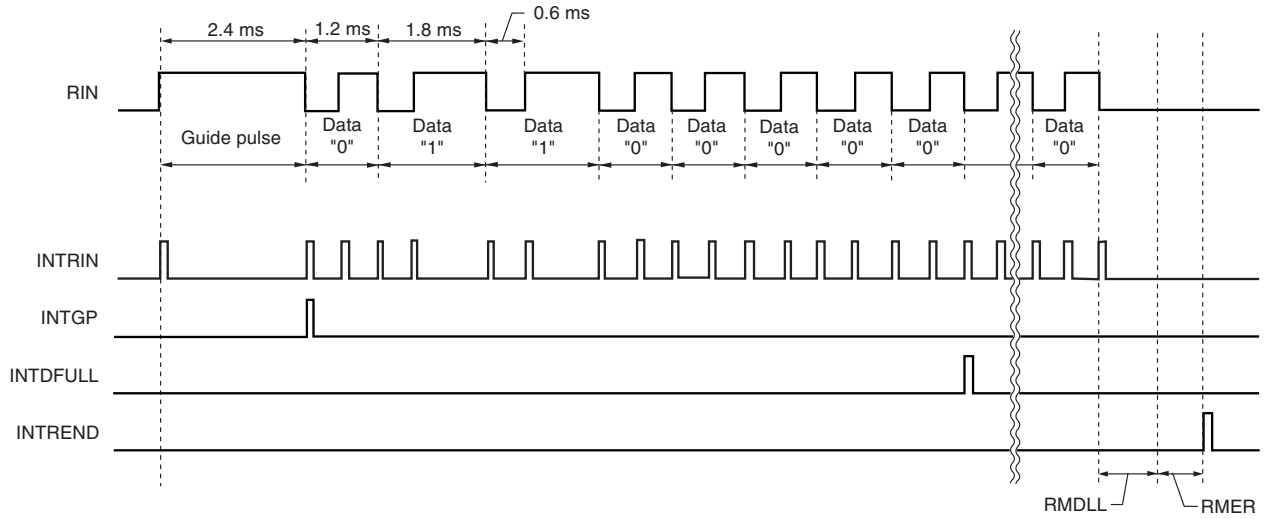
  Data 0 and data 1 are identified by the length of the data high-level width (2 types), with the data low-level width fixed.



18.5.1 Format of type A reception mode

Figure 18-22 shows the data format for type A.

Figure 18-22. Example of Type A Data Format



18.5.2 Operation flow of type A reception mode

Figure 18-23 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

2. When data has been set to all the bits of RMSR, the following processing is automatically performed.

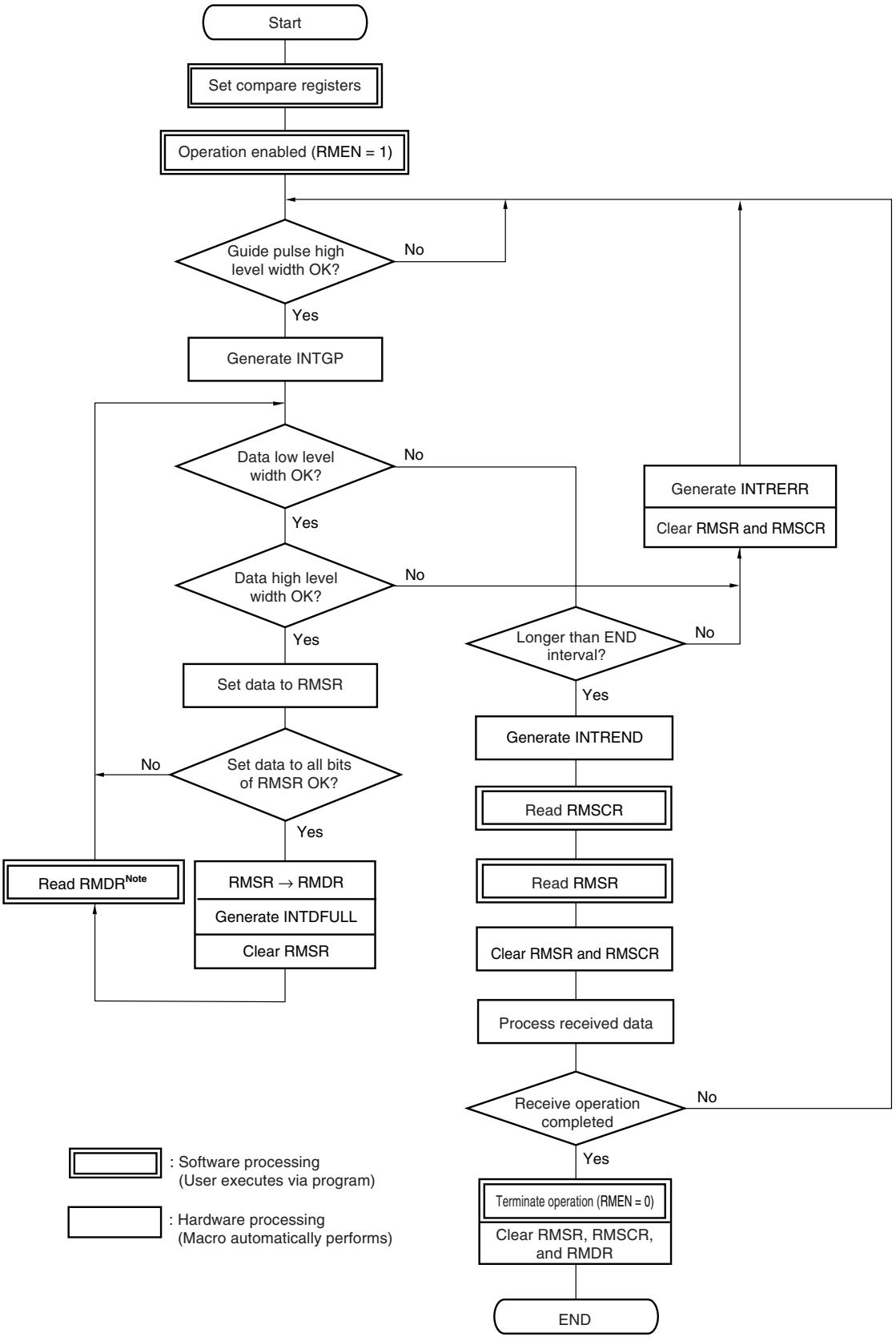
- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

Figure 18-23. Operation Flow of Type A Reception Mode

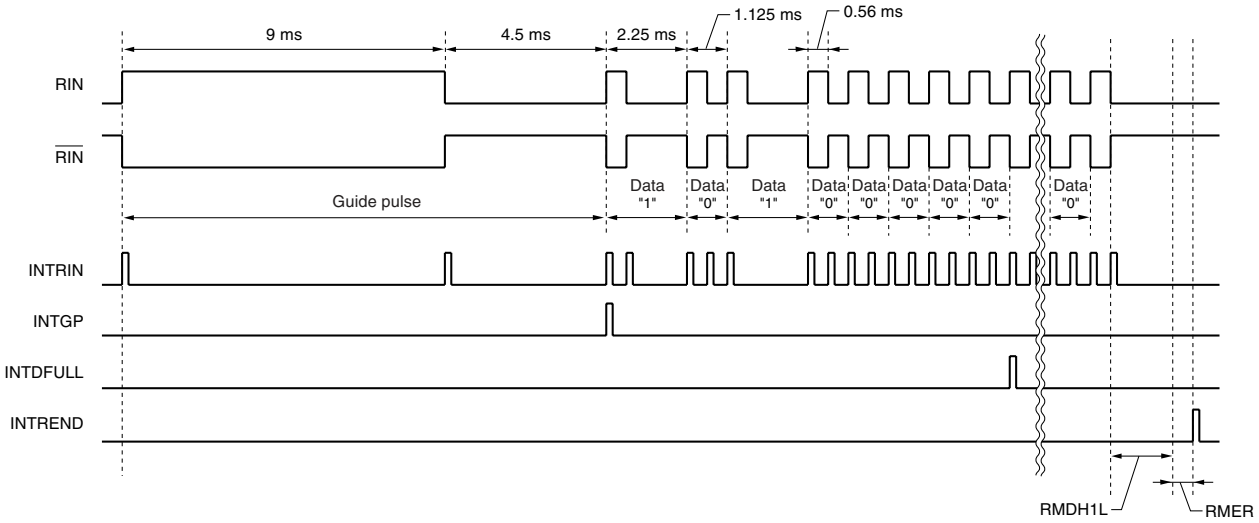


Note Read RMDR before data has been set to all the bits of RMSR.

18.5.3 Format of type B reception mode

Figure 18-24 shows the data format for type B.

Figure 18-24. Example of Type B Data Format



Remark $\overline{\text{RIN}}$ is the internally inverted signal of RIN.

18.5.4 Operation flow of type B reception mode

Figure 18-25 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

2. When data has been set to all the bits of RMSR, the following processing is automatically performed.

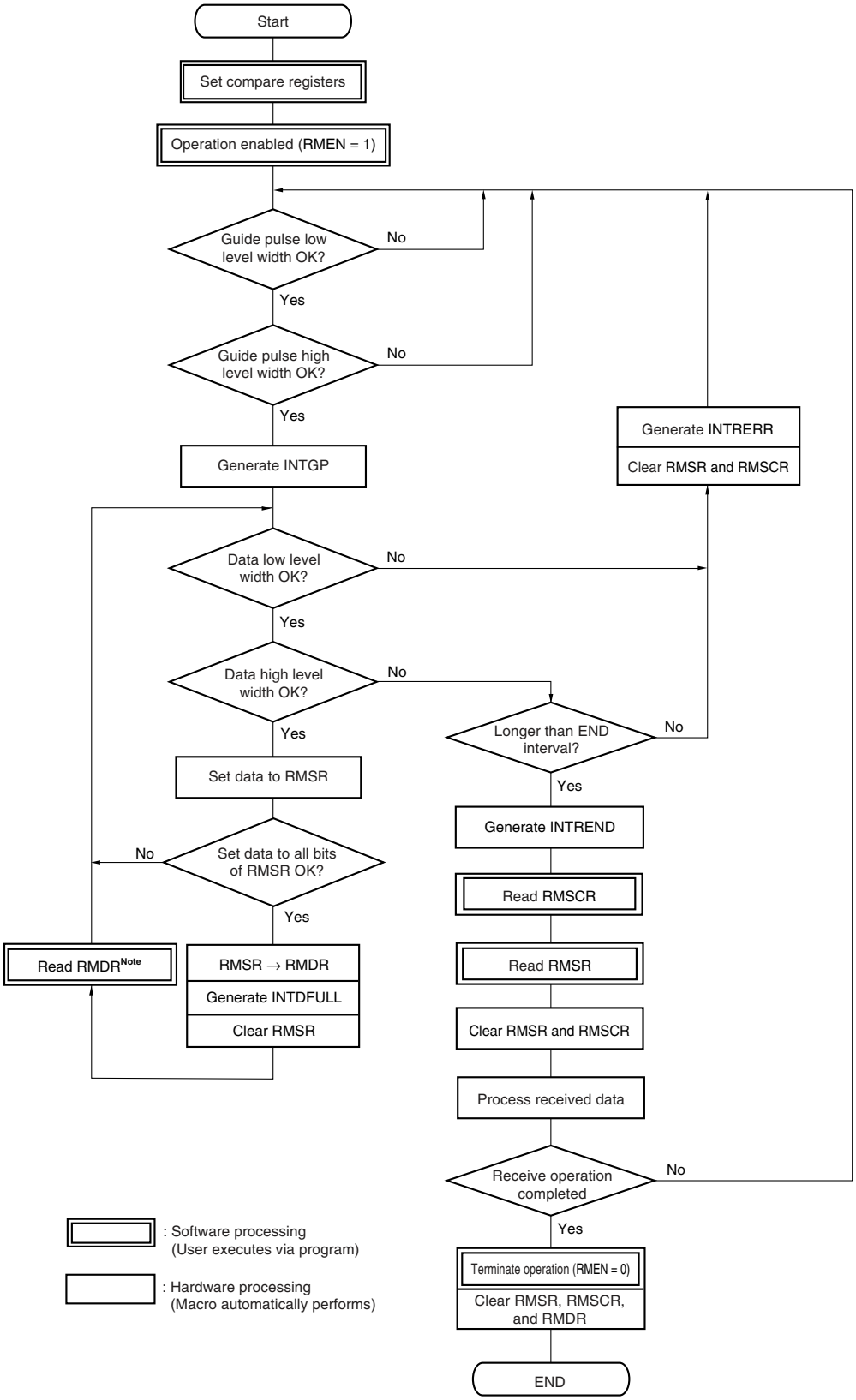
- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

Figure 18-25. Operation Flow of Type B Reception Mode

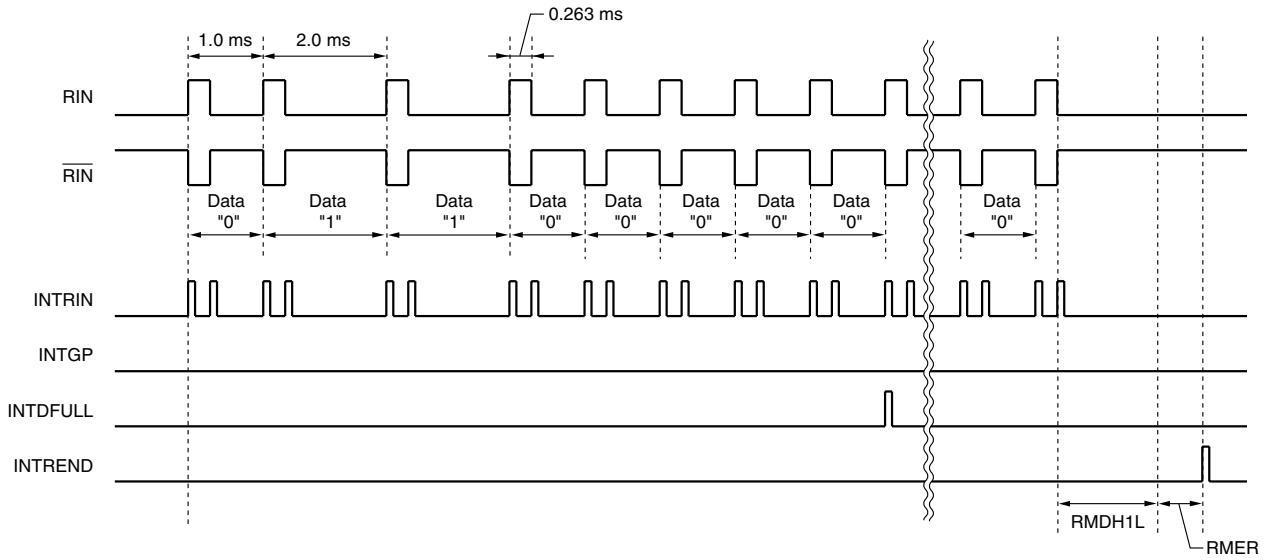


Note Read RMDR before data has been set to all the bits of RMSR.

18.5.5 Format of type C reception mode

Figure 18-26 shows the data format for type C.

Figure 18-26. Example of Type C Data Format



Remark $\overline{RIN}$ is the internally inverted signal of RIN.

18.5.6 Operation flow of type C reception mode

Figure 18-27 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

2. When data has been set to all the bits of RMSR, the following processing is automatically performed.

- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

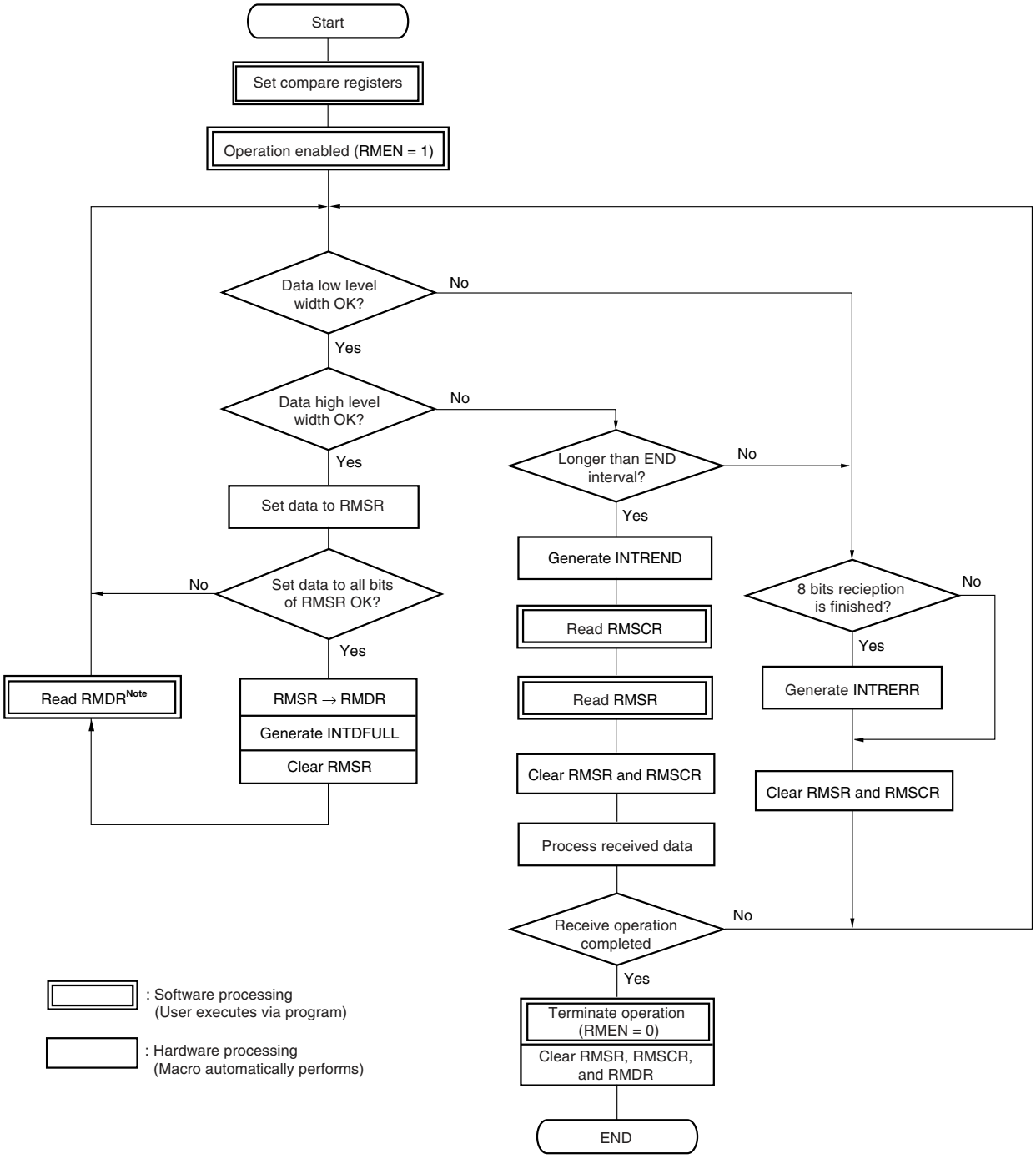
RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

5. In type C reception mode, if the conditions for receiving a data low-/high-level width are not met before the first INTDFULL interrupt is generated, INTRERR will not be generated. However, RMSR and RMSCR will be cleared.

Figure 18-27. Operation Flow of Type C Reception Mode

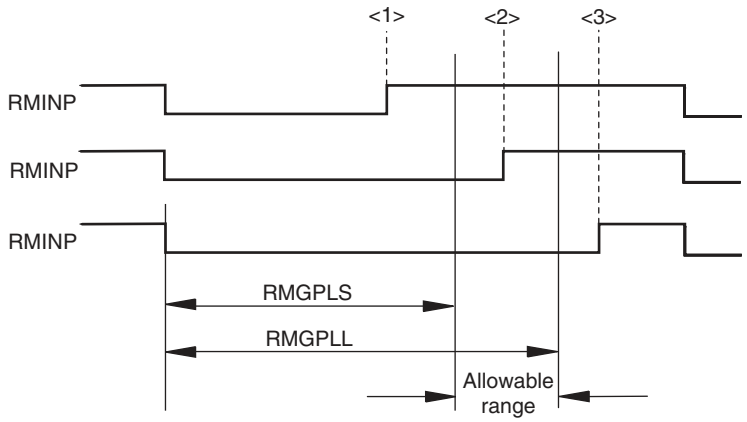


Note Read RMDR before data has been set to all the bits of RMSR.

18.5.7 Timing

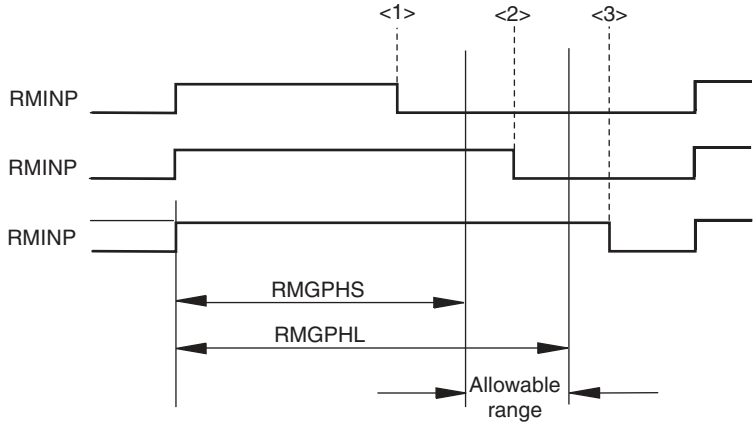
Operation varies depending on the positions of the RMINP input waveform below.

(1) Guide pulse low level width determination (Type B reception mode only)



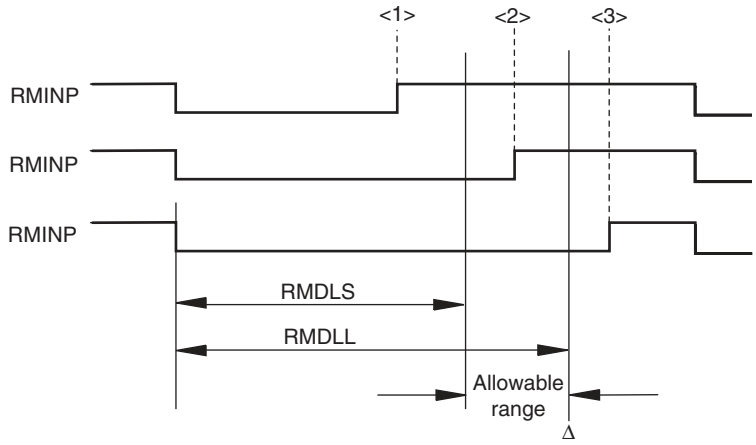
Condition		Determination	Subsequent operation
<1>	Counter < RMGPLS	NG (Short)	Detecting the guide pulse continues. Measuring guide pulse low-level width is started from the next falling edge.
<2>	RMGPLS ≤ counter < RMGPLL	OK (Within the range)	Guide pulse high-level width measurement is started.
<3>	RMGPLL ≤ counter	NG (Long)	Detecting the guide pulse continues. Measuring guide pulse low-level width is started from the next falling edge.

(2) Guide pulse high level width determination (Type A or Type B reception modes only)



Condition		Determination	Subsequent operation
<1>	Counter < RMGPHS	NG (Short)	(Type A reception mode) Detecting the guide pulse continues. Measuring guide pulse high-level width is started from the next rising edge. (Type B reception modes) Detecting the guide pulse continues and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from this falling edge.
<2>	RMGPHS ≤ counter < RMGPHL	OK (Within the range)	INTGP is generated. Data measurement is started.
<3>	RMGPHL ≤ counter	NG (Long)	(Type A reception mode) Detecting the guide pulse continues. Measuring guide pulse high-level width is started from the next rising edge. (Type B reception modes) Detecting the guide pulse continues and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from this falling edge.

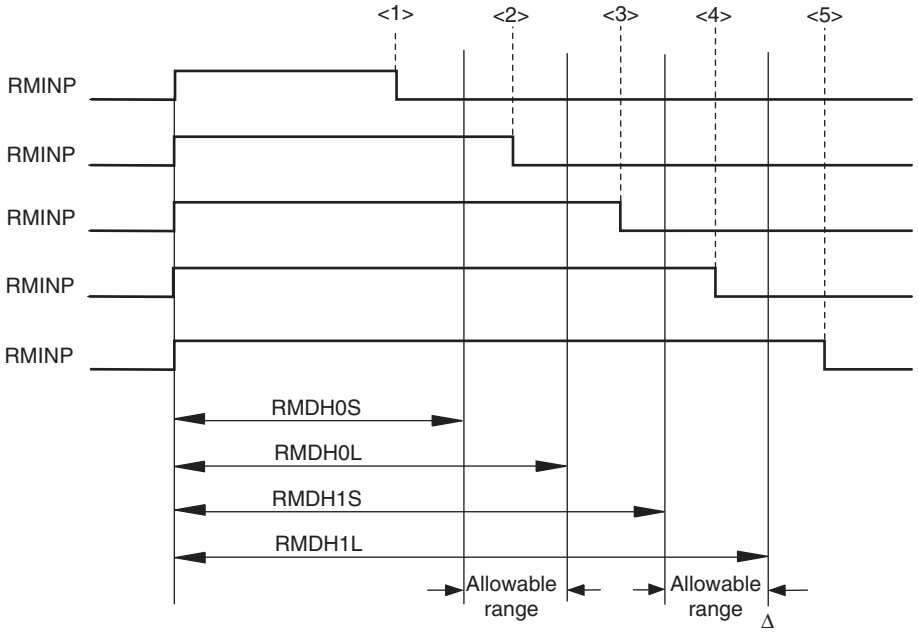
(3) Data low level width determination



Condition		Determination	Subsequent operation
<1>	Counter < RMDLS	NG (Short)	Error interrupt INTRERRn is generated [Note]. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse low-level width is started from this rising edge. (Type B reception mode) Detecting the guide pulse starts and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from the next falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from the next falling edge.
<2>	RMDLS ≤ counter < RMDLL	OK (Within the range)	Measuring data high-level width is started.
<3>	RMDLL ≤ counter	OK (Type A) / NG (Type B, Type C)	(Type A reception mode) Measuring the end width is started from the Δ point. (Type B reception mode) Error interrupt INTRERR is generated at the Δ point. Measuring guide pulse low-level width is started from the next falling edge. (Type C reception modes) Error interrupt INTRERRn is generated at the Δ point [Note]. Detecting the data continues. Measuring data low-level width is started from the next falling edge.

Note In type C reception mode, before the first INTDFULL interrupt is generated, INTRERR will not be generated. However, RMSR and RMSCR will be cleared.

(4) Data high level width determination

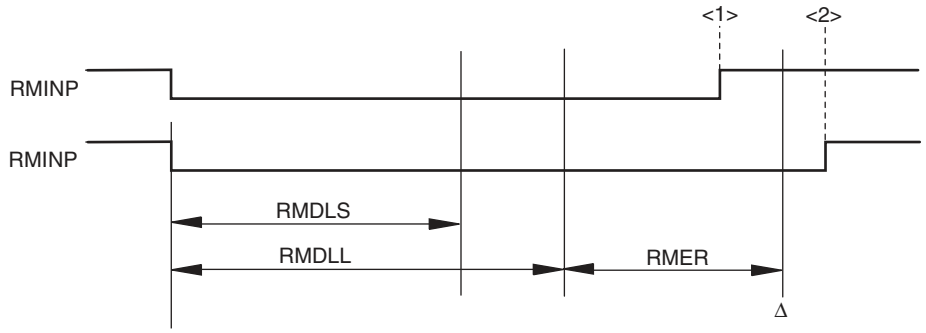


Condition		Determination	Subsequent operation
<1>	Counter < RMDH0S	NG (Short)	Error interrupt INTRERRn is generated Note. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse high-level width is started at the next rising edge. (Type B reception mode) Detecting the guide pulse starts. Measuring guide pulse low-level width is started from this falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from this falling edge.
<2>	RMDH0S ≤ counter < RMDH0L	OK (Within the range)	Data 0 is received. Detecting the data continues. Measuring data low-level width is started from this falling edge.
<3>	RMDH0L ≤ counter < RMDH1S	NG (Outside of the range)	Error interrupt INTRERR is generated Note. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse high-level width is started at the next rising edge. (Type B reception mode) Detecting the guide pulse starts. Measuring guide pulse low-level width is started from this falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from this falling edge.
<4>	RMDH1S ≤ counter < RMDH1L	OK (Within the range)	Data 1 is received. Detecting the data continues. Measuring the data low-level width is started from this falling edge.
<5>	RMDH1L ≤ counter	NG (Type A) / OK (Type B, Type C)	(Type A reception mode) Error interrupt INTRERR is generated at the Δ point. Measuring the guide pulse is started at the next rising edge. (Type B, Type C reception modes) Measuring the end width is started from the Δ point.

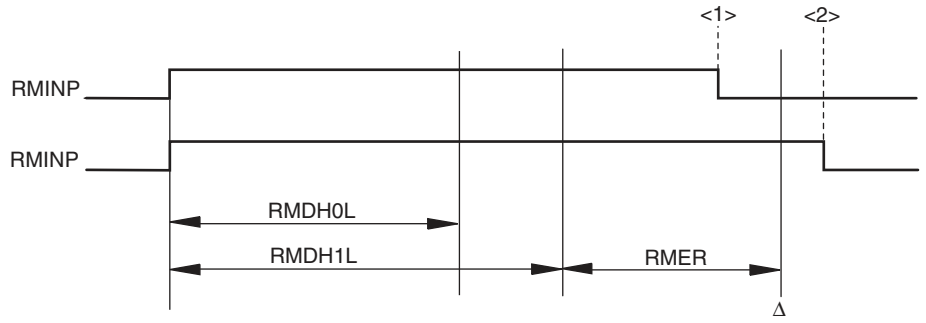
Note In type C reception mode, before the first INTDFULL interrupt is generated, INTRERR will not be generated. However, RMSR and RMSCR will be cleared.

(5) End width determination

(a) Type A reception mode



(b) Type B, Type C reception modes



Condition		Determination	Subsequent operation
<1>	Counter < RMER	NG (Short)	Error interrupt INTRERRn is generated [Note]. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse high-level width is started from this rising edge. (Type B reception mode) Detecting the guide pulse starts and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from the this falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from this falling edge.
<2>	RMER ≤ counter	OK (Long)	INTREND is generated at the Δ point [Note]. Reception via stops until RMSR is read.

Note In type C reception mode, before the first INTDFULL interrupt is generated, INTRERR will not be generated. However, RMSR and RMSCR will be cleared.

18.5.8 Compare register setting

This remote controller receiver has the following compare registers.

- Remote controller receive GPLS compare register (RMGPLS)
- Remote controller receive GPLL compare register (RMGPLL)
- Remote controller receive GPHS compare register (RMGPHS)
- Remote controller receive GPHL compare register (RMGPHL)
- Remote controller receive DLS compare register (RMDLS)
- Remote controller receive DLL compare register (RMDLL)
- Remote controller receive DH0S compare register (RMDH0S)
- Remote controller receive DH0L compare register (RMDH0L)
- Remote controller receive DH1S compare register (RMDH1S)
- Remote controller receive DH1L compare register (RMDH1L)
- Remote controller receive end width select register (RMER)

Use formulas (1) to (3) below to set the value of each compare register.

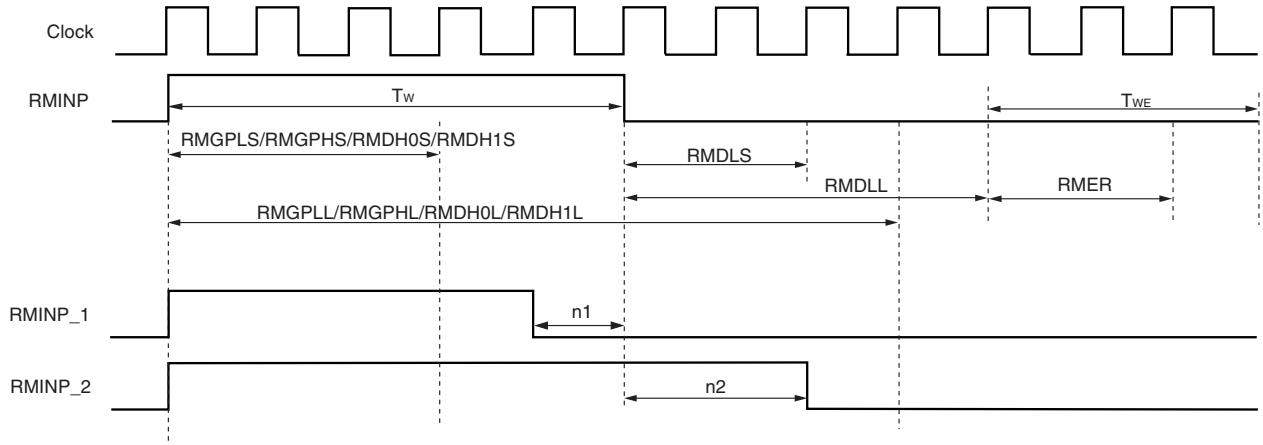
Making allowances for tolerance enables a normal reception operation, even if the RMINP input waveform is RMINP_1 or RMINP_2 shown in Figures 18-28 due to the effect of noise.

Cautions 1. Always set each compare register while remote controller reception is disabled (RMEN = 0).

2. Set the set values so that they satisfy all the following four conditions.

- **RMGPLS < RMGPLL**
- **RMGPHS < RMGPHL**
- **RMDLS < RMDLL**
- **RMDH0S < RMDH0L ≤ RMDH1S < RMDH1L**

Figure 18-28. Setting Example (Where n1 = 1, n2 = 2)



(1) Formula for RMGPLS, RMGPHS, RMDLS, RMDH0S, and RMDH1S

$$\left(\frac{T_W \times (1 - a/100)}{1/f_{REMPRS}} \right)_{INT} - 2 - n1$$

(2) Formula for RMGPLL, RMGPHL, RMDLL, RMDH0L, and RMDH1L

$$\left(\frac{T_W \times (1 + a/100)}{1/f_{REMPRS}} \right)_{INT} + 1 + n2$$

(3) Formula for RMER

$$\left(\frac{T_{WE} \times (1 - a/100)}{1/f_{REMPRS}} \right)_{INT} - 1$$

T_W: Width of RMINP waveform

$1/f_{REMPRS}$: Width of internal operation clock cycle after division control by RMPRS

a: Tolerance (%)

$[\ ]_{INT}$: Round down the fractional portion of the value produced by the formula in the brackets.

n1, n2: Variables of waveform change caused by noise[Note 1]

T_{WE}: End width of RMINP[Note 2]

Notes 1. Set the values of n1 and n2 as required to meet the user's system specification.

2. This end width is counted after RMDLL.

The low-level width actually required after the last data has been received is as follows:

(RMDLL + 1 + RMER + 1)× (width of internal operation clock cycle after division control by RMPRS)

18.5.9 Error interrupt generation timing

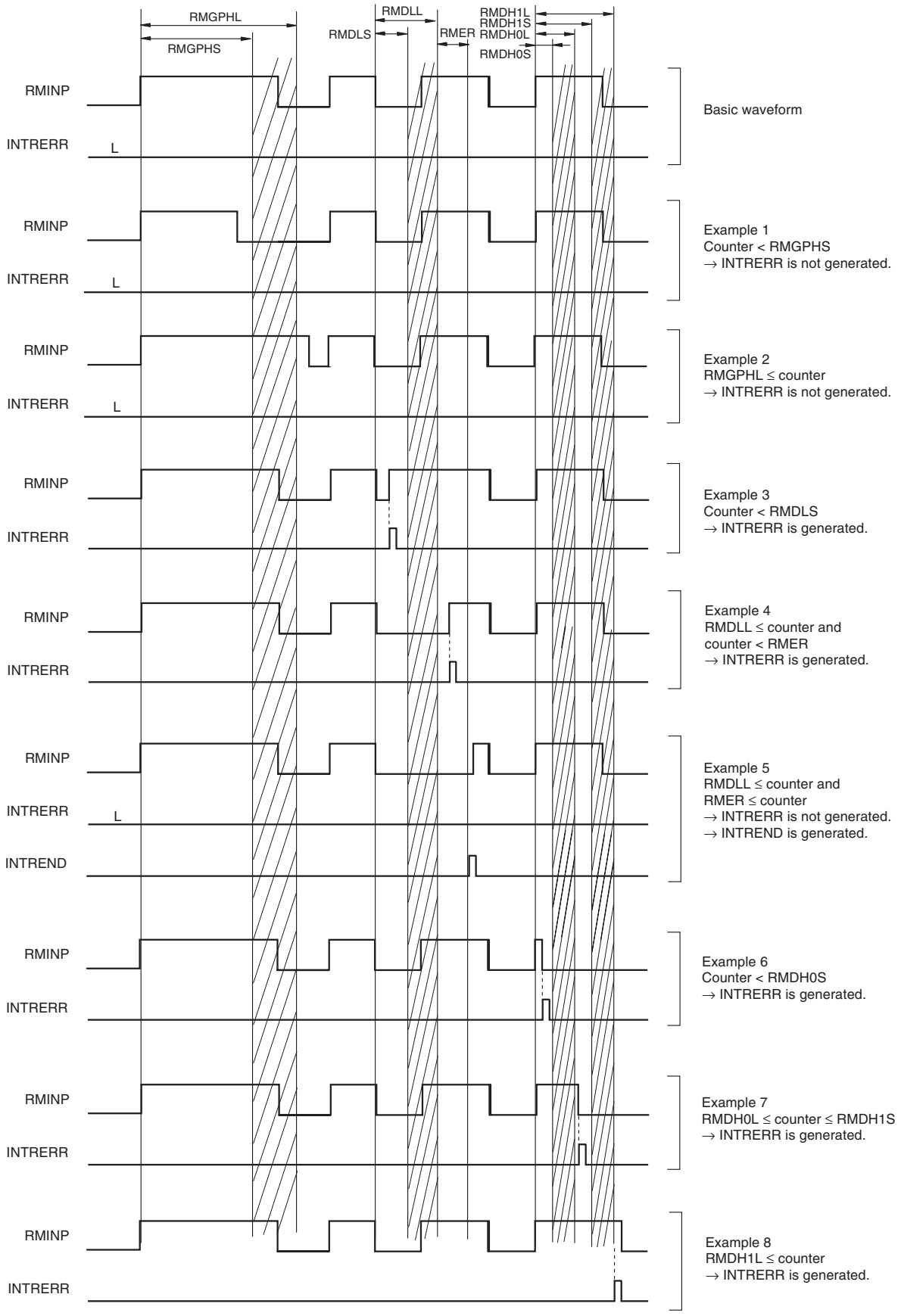
(1) Type A reception mode

After the guide pulse has been detected normally, the INTRERR signal is generated under any of the following conditions.

- Counter < RMDLS at the rising edge of RMINP
- RMDLL ≤ counter and counter after RMDLL < RMER at the rising edge of RMINP
- Counter < RMDH0S at the falling edge of RMINP
- RMDH0L ≤ counter < RMDH1S at the falling edge of RMINP
- RMDH1L ≤ counter while RMINP is at high level

The INTRERR signal is not generated until the guide pulse is detected.
Once the INTRERR signal has been generated, it will not be generated again until the next guide pulse is detected.
The generation timing of the INTRERR signal is shown in **Figure 18-29**.

Figure 18-29. Generation Timing of INTRERR Signal (Type A reception mode)



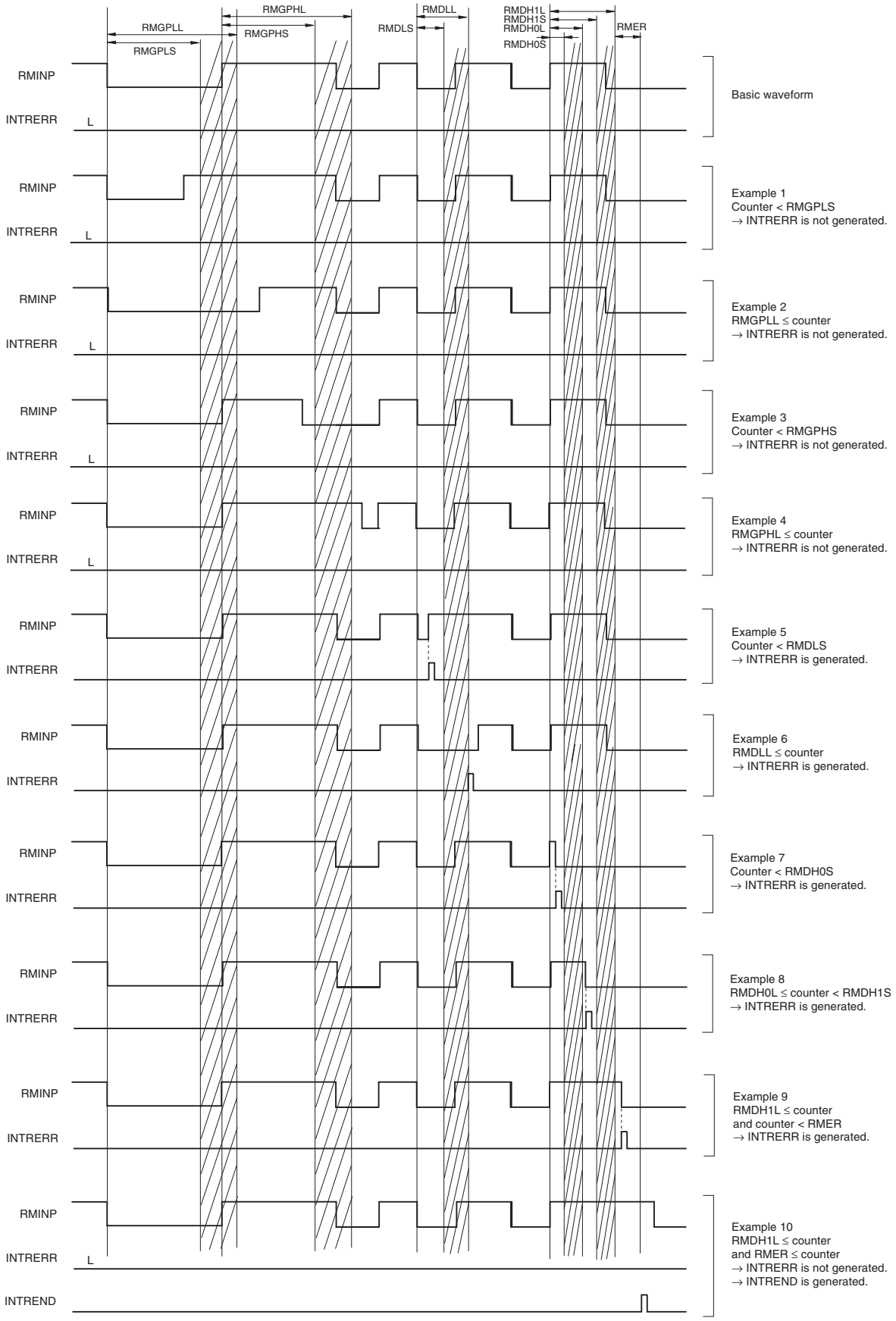
(2) Type B reception mode

After the guide pulse has been detected normally, the INTRERR signal is generated under any of the following conditions.

- Counter < RMDLS at the rising edge of RMINP
- RMDLL ≤ counter while RMINP is at low level
- Counter < RMDH0S at the falling edge of RMINP
- RMDH0L ≤ counter < RMDH1S at the falling edge of RMINP
- RMDH1L ≤ counter and counter after RMDH1L < RMER at the falling edge of RMINP

The INTRERR signal is not generated until the guide pulse is detected.
Once the INTRERR signal has been generated, it will not be generated again until the next guide pulse is detected.
The generation timing of the INTRERR signal is shown in **Figure 18-30**.

Figure 18-30. Generation Timing of INTRERRn Signal (Type B reception mode)



(3) Type C reception mode

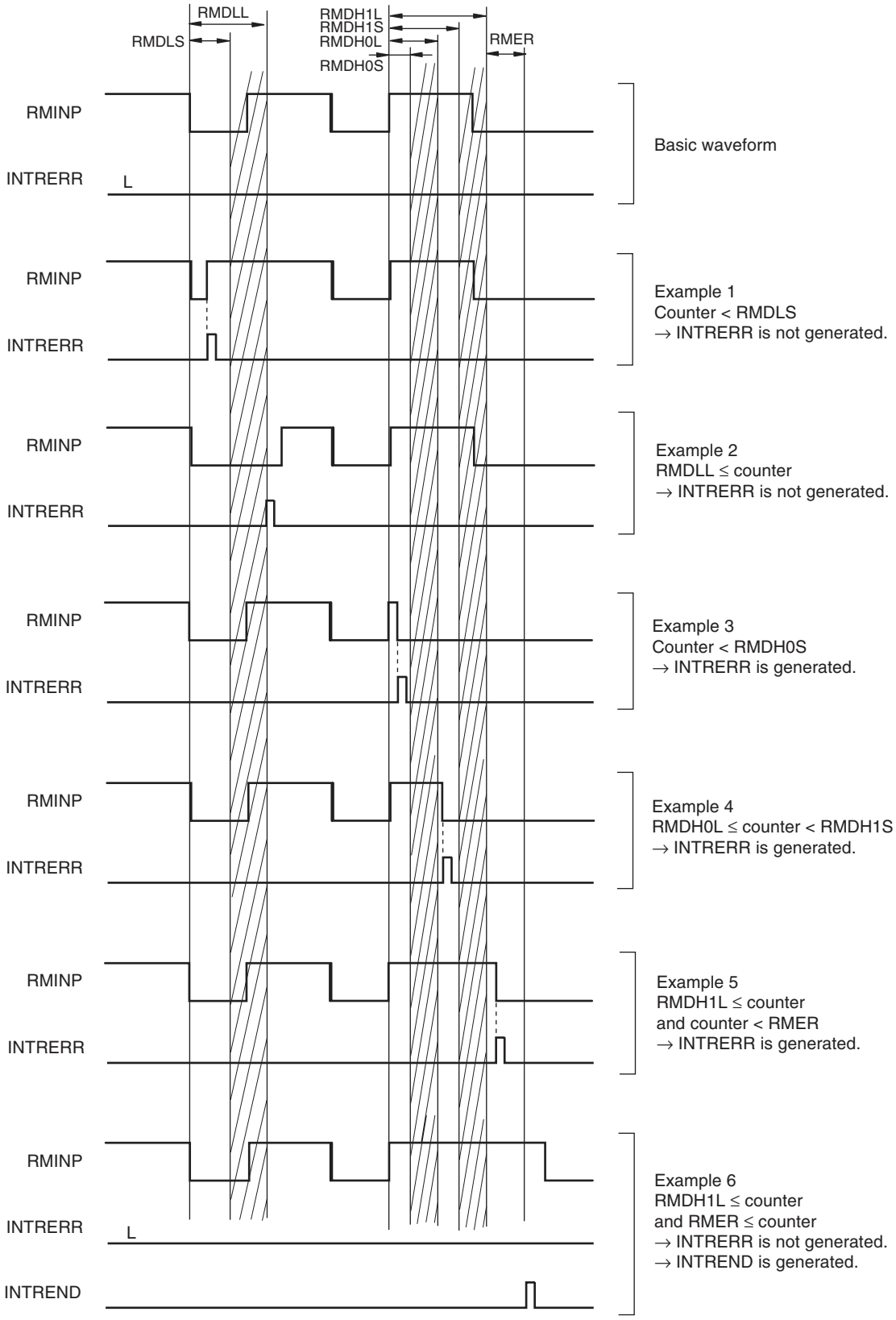
The INTRERR signal is generated under any of the following conditions.

- Counter < RMDLS at the rising edge of RMINP
- RMDLL ≤ counter while RMINP is at low level
- Counter < RMDH0S at the falling edge of RMINP
- RMDH0L ≤ counter < RMDH1S at the falling edge of RMINP
- RMDH1L ≤ counter and counter after RMDH1L < RMER at the falling edge of RMINP

However, before the first INTDFULL interrupt is generated, INTRERR signal will not be generated.

The generation timing of the INTRERR signal is shown in **Figure 18-31**.

Figure 18-31. Generation Timing of INTRERRn Signal (Type C reception mode)



CHAPTER 19 MULTIPLIER/DIVIDER

Caution If multiplier/divider is used, INTTM012 of TM02 cannot be used.

19.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 16 bits = 32 bits, 16-bit remainder (division)

19.2 Configuration of Multiplier/Divider

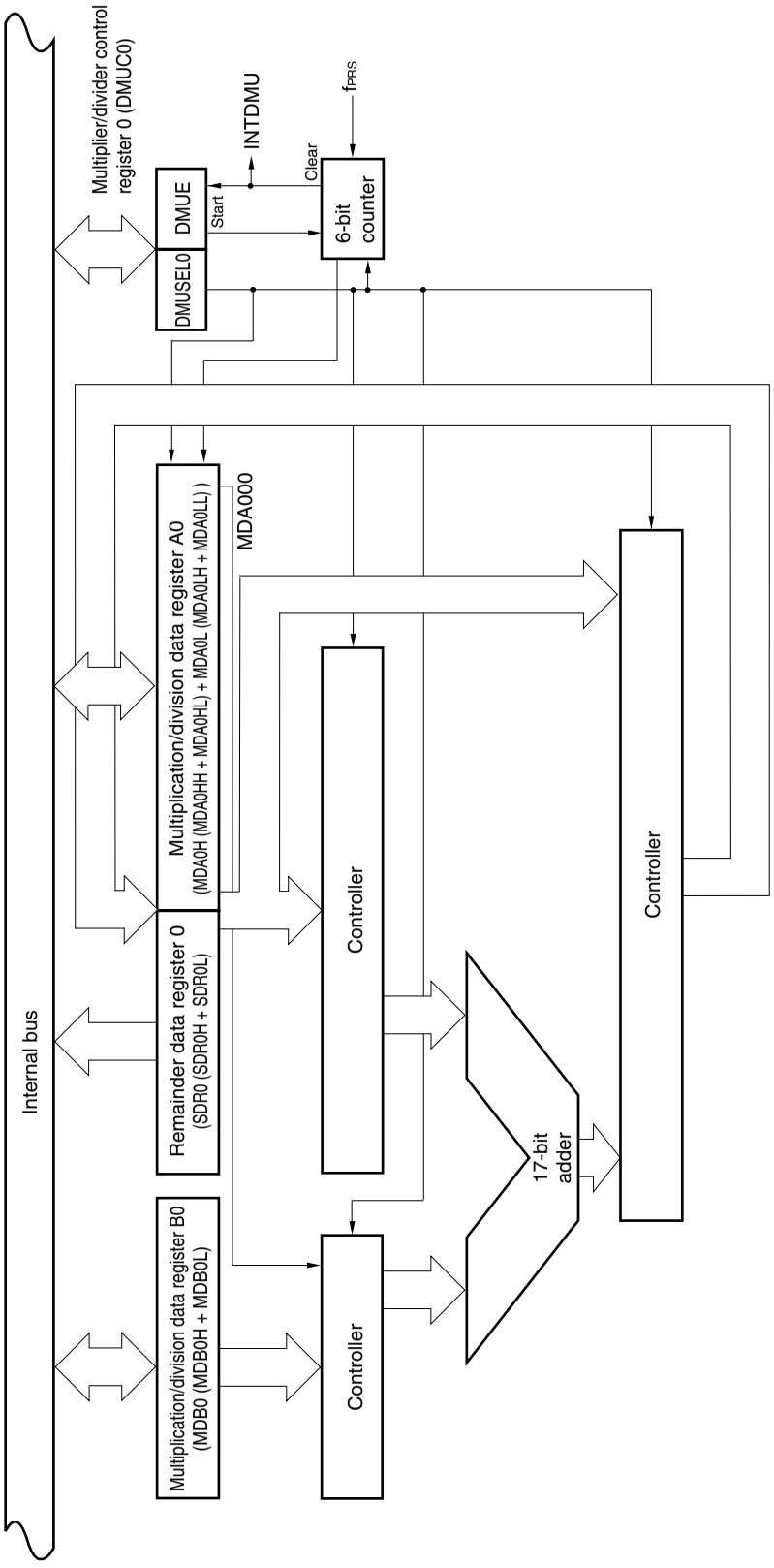
The multiplier/divider includes the following hardware.

Table 19-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Remainder data register 0 (SDR0) Multiplication/division data registers A0 (MDA0H, MDA0L) Multiplication/division data registers B0 (MDB0)
Control register	Multiplier/divider control register 0 (DMUC0)

Figure 19-1 shows the block diagram of the multiplier/divider.

Figure 19-1. Block Diagram of Multiplier/Divider



(1) Remainder data register 0 (SDR0)

SDR0 is a 16-bit register that stores a remainder. This register stores 0 in the multiplication mode and the remainder of an operation result in the division mode.

SDR0 can be read by an 8-bit or 16-bit memory manipulation instruction.

Reset signal generation clears SDR0 to 0000H.

Figure 19-2. Format of Remainder Data Register 0 (SDR0)

Address: FF60H, FF61H After reset: 0000H R

Symbol	FF61H (SDR0H)								FF60H (SDR0L)							
SDR0	SDR 015	SDR 014	SDR 013	SDR 012	SDR 011	SDR 010	SDR 009	SDR 008	SDR 007	SDR 006	SDR 005	SDR 004	SDR 003	SDR 002	SDR 001	SDR 000

Cautions 1. The value read from SDR0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1) is not guaranteed.

2. SDR0 is reset when the operation is started (when DMUE is set to 1).

(2) Multiplication/division data register A0 (MDA0H, MDA0L)

MDA0 is a 32-bit register that sets a 16-bit multiplier A in the multiplication mode and a 32-bit dividend in the division mode, and stores the 32-bit result of the operation (higher 16 bits: MDA0H, lower 16 bits: MDA0L).

Figure 19-3. Format of Multiplication/Division Data Register A0 (MDA0H, MDA0L)

Address: FF62H, FF63H, FF64H, FF65H After reset: 0000H, 0000H R/W

Symbol	FF65H (MDA0HH)								FF64H (MDA0HL)							
MDA0H	MDA 031	MDA 030	MDA 029	MDA 028	MDA 027	MDA 026	MDA 025	MDA 024	MDA 023	MDA 022	MDA 021	MDA 020	MDA 019	MDA 018	MDA 017	MDA 016

Symbol	FF63H (MDA0LH)								FF62H (MDA0LL)							
MDA0L	MDA 015	MDA 014	MDA 013	MDA 012	MDA 011	MDA 010	MDA 009	MDA 008	MDA 007	MDA 006	MDA 005	MDA 004	MDA 003	MDA 002	MDA 001	MDA 000

Cautions 1. MDA0H is cleared to 0 when an operation is started in the multiplication mode (when multiplier/divider control register 0 (DMUC0) is set to 81H).

2. Do not change the value of MDA0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

3. The value read from MDA0 during operation processing (while DMUE is 1) is not guaranteed.

The functions of MDA0 when an operation is executed are shown in the table below.

Table 19-2. Functions of MDA0 During Operation Execution

DMUSEL0	Operation Mode	Setting	Operation Result
0	Division mode	Dividend	Division result (quotient)
1	Multiplication mode	Higher 16 bits: 0, Lower 16 bits: Multiplier A	Multiplication result (product)

Remark DMUSEL0: Bit 0 of multiplier/divider control register 0 (DMUC0)

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>

  MDA0 (bits 15 to 0) × MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0)

- Register configuration during division

  <Dividend> <Divisor> <Quotient> <Remainder>

  MDA0 (bits 31 to 0) ÷ MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0) … SDR0 (bits 15 to 0)

MDA0 fetches the calculation result as soon as the clock is input, when bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is set to 1.
MDA0H and MDA0L can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDA0H and MDA0L to 0000H.

(3) Multiplication/division data register B0 (MDB0)

MDB0 is a register that stores a 16-bit multiplier B in the multiplication mode and a 16-bit divisor in the division mode.
MDB0 can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDB0 to 0000H.

Figure 19-4. Format of Multiplication/Division Data Register B0 (MDB0)

Address: FF66H, FF67H After reset: 0000H R/W

Symbol	FF67H (MDB0H)								FF66H (MDB0L)							
MDB0	MDB 015	MDB 014	MDB 013	MDB 012	MDB 011	MDB 010	MDB 009	MDB 008	MDB 007	MDB 006	MDB 005	MDB 004	MDB 003	MDB 002	MDB 001	MDB 000

Cautions 1. Do not change the value of MDB0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

2. Do not clear MDB0 to 0000H in the division mode. If set, undefined operation results are stored in MDA0 and SDR0.

19.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by multiplier/divider control register 0 (DMUC0).

(1) Multiplier/divider control register 0 (DMUC0)

DMUC0 is an 8-bit register that controls the operation of the multiplier/divider.

DMUC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears DMUC0 to 00H.

Figure 19-5. Format of Multiplier/Divider Control Register 0 (DMUC0)

Address: FF68H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0

DMUE[Note]	Operation start/stop
0	Stops operation
1	Starts operation

DMUSEL0	Operation mode (multiplication/division) selection
0	Division mode
1	Multiplication mode

Note When DMUE is set to 1, the operation is started. DMUE is automatically cleared to 0 after the operation is complete.

Cautions 1. If DMUE is cleared to 0 during operation processing (when DMUE is 1), the operation result is not guaranteed. If the operation is completed while the clearing instruction is being executed, the operation result is guaranteed, provided that the interrupt flag is set.

2. Do not change the value of DMUSEL0 during operation processing (while DMUE is 1). If it is changed, undefined operation results are stored in multiplication/division data register A0 (MDA0) and remainder data register 0 (SDR0).

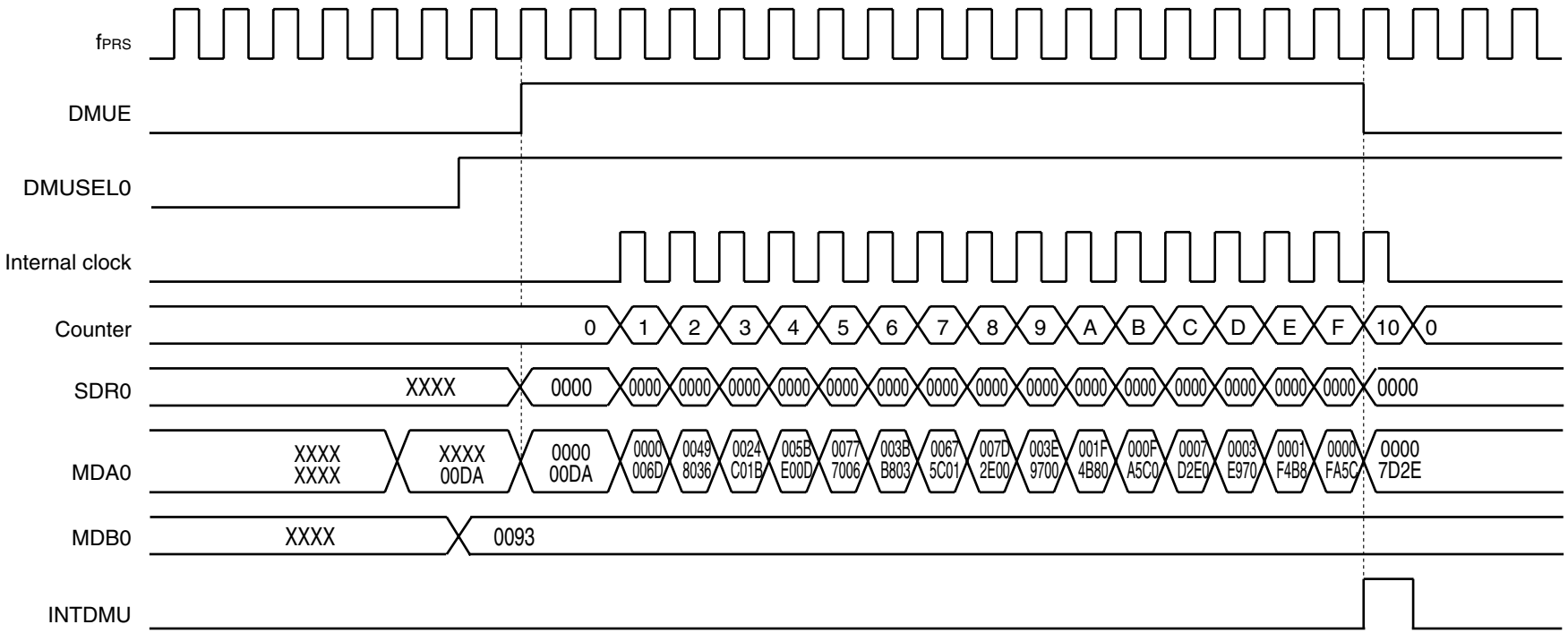
3. If DMUE is cleared to 0 during operation processing (while DMUE is 1), the operation processing is stopped. To execute the operation again, set multiplication/division data register A0 (MDA0), multiplication/division data register B0 (MDB0), and multiplier/divider control register 0 (DMUC0), and start the operation (by setting DMUE to 1).

19.4 Operations of Multiplier/Divider

19.4.1 Multiplication operation

- Initial setting
  1. Set operation data to multiplication/division data register A0L (MDA0L) and multiplication/division data register B0 (MDB0).
  2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 1. Operation will start.
- During operation
  3. The operation will be completed when 16 peripheral hardware clocks (f_{PRS}) have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers during operation, and therefore the read values of these registers are not guaranteed).
- End of operation
  4. The operation result data is stored in the MDA0L and MDA0H registers.
  5. DMUE is cleared to 0 (end of operation).
  6. After the operation, an interrupt request signal (INTDMU) is generated.
- Next operation
  7. To execute multiplication next, start from the initial setting in **19.4.1 Multiplication operation**.
  8. To execute division next, start from the initial setting in **19.4.2 Division operation**.

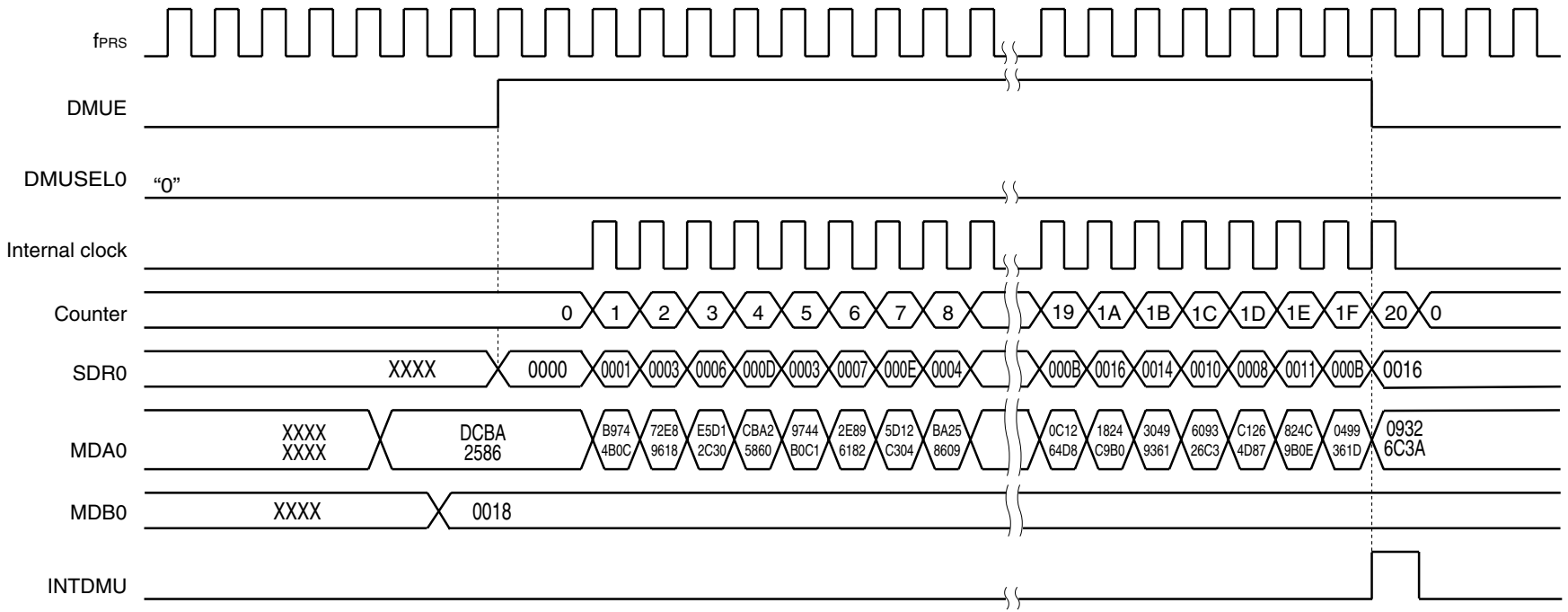
Figure 19-6. Timing Chart of Multiplication Operation (00DAH × 0093H)



19.4.2 Division operation

- Initial setting
  1. Set operation data to multiplication/division data register A0 (MDA0L and MDA0H) and multiplication/division data register B0 (MDB0).
  2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 0 and 1, respectively. Operation will start.
- During operation
  3. The operation will be completed when 32 peripheral hardware clocks (f_{PRS}) have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers and remainder data register 0 (SDR0) during operation, and therefore the read values of these registers are not guaranteed).
- End of operation
  4. The result data is stored in the MDA0L, MDA0H, and SDR0 registers.
  5. DMUE is cleared to 0 (end of operation).
  6. After the operation, an interrupt request signal (INTDMU) is generated.
- Next operation
  7. To execute multiplication next, start from the initial setting in **19.4.1 Multiplication operation**.
  8. To execute division next, start from the initial setting in **19.4.2 Division operation**.

Figure 19-7. Timing Chart of Division Operation (DCBA2586H ÷ 0018H)



CHAPTER 20 INTERRUPT FUNCTIONS

20.1 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 a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L, PR1H).

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 20-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

External: 8, internal: 27

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

20.2 Interrupt Sources and Configuration

The 78K0/Kx2-C has a total of 32 interrupt sources, including nonmaskable interrupts, maskable interrupts, and software interrupts. In addition, they also have up to four reset sources (see **Table 20-1**).

Table 20-1. 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			
Nonmas-kable	–	INTCK2	An address break occurs before OCD is executed.	Internal	0002H	(A)
		INTNMI	An OCD break other than the above occurs.			
Maskable	0	INTLVI	Low-voltage detection[Note 3]	Internal	0004H	(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4/ INTDA	Pin input edge detection/ End of CEC 1-byte communication	External/ Internal	000EH	(C)/(B)
	6	INTP5/ INTCE/ INTERR	Pin input edge detection/ End of CEC communication/ CEC communication error occurrence	External/ Internal	0010H	(C)/(B)
	7	INTP6/ INTRIN	Pin input edge detection/ Remote controller reception edge detection	External/ Internal	0012H	(C)/(B)
	8	INTSR60/ INTSRE60	End of UART60 reception or reception error generation/ UART60 reception error generation	Internal	0014H	(B)
	9	INTST60	End of UART60 transmission		0016H	
	10	INTCSI10/ INTST0	End of CSI10 communication/ End of UART0 transmission		0018H	
	11	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)		001AH	
	12	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		001CH	
	13	INTTM50	Match between TM50 and CR50 (when compare register is specified)		001EH	
	14	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		0020H	
	15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	
	16	INTAD	End of A/D conversion		0024H	
	17	INTSR0/ INTSRE0	End of UART0 reception or reception error generation/ UART0 reception error generation		0026H	

Notes 1. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 28 indicates the lowest priority.

2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 20-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

Table 20-1. 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	18	INTRTC/ INTRTCI	Fixed-cycle signal of real-time counter/alarm match detection/ Interval signal detection of real-time counter	Internal	0028H	(B)
	19	INTTM51 [Note 3]	Match between TM51 and CR51 (when compare register is specified)		002AH	
	20	INTIICA0	End of IICA00 communication		002CH	
	21	INTIICA1	End of IICA01 communication		002EH	
	22	INTKR/ INTIICA2	Key interrupt detection/ End of IICA02 communication	External/ Internal	0030H	(C)/(B)
	23	INTCSI11	End of CSI11 transfer end or buffer empty interrupt	Internal	0032H	(B)
	24	INTTM001/ INTSR61[Note 4]/ INTSRE61[Note 4]	Match between TM01 and CR001 (when compare register is specified), TI011 pin valid edge detection (when capture register is specified)[Note 5]/ End of UART61 reception or reception error generation[Note 4]/ UART61 reception error generation[Note 4]		0034H	
	25	INTTM011/ INTST61[Note 4]	Match between TM01 and CR011 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)[Note 5]/ End of UART61 transmission[Note 4]		0036H	
	26	INTTM002	Match between TM02 and CR002 (when compare register is specified), TI012 pin valid edge detection (when capture register is specified)[Note 5]		0038H	
	27	INTTM012/ INTDMU	Match between TM02 and CR012 (when compare register is specified), TI002 pin valid edge detection (when capture register is specified)[Note 5]/ multiply/divide operation		003AH	
	28	INTRERR/ INTGP/ INTREND/ INTDFULL	Remote controller reception error occurrence Remote controller guide pulse detection Remote controller data reception completion Read request for remote controller 8-bit shift data		003CH	

Notes 1. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 28 indicates the lowest priority.

2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 20-1.

3. When 8-bit timer/event counter 51 and 8-bit timer H1 are used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (see **Figure 8-13 Transfer Timing**).

4. 78K0/KE2-C only.

5. Only compare registers can be specified for TM01 and TM02 of the 78K0/KC2-C.

Table 20-1. Interrupt Source List (3/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Software	–	BRK	BRK instruction execution	–	003EH	(E)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on clear			
		LVI	Low-voltage detection[Note 3]			
		WDT	WDT overflow			

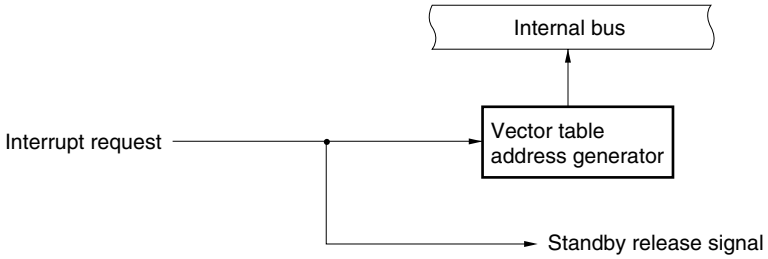
Notes 1. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 28 indicates the lowest priority.

2. Basic configuration types (A) to (E) correspond to (A) to (E) in Figure 20-1.

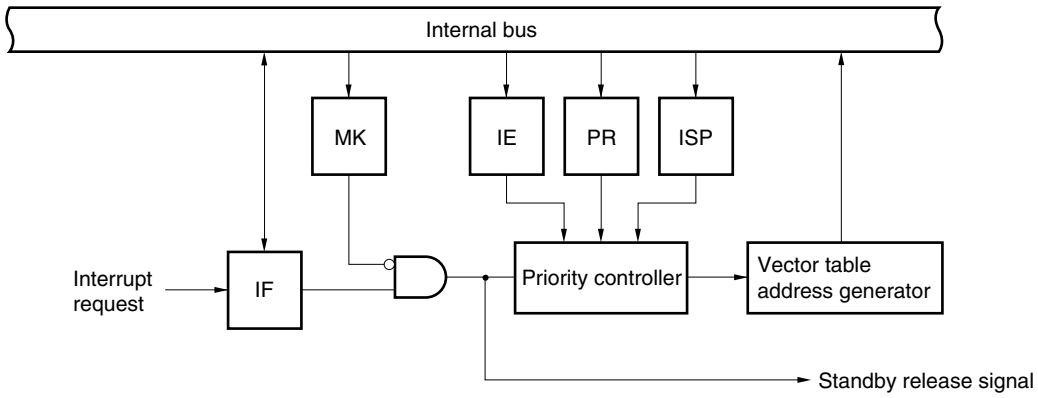
3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

Figure 20-1. Basic Configuration of Interrupt Function (1/2)

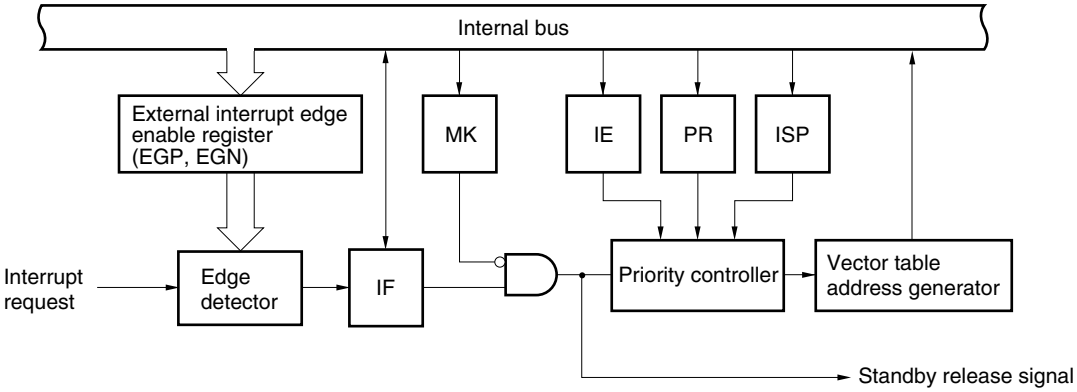
(A) Internal nonmaskable interrupt



(B) Internal maskable interrupt



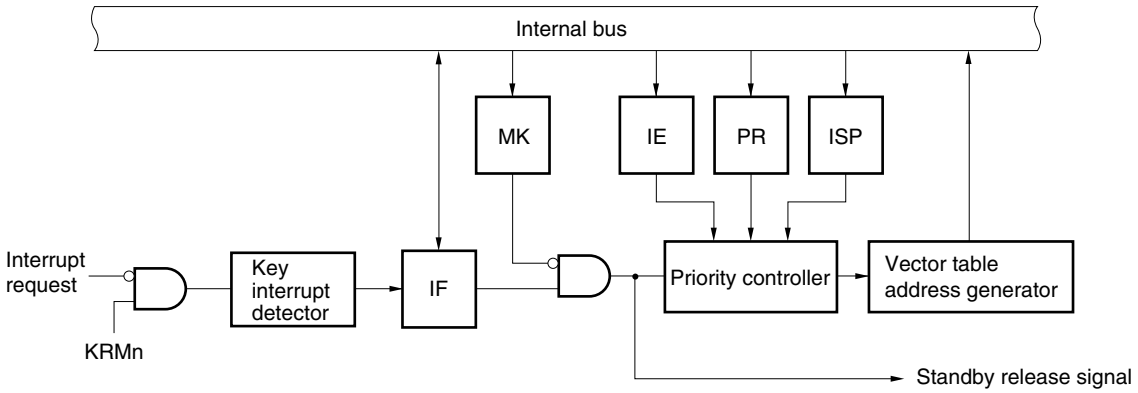
(C) External maskable interrupt (INTP0 to INTP6)



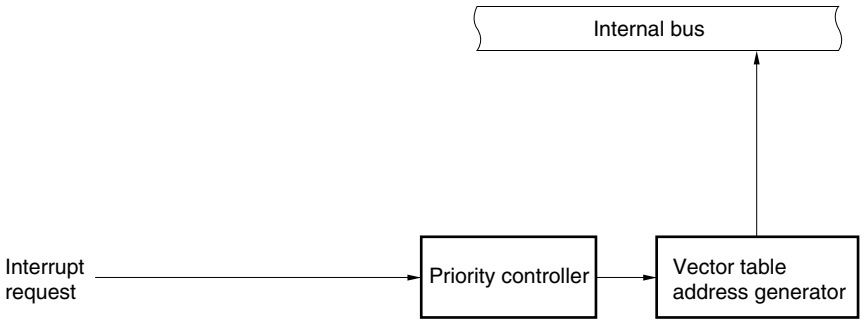
Remark IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

Figure 20-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (INTKR)



(E) Software interrupt



Remarks 1. IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag
KRM: Key return mode register

2. n = 0 to 3: 78K0/KC2-C
n = 0 to 7: 78K0/KE2-C

20.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L, IF1H)
- Interrupt mask flag register (MK0L, MK0H, MK1L, MK1H)
- Priority specification flag register (PR0L, PR0H, PR1L, PR1H)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 20-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 20-2. Flags Corresponding to Interrupt Request Sources (1/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4 Note 1	PIF4 Note 1		PMK4 Note 1		PPR4 Note 1	
INTDA Note 1	DAIF Note 1		DAMK Note 1		DAPR Note 1	
INTP5 Note 2	PIF5 Note 2		PMK5 Note 2		PPR5 Note 2	
INTCE Note 2	CEIF Note 2		CEMK Note 2		CEPR Note 2	
INTERR Note 2	ERRIF Note 2		ERRMK Note 2		ERRPR Note 2	
INTP6 Note 3	PIF6 Note 3		PMK6 Note 3		PPR6 Note 3	
INTRIN Note 3	RINIF Note 3		RINMK Note 3		RINPR Note 3	

Notes 1. If either interrupt source INTP4 or INTDA is generated, bit 5 of IF0L is set (1). Bit 6 of MK0L and PR0L supports both interrupt sources.

2. If either interrupt source INTP5, INTCE, or INTERR is generated, bit 6 of IF0L is set (1). Bit 6 of MK0L and PR0L supports these three interrupt sources.

3. If either interrupt source INTP6 or INTRIN is generated, bit 7 of IF0L is set (1). Bit 7 of MK0L and PR0L supports both interrupt sources.

Table 20-2. Flags Corresponding to Interrupt Request Sources (2/3)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register
INTSR60 Note 1	SRIF60 Note 1	IF0H	SRMK60 Note 1	MK0H	SRPR60 Note 1	PR0H
INTSRE60 Note 1	SREIF60 Note 1		SREMK60 Note 1		SREPR60 Note 1	
INTCSI10 Note 2	CSIIF10 Note 2		CSIMK10 Note 2		CSIPR10 Note 2	
INTST0 Note 2	STIF0 Note 2		STMK0 Note 2		STPR0 Note 2	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
INTSR0 Note 3	SRIF0 Note 3		SRMK0 Note 3		SRPR0 Note 3	
INTSRE0 Note 3	SREIF0 Note 3		SREMK0 Note 3		SREPR0 Note 3	
INTRTC Note 4	RTCIF Note 4		RTCMK Note 4		RTCPR Note 4	
INTRTCI Note 4	RTCIIF Note 4		RTCIMK Note 4		RTCIPR Note 4	
INTTM51 Note 5	TMIF51		TMMK51		TMPR51	
INTIICA0	IICAIF0		IICAMK0		IICAPR0	
INTIICA1	IICAIF1		IICAMK1		IICAPR1	
INTKR Note 6	KRIF Note 6		KRMK Note 6		KRPR Note 6	
INTIICA2 Note 6	IICAIF2 Note 6		IICAMK2 Note 6		IICAPR2 Note 6	
INTCSI11	CSIIF11		CSIMK11		CSIPR11	

Notes 1. If either interrupt source INTSAR60 or INTSRE60 is generated, bit 0 of IF0H is set (1). Bit 0 of MK0H and PR0H supports both interrupt sources.

2. If either interrupt source INTCSI10 or INTST0 is generated, bit 2 of IF0H is set (1). Bit 2 of MK0H and PR0H supports both interrupt sources.

3. If either interrupt source INTSR0 or INTSRE0 is generated, bit 1 of IF1L is set (1). Bit 1 of MK1L and PR1L supports both interrupt sources.

4. If either interrupt source INTRTC or INTRTCI is generated, bit 2 of IF1L is set (1). Bit 2 of MK1L and PR1L supports both interrupt sources.

5. When 8-bit timer/event counter 51 and 8-bit timer H1 are used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (see **Figure 8-15 Transfer Timing**).

6. If either interrupt source INTKR or INTIICA2 is generated, bit 6 of IF1L is set (1). Bit 6 of MK1L and PR1L supports both interrupt sources.

Table 20-2. Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register
INTTM001 Note 1	TMIF001 Note 1	IF1H	TMMK001 Note 1	MK1H	TMPR001 Note 1	PR1H
INTSR61 Notes 1, 2	SRIF61 Notes 1, 2		SRMK61 Notes 1, 2		SRPR61 Notes 1, 2	
INTSRE61 Notes 1, 2	SREIF61 Notes 1, 2		SREMK61 Notes 1, 2		SREPR61 Notes 1, 2	
INTTM011 Note 3	TMIF011 Note 3		TMMK011 Note 3		TMPR011 Note 3	
INTST61 Notes 2, 3	STIF61 Notes 2, 3		STMK61 Notes 2, 3		STPR61 Notes 2, 3	
INTTM002	TMIF002		TMMK002		TMPR002	
INTTM012 Note 4	TMIF012 Note 4		TMMK012 Note 4		TMPR012 Note 4	
INTDMU Note 4	DMUIF Note 4		DMUMK Note 4		DMUPR Note 4	
INTRERR Note 5	RERRIF Note 5		RERRMK Note 5		RERRPR Note 5	
INTGP Note 5	GPIF Note 5		GPMK Note 5		GPPR Note 5	
INTREND Note 5	RENDIF Note 5		RENDMK Note 5		RENDPR Note 5	
INTDFULL Note 5	DFULLIF Note 5		DFULLMK Note 5		DFULLPR Note 5	

Notes 1. If either interrupt source INTTM001, INTSR61, or INTSRE61 is generated, bit 0 of IF1H is set (1). Bit 0 of MK1H and PR1H supports these three interrupt sources.

2. 78K0/KE2-C only

3. If either interrupt source INTTM011 or INTST61 is generated, bit 1 of IF1H is set (1). Bit 1 of MK1H and PR1H supports both interrupt sources.

4. If either interrupt source INTTM012 or INTDMU is generated, bit 3 of IF1H is set (1). Bit 3 of MK1H and PR1H supports both interrupt sources.

5. If either interrupt source INTRERR, INTGP, INTREND, or INTDFULL is generated, bit 4 of IF1H is set (1). Bit 4 of MK1H and PR1H supports these four interrupt sources.

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H)

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, and IF1H are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, and IF1L and IF1H are combined to form 16-bit registers IF0 and IF1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 20-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF6 RINIF	PIF5 CEIF ERRIF	PIF4 DAIF	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	CSIIF10 STIF0	STIF60	SRIF60 SREIF60

Address: FFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	CSIIF11	KRIF IICAIF2	IICAIF1	IICAIF0	TMIF51	RTCIF RTCIIF	SRIF0 SREIF0	ADIF

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF1H	0	0	0	RERRIF GPIF RENDIF DFULLIF	TMIF012 DMUIF	TMIF002	TMIF011 STIF61 Note	TMIF001 SRIF61 Note SREIF61 Note

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Note 78K0/KE2-C only.

Cautions 1. Be sure to clear bits 5 to 7 of IF1H to 0.

2. 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.

Cautions 3. 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)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, and MK1H are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, and MK1L and MK1H are combined to form 16-bit registers MK0 and MK1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 20-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK6 RINMK	PMK5 CEMK ERRMK	PMK4 DAMK	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	CSIMK10 STMK0	STMK60	SRMK60 SREMK60

Address: FFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	CAIMK11	KRMK IICAMK2	IICAMK1	IICAMK0	TMMK51	RTCMK RTCIMK	SRMK0 SREMK0	ADMK

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK1H	1	1	1	RERRMK GPMK RENDMK DFULLMK	TMMK012 DMUMK	TMMK002	TMMK011 STMK61 Note	TMMK001 SRMK61 Note SREMK61 Note

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note 78K0/KE2-C only.

Caution Be sure to set bits 5 to 7 of MK1H to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, PR1L, and PR1H are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H, and PR1L and PR1H are combined to form 16-bit registers PR0 and PR1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 20-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	PPR6 RINPR	PPR5 CEPR ERRPR	PPR4 DAPR	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	CSIPR10 STPR0	STPR60	SRPR60 SREPR60

Address: FFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR1L	CSIPR11	KRPR IICAPR2	IICAPR1	IICAPR0	TMPR51	RTCPR RTCIPR	SRPR0 SREPR0	ADPR

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
PR1H	1	1	1	RERRPR GPPR RENDPR DFULLPR	TMPR012 DMUPR	TMPR002	TMPR011 STPR61 Note	TMPR001 SRPR61 Note SREPR61 Note

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Note 78K0/KE2-C only.

Caution Be sure to set bits 5 to 7 of PR1H to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP6.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 20-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 6)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 20-3 shows the ports corresponding to EGPn and EGNn.

Table 20-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120/EXLVI	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31/OCD1A	INTP2
EGP3	EGN3	P32/ROUT/OCD1B	INTP3
EGP4	EGN4	P33/TI51/TO51	INTP4
EGP5	EGN5	P16/TOH1	INTP5
EGP6	EGN6	P140/PCL/RIN	INTP6

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 6

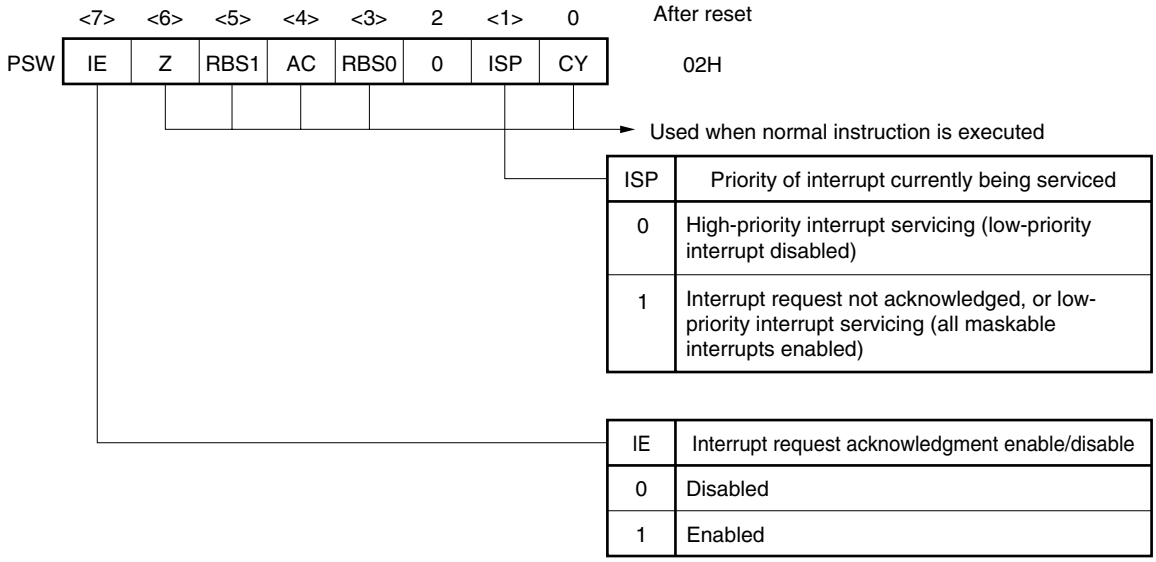
(5) Program status word (PSW)

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 ISP flag that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. 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 02H.

Figure 20-6. Format of Program Status Word



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

20.4 Interrupt Servicing Operations

20.4.1 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 (when the ISP flag is reset to 0).

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 20-4 below.

For the interrupt request acknowledgment timing, see **Figures 20-8** and **20-9**.

Table 20-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: 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.

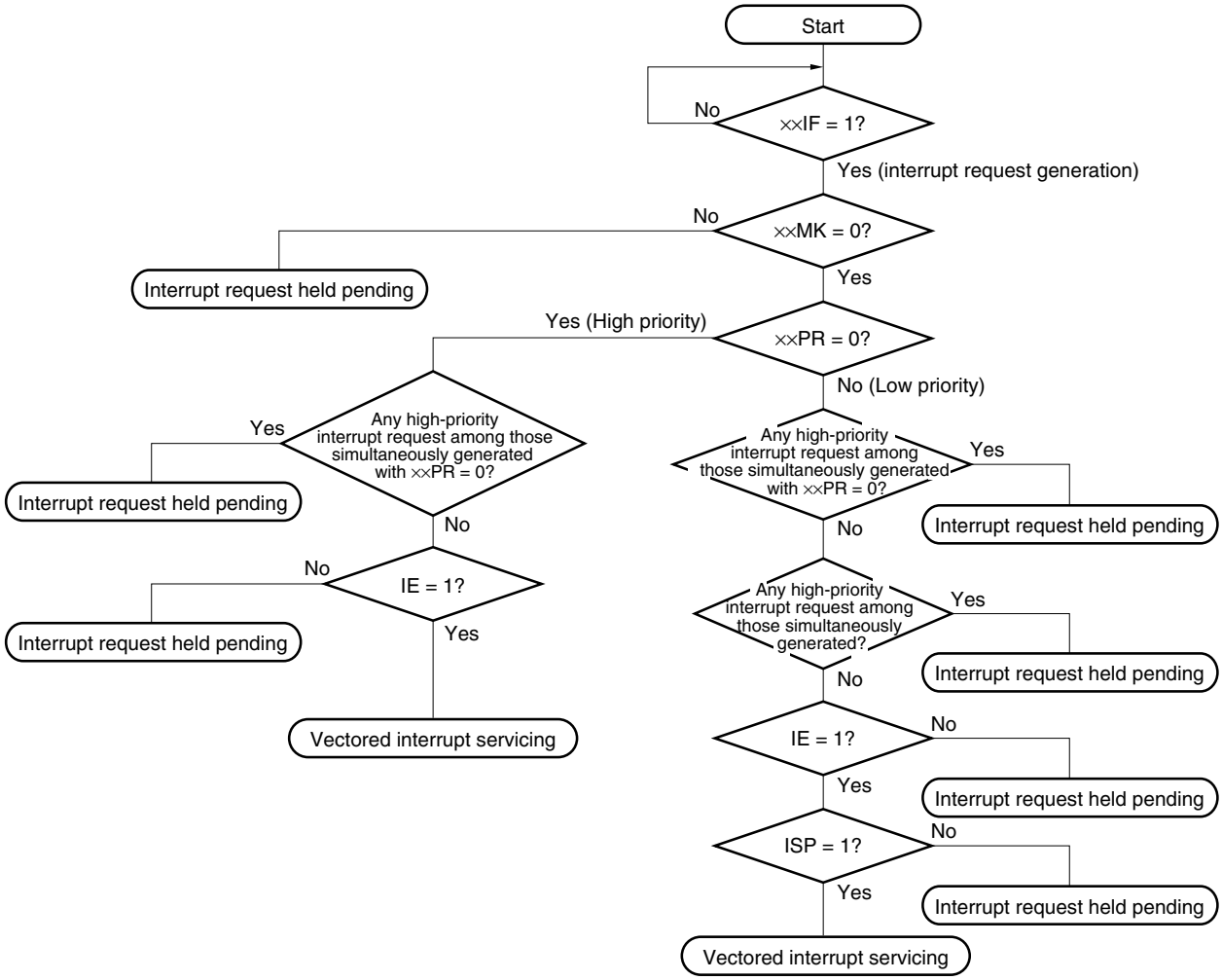
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 20-7 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 ISP flag. The vector table data determined for each interrupt request is the loaded into the PC and branched.

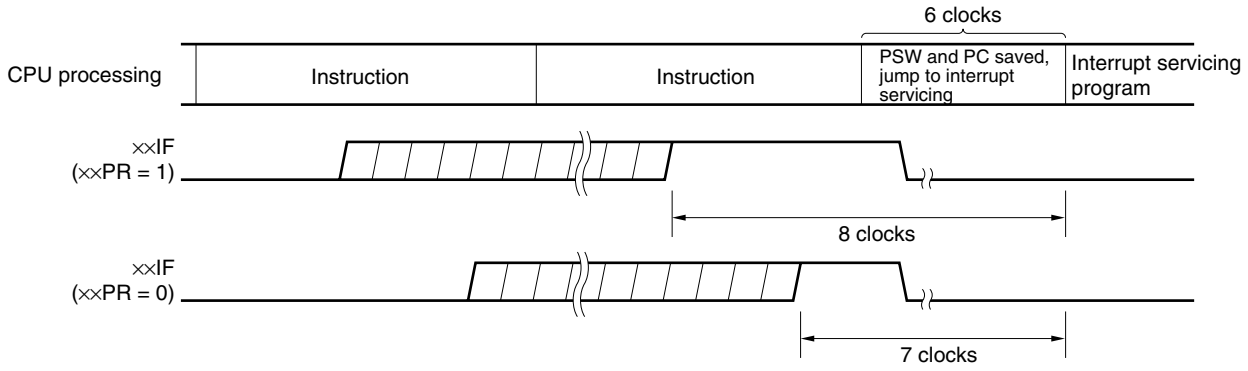
Restoring from an interrupt is possible by using the RETI instruction.

Figure 20-7. Interrupt Request Acknowledgment Processing Algorithm



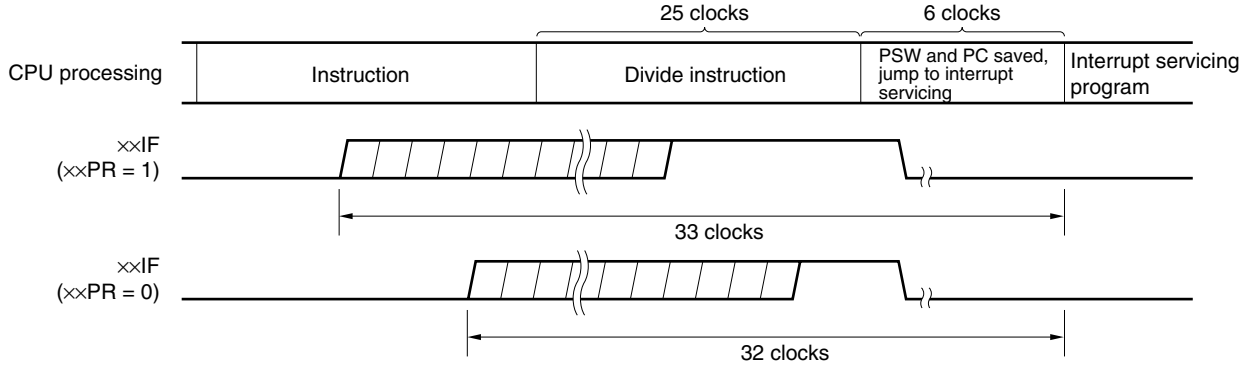
××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR: Priority specification flag
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 20-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 20-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

20.4.2 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 (003EH, 003FH) 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.

20.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 20-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 20-10 shows multiple interrupt servicing examples.

Table 20-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request				Software Interrupt Request
		PR = 0		PR = 1		
		IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
Software interrupt		○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP and IE are flags contained in the PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

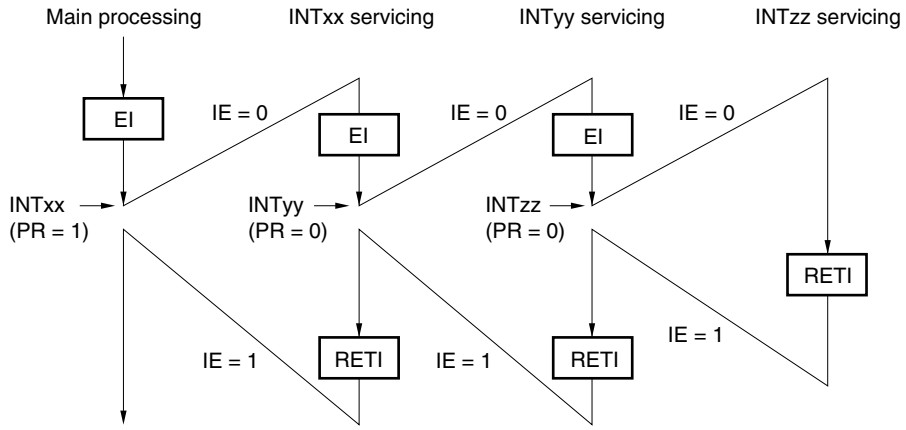
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

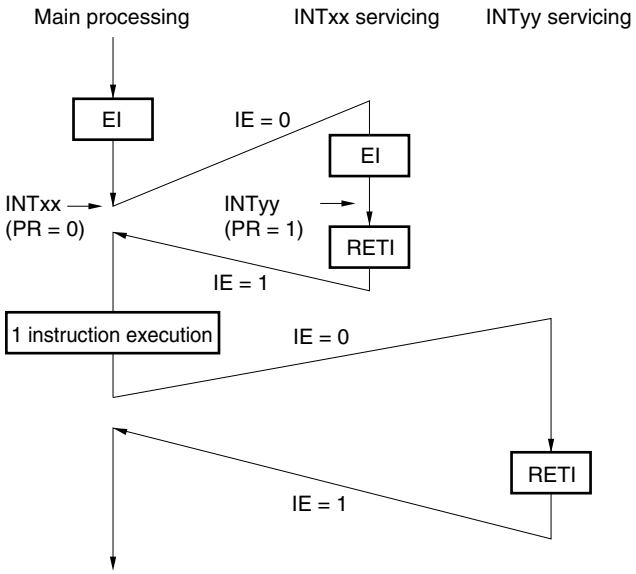
Figure 20-10. Examples of Multiple Interrupt Servicing (1/2)

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

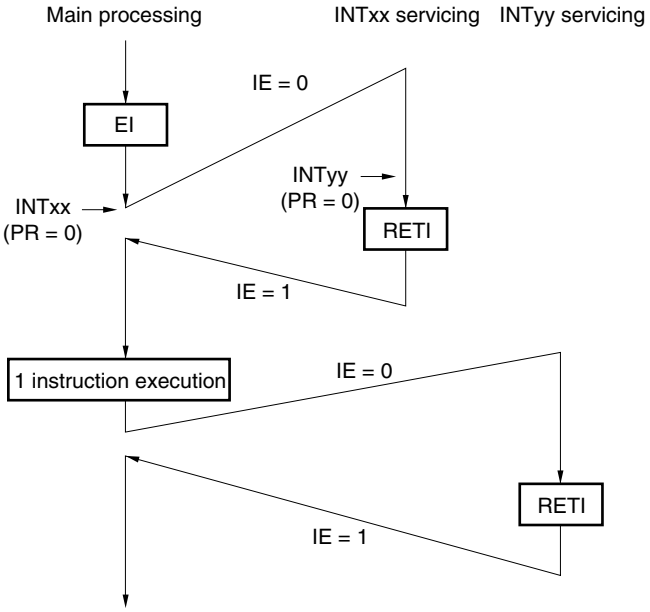


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 20-10. 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 = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

20.4.4 Interrupt request hold

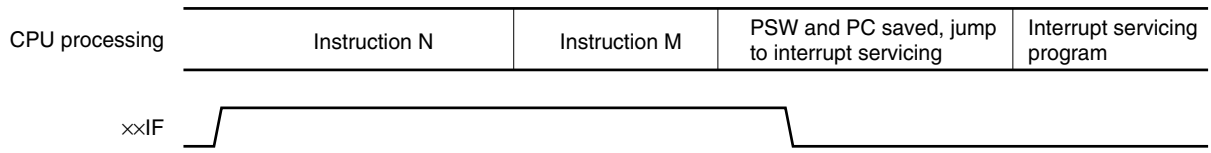
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, $addr16
- BF PSW. bit, $addr16
- BTCLR PSW. bit, $addr16
- EI
- DI
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, and PR1H 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 20-11 shows the timing at which interrupt requests are held pending.

Figure 20-11. 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 21 KEY INTERRUPT FUNCTION

	78K0/KC2-C (PD78F0760, 78F0761, 78F0762)	78K0/KE2-C (PD78F0763, 78F0764, 78F0765)
Key interrupt	4 ch	8 ch

Caution If key interrupt function is used, IICA02 cannot be used.

21.1 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 pins (KRn).

Table 21-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRMn	Controls KRn signal in 1-bit units.

Remark n = 0 to 3: 78K0/KC2-C
n = 0 to 7: 78K0/KE2-C

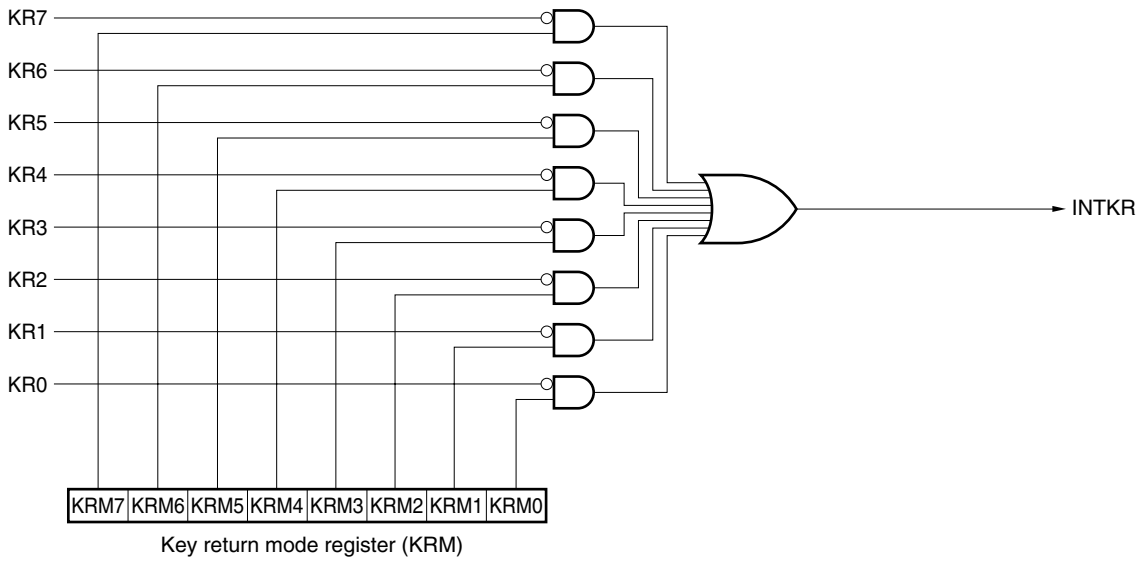
21.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 21-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 21-1. Block Diagram of Key Interrupt



Remark KR0 to KR3, KRM0 to KRM3: 78K0/KC2-C
KR0 to KR7, KRM0 to KRM7: 78K0/KE2-C

21.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register controls the KRMn bit using the KRn signal.

KRM is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears KRM to 00H.

Figure 21-2. Format of Key Return Mode Register (KRM)

(1) 78K0/KC2-C

Address: FF6EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	0	0	0	0	KRM3	KRM2	KRM1	KRM0

(2) 78K0/KE2-C

Address: FF6EH 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 bits used is set to 1, set bit n (PU7n) of the corresponding pull-up resistor register 7 (PU7) to 1.

2. If KRM is changed, the interrupt request flag may be set. Therefore, disable interrupts and then change the KRM register. Clear the interrupt request flag and enable interrupts.

3. The bits not used in the key interrupt mode can be used as normal ports.

4. For the 78K0/KC2-C, be sure to set bits 4 to 7 of KRM to "0".

Remark n = 0 to 3: 78K0/KC2-C

n = 0 to 7: 78K0/KE2-C

CHAPTER 22 STANDBY FUNCTION

22.1 Standby Function and Configuration

22.1.1 Standby function

The standby function is designed to reduce the operating current of the system. 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, internal low-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 subsystem clock oscillation cannot be stopped. 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.

22.1.2 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. When X1 clock oscillation starts with the internal high-speed oscillation clock or subsystem clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

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, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 22-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status		
						f_X = 10 MHz	f_X = 20 MHz
1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	0	0	$2^{14}/f_X$ min.	1.64 ms min.	819.2 μs min.
1	1	1	1	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_X$ min.	6.55 ms min.	3.27 ms min.

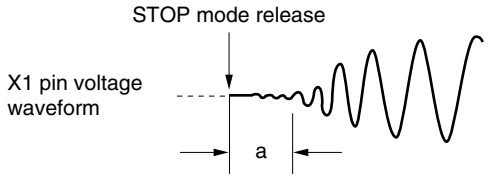
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

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



Remark f_X: 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 OSTS to 05H.

Figure 22-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fx = 10 MHz	fx = 20 MHz
0	0	1	2^{11}/fx	204.8 μs	102.4 μs
0	1	0	2^{13}/fx	819.2 μs	409.6 μs
0	1	1	2^{14}/fx	1.64 ms	819.2 μs
1	0	0	2^{15}/fx	3.27 ms	1.64 ms
1	0	1	2^{16}/fx	6.55 ms	3.27 ms
Other than above			Setting prohibited		

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

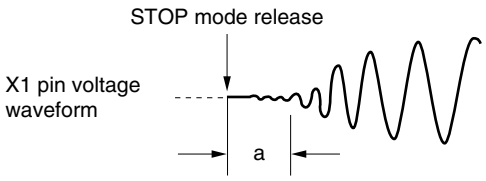
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. 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.

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



Remark fx: X1 clock oscillation frequency

22.2 Standby Function Operation

22.2.1 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, or subsystem clock. The operating statuses in the HALT mode are shown below.

Table 22-1. 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_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	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)	Status before HALT mode was set is retained
		f_{EXCLK}	Operates or stops by external clock input		Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
	f_{RL}		Status before HALT mode was set is retained. Clock supply to watchdog timer stops when "internal low-speed oscillator cannot be stopped" is set by option byte.		
CPU			Operation stopped		
Flash memory					
RAM			Status before HALT mode was set is retained		
Port (latch)					
16-bit timer/event counter		00	These are set up according to the settings from before entering the HALT mode[Note 1].		
		01			
		02			
8-bit timer/event counter		50			
		51			
8-bit timer		H0			
		H1			
Real-time counter					
Watchdog timer			These are set up according to the settings from before entering the HALT mode[Note 1]. Clock supply to watchdog timer stops when "internal low-speed oscillator can be stopped by software" is set by option byte.		
Clock output			These are set up according to the settings from before entering the HALT mode[Note 1].		
Buzzer output[Note 2]					
A/D converter					
Serial interface	UART0, UART60, UART61[Note 2]				
	CSI10, CSI11				
	IICA00 to IICA02				
	CEC				
Remote controller receiver					
Multiplier/divider					
Power-on-clear function			Operable		
Low-voltage detection function			Low voltages can be detected according to the settings from before entering the HALT mode.		
External interrupt			Low voltages can be operated according to the settings from before entering the HALT mode.		

Notes 1. They remain stopped if they were stopped before the HALT mode was entered. If they were running before the HALT mode was entered, they are used according to the clock signal selected by each peripheral hardware macro.

2. 78K0/KE2-C only

Remarks 1. f_{RH}: Internal high-speed oscillation clock, f_X: X1 clock,
f_{EXCLK}: External main system clock,
f_{XT}: XT1 clock, f_{RL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. See **1.7 Block Diagram** and **1.8 Outline of Functions**.

Table 22-1. 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_{RH}	Status before HALT mode was set is retained	
		f_X		
		f_{EXCLK}	Operates or stops by external clock input	
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained
		f_{EXCLKS}	Operates or stops by external clock input	Operation continues (cannot be stopped)
	f_{RL}		Status before HALT mode was set is retained. Clock supply to watchdog timer stops when "internal low-speed oscillator cannot be stopped" is set by option byte.	
CPU			Operation stopped	
Flash memory				
RAM			Status before HALT mode was set is retained	
Port (latch)				
16-bit timer/event counter		00 Note 1	These are set up according to the settings from before entering the HALT mode Note 2.	
		01 Note 1		
		02 Note 1		
8-bit timer/event counter		50 Note 1		
		51 Note 1		
8-bit timer		H0		
		H1		
Real-time counter				
Watchdog timer			These are set up according to the settings from before entering the HALT mode Note 2. Clock supply to watchdog timer stops when "internal low-speed oscillator can be stopped by software" is set by option byte.	
Clock output			These are set up according to the settings from before entering the HALT mode Note 2.	
Buzzer output Note 3				
A/D converter				
Serial interface	UART0, UART60, UART61 Note 3			
	CSI10, CSI11 Note 1			
	IICA00 to IICA02 Note 1			
	CEC			
Remote controller receiver				
Multiplier/divider				
Power-on-clear function			Operable	
Low-voltage detection function			Low voltages can be detected according to the settings from before entering the HALT mode.	
External interrupt			Low voltages can be operated according to the settings from before entering the HALT mode.	

Notes 1. When the CPU is operating on the subsystem clock and the internal high-speed oscillation clock and high-speed system clock have been stopped, do not start operation of these functions on the external clock input from peripheral hardware pins.

2. They remain stopped if they were stopped before the HALT mode was entered. If they were running before the HALT mode was entered, they are used according to the clock signal selected by each peripheral hardware macro.

3. 78K0/KE2-C only

Remarks 1. f_{RH}: Internal high-speed oscillation clock, f_X: X1 clock ,
f_{EXCLK}: External main system clock,
f_{XT}: XT1 cloc f_{RL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. See **1.7 Block Diagram** and **1.8 Outline of Functions**.

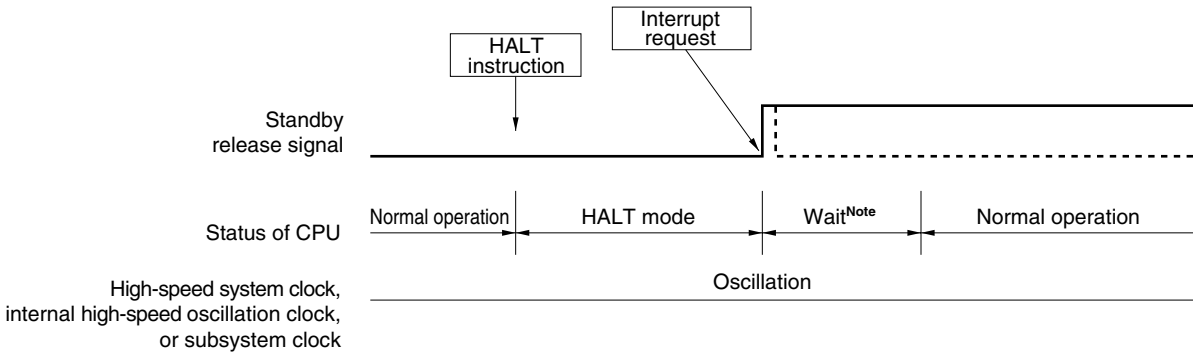
(2) HALT mode release

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 22-3. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 11 or 12 clocks
- When vectored interrupt servicing is not carried out: 4 or 5 clocks

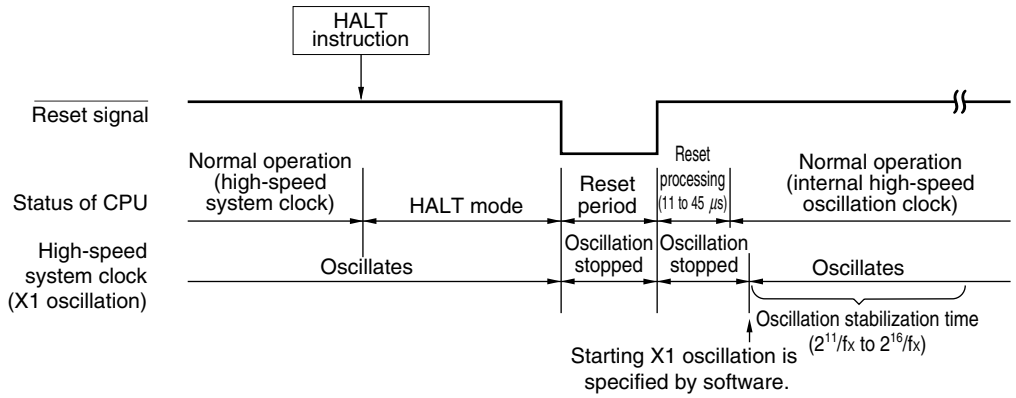
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

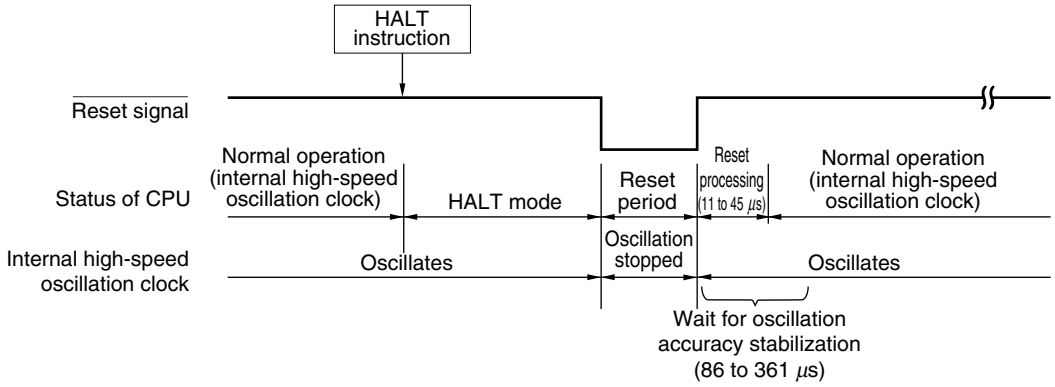
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 22-4. HALT Mode Release by Reset

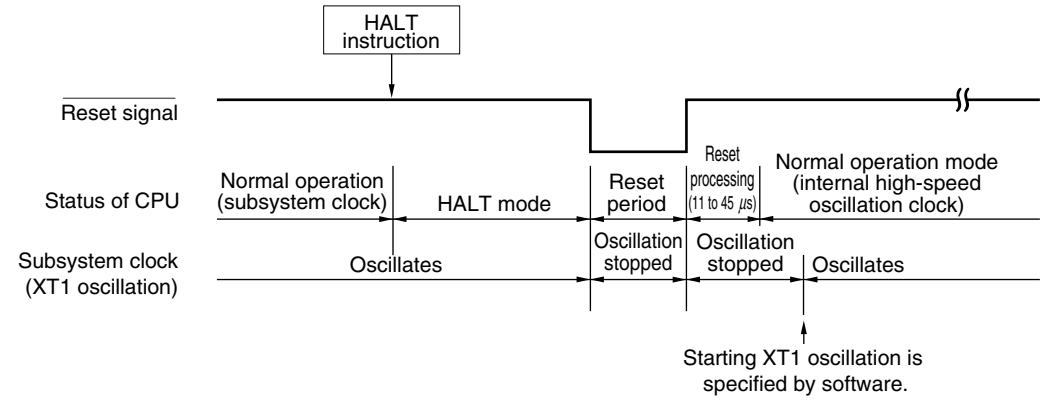
(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Table 22-2. Operation in Response to Interrupt Request in HALT Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
Reset	–	–	×	×	Reset processing

×: don't care

22.2.2 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.

Caution Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 22-3. Operating Statuses in STOP Mode

Item \ STOP Mode Setting			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Stopped		
		f_X			
		f_{EXCLK}	Input invalid		
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
	f_{RL}		Status before STOP mode was set is retained.		
CPU			Operation stopped		
Flash memory					
RAM			Status before STOP mode was set is retained. Clock supply to watchdog timer stops when "internal low-speed oscillator cannot be stopped" is set by option byte.		
Port (latch)					
16-bit timer/event counter		00[Note 1]	Operation stopped		
		01[Note 1]			
		02[Note 1]			
8-bit timer/event counter		50[Note 1]	These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when TI50 is selected as the count clock.		
		51[Note 1]	These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when TI51 is selected as the count clock.		
8-bit timer		H0	These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation.		
		H1	These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when $f_{RL}/2^7$ is selected as the count clock.		
Real-time counter			These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when subsystem clock is selected as the count clock.		
Watchdog timer			These are set up according to the settings from before entering the STOP mode[Note 2]. Clock supply to watchdog timer stops when "internal low-speed oscillator can be stopped by software" is set by option byte.		
Clock output			These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when subsystem clock is selected as the count clock.		
Buzzer output[Note 3]			Operation stopped		
A/D converter					
Serial interface	UART0, UART60, UART61[Note 3]		These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when TM50 output is selected as the serial clock during 8-bit timer/event counter 50 operation.		
	CSI10, CSI11[Note 1]		These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when external clock is selected as the serial clock.		
	IICA00 to IICA02[Note 1]		These are set up according to the settings from before entering the STOP mode[Note 2]. Wakeup by address match operable.		
	CEC		These are set up according to the settings from before entering the STOP mode[Note 2]. Operable only when subsystem clock is selected as the count clock.		
Remote controller receiver					
Multiplier/divider			Operation stopped		
Power-on-clear function			Operable		
Low-voltage detection function			Low voltages can be detected according to the settings from before entering the STOP mode.		
External interrupt			Low voltages can be operated according to the settings from before entering the STOP mode.		

Note 1. Do not start operation of these functions on the external clock input from peripheral hardware pins in the stop mode.

Notes 2. They remain stopped if they were stopped before the STOP mode was entered. If they were running before the STOP mode was entered, they are used according to the clock signal selected by each peripheral hardware macro.

3. 78K0/KE2-C only

Remarks 1. f_{RH}: Internal high-speed oscillation clock, f_X: X1 clock,
f_{EXCLK}: External main system clock, f_{XT}: XT1 clock,
f_{RL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. See **1.7 Block Diagram** and **1.8 Outline of Functions**.

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. Even if "internal low-speed oscillator can be stopped by software" is selected by the option byte, the internal low-speed oscillation clock continues in the STOP mode in the status before the STOP mode is set. To stop the internal low-speed oscillator's oscillation in the STOP mode, stop it by software and then execute the STOP instruction.

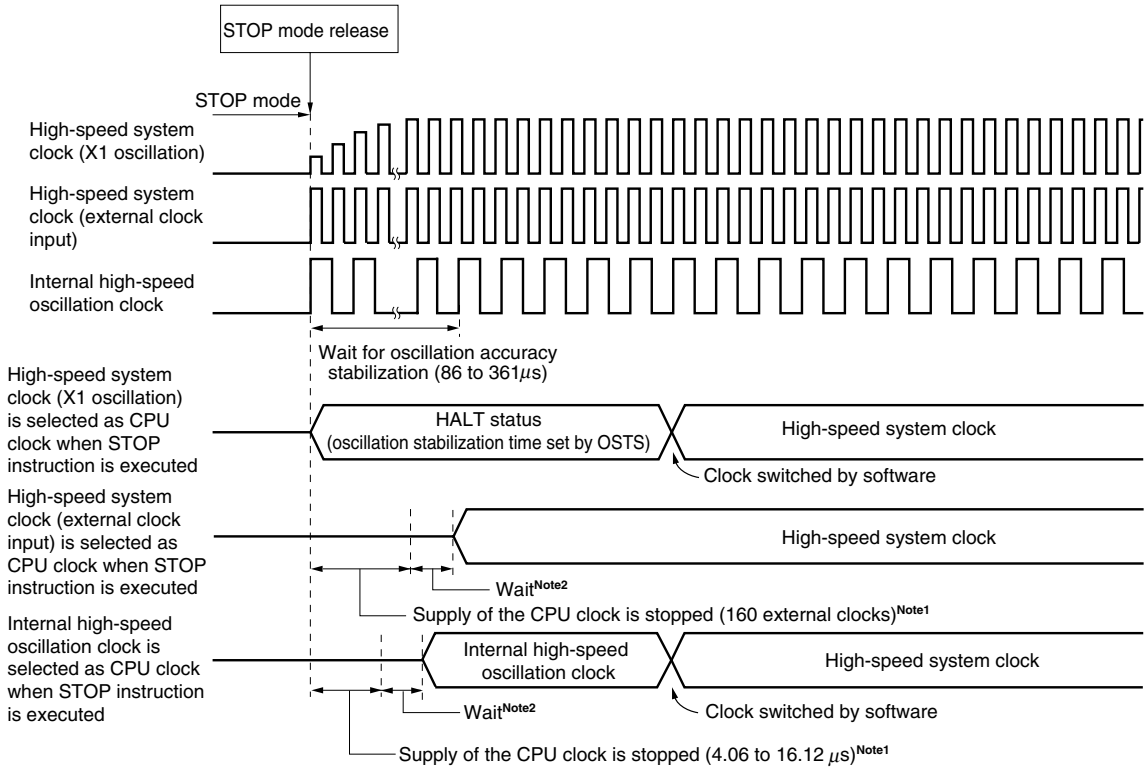
3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), switch the CPU clock to the internal high-speed oscillation clock before the execution of the STOP instruction using the following procedure.
<1> Set RSTOP to 0 (starting oscillation of the internal high-speed oscillator) → <2> Set MCM0 to 0 (switching the CPU from X1 oscillation to internal high-speed oscillation) → <3> Check that MCS is 0 (checking the CPU clock) → <4> Check that RSTS is 1 (checking internal high-speed oscillation operation) → <5> Execute 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. If the STOP instruction is executed when AMPH = 1, supply of the CPU clock is stopped for 4.06 to 16.12 μs after the STOP mode is released when the internal high-speed oscillation clock is selected as the CPU clock, or for the duration of 160 external clocks when the high-speed system clock (external clock input) is selected as the CPU clock.

5. Execute the STOP instruction after having confirmed that the internal high-speed oscillator is operating stably (RSTS = 1).

(2) STOP mode release

Figure 22-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Notes 1. When AMPH = 1

2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

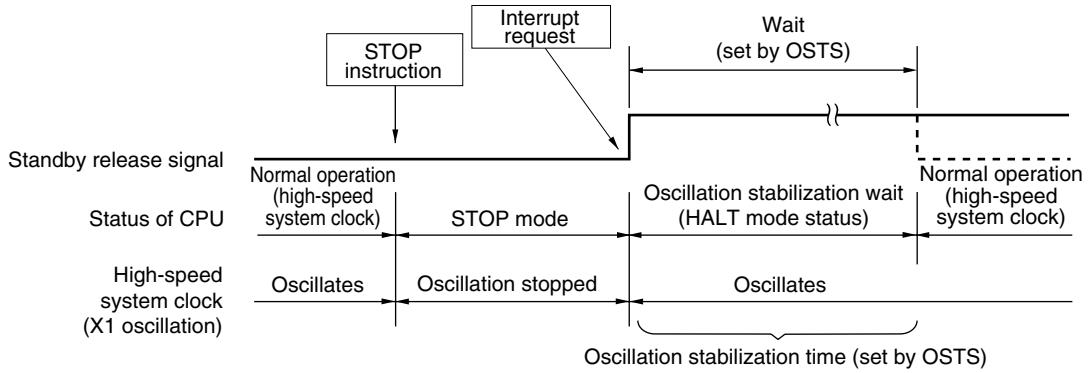
The STOP mode can be released by the following two sources.

(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.

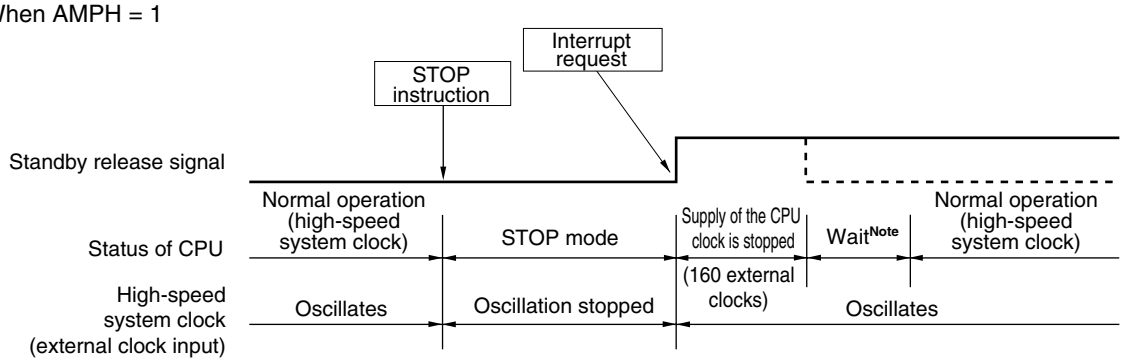
Figure 22-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock

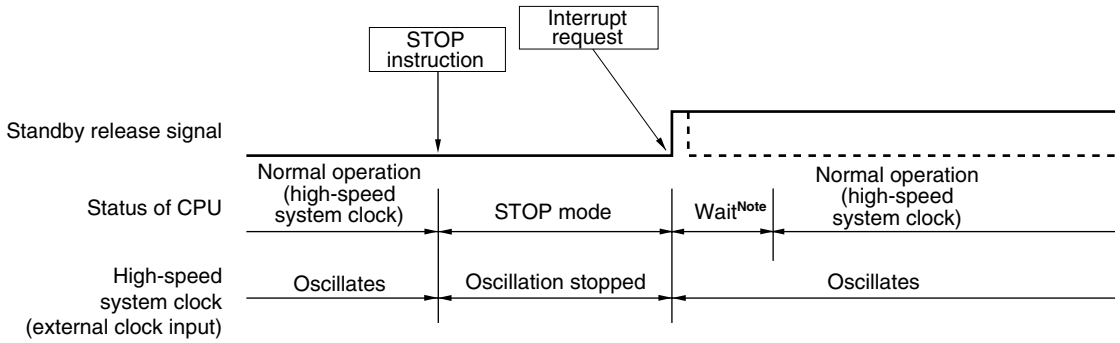


(2) When high-speed system clock (external clock input) is used as CPU clock

- When AMPH = 1



- When AMPH = 0



Note The wait time is as follows:

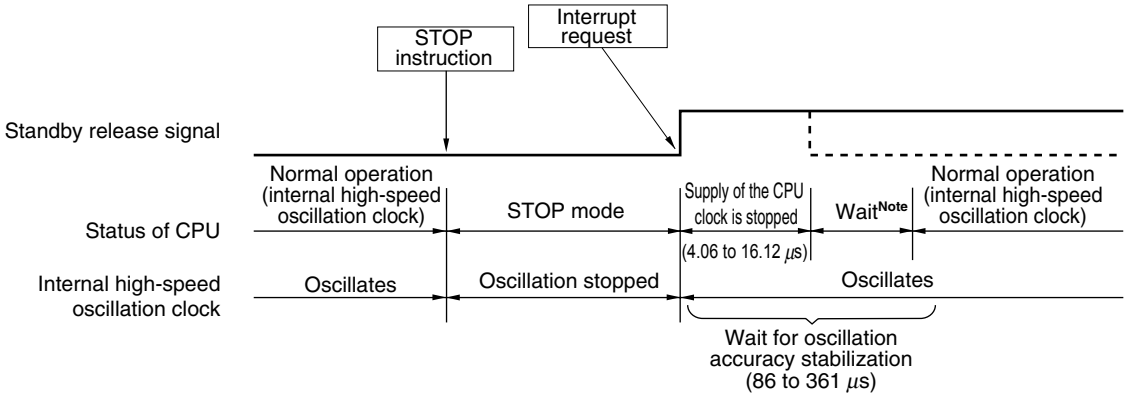
- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

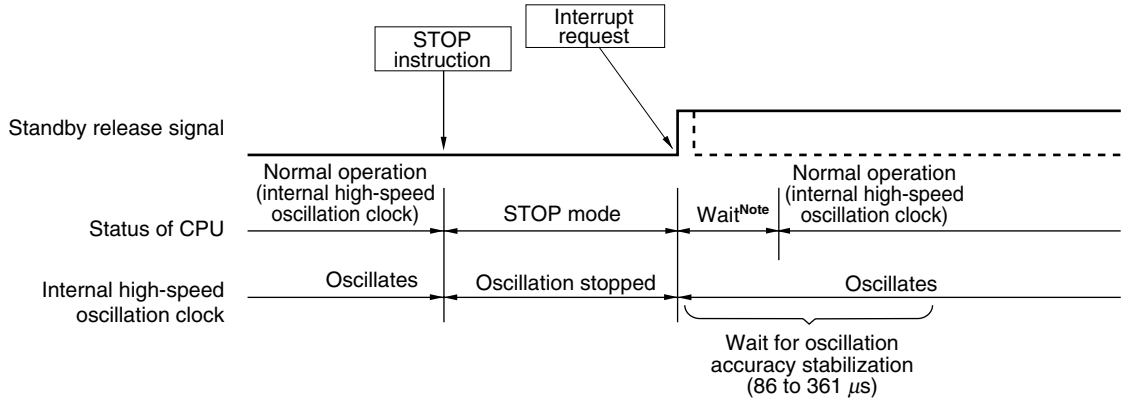
Figure 22-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock

- When AMPH = 1



- When AMPH = 0



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

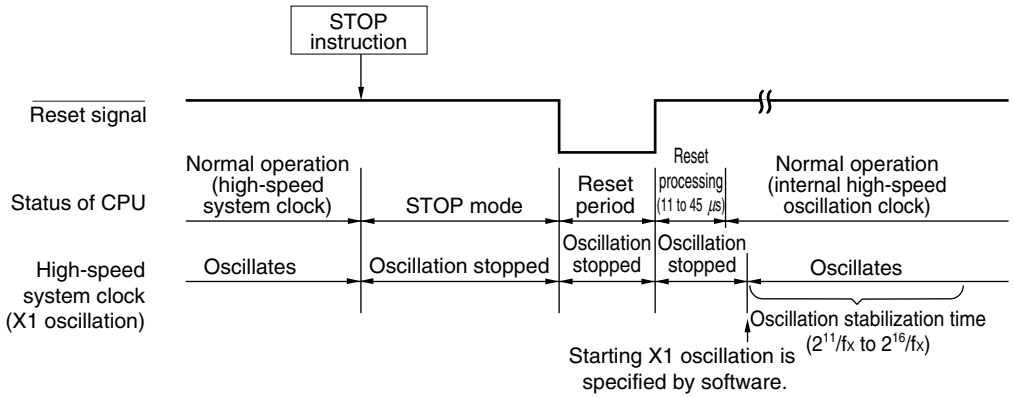
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

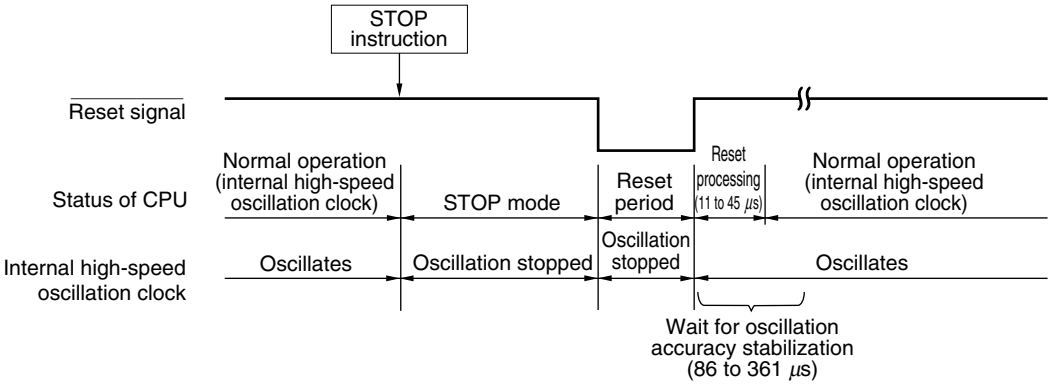
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 22-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Table 22-4. Operation in Response to Interrupt Request in STOP Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	STOP mode held
Reset	–	–	×	×	Reset processing

×: don't care

CHAPTER 23 RESET FUNCTION

The following four 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 and detection voltage of low-power-supply detector (LVI)

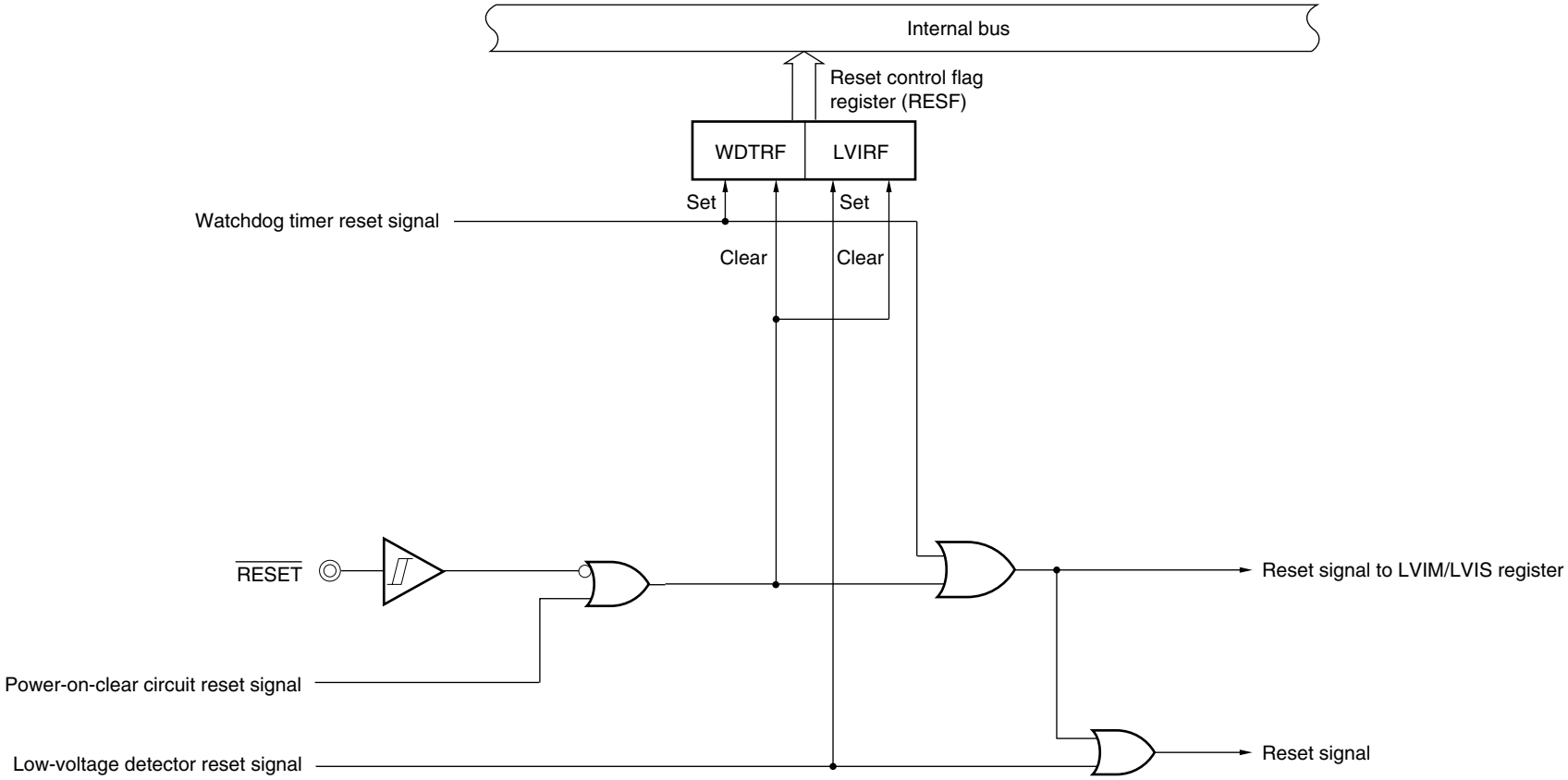
External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is generated.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Tables 23-1 and 23-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

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 23-2** to **23-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 25 LOW-VOLTAGE DETECTOR**) after reset processing.

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance.

Figure 23-1. 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 selection register

Figure 23-2. Timing of Reset by $\overline{\text{RESET}}$ Input

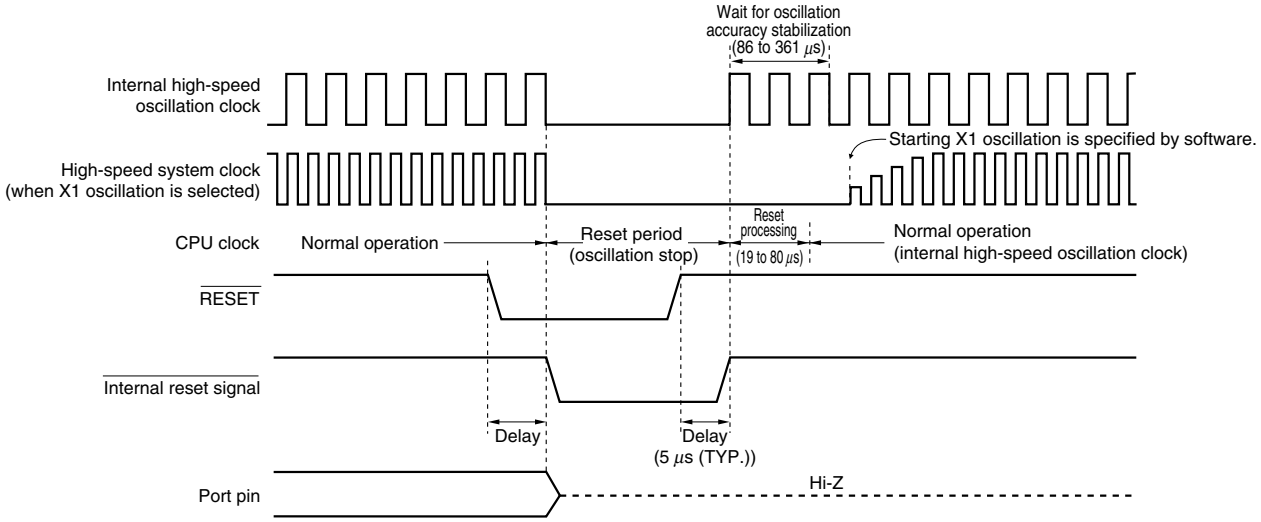
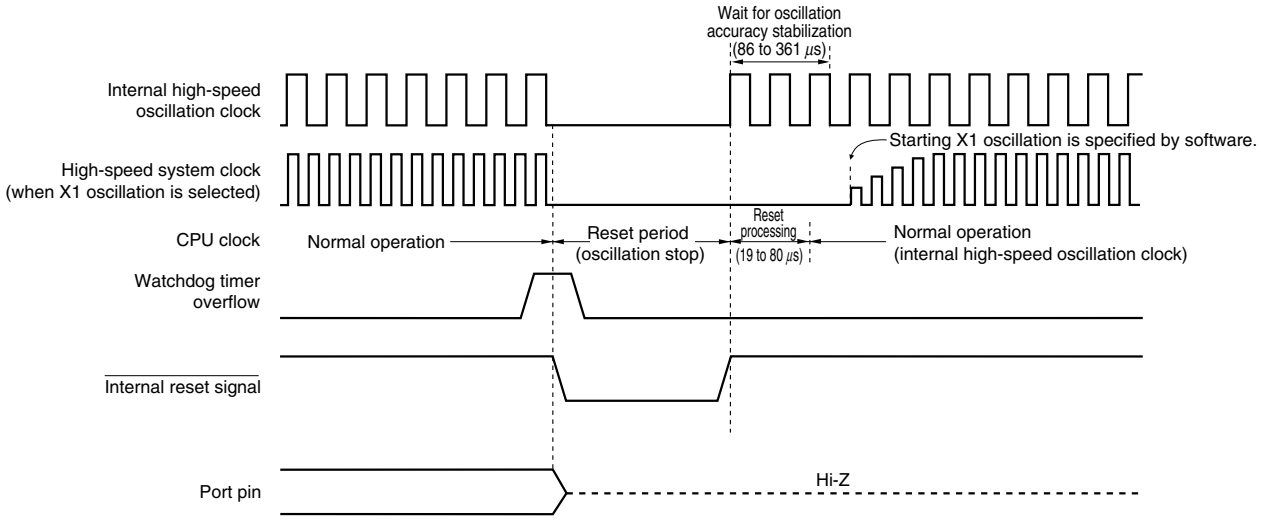
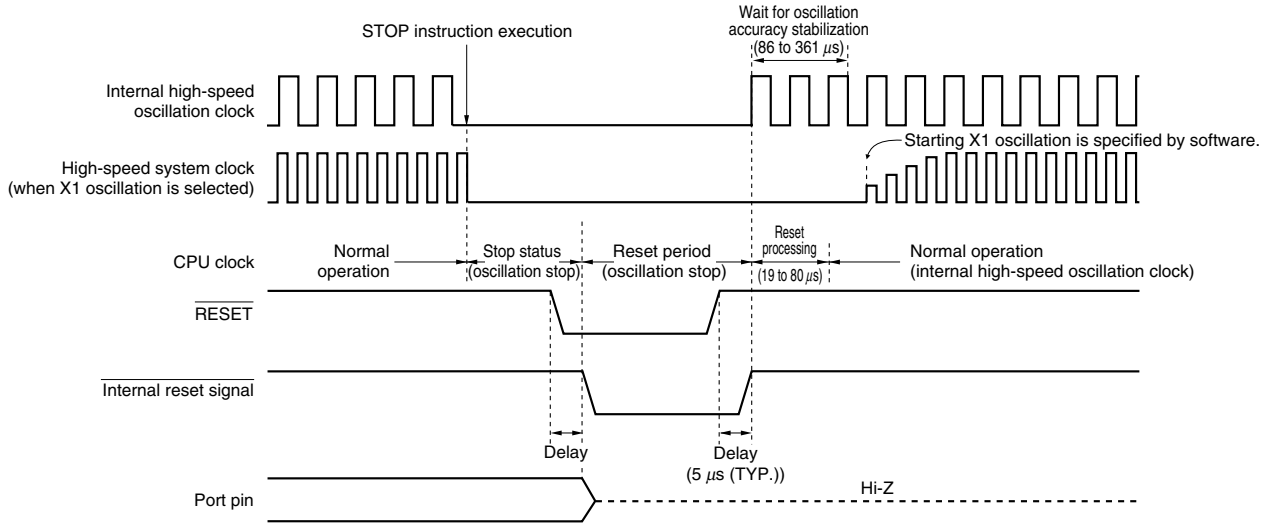


Figure 23-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

Figure 23-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 25 LOW-VOLTAGE DETECTOR**.

Table 23-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{RH}	Operation stopped
		f_X	Operation stopped (pin is I/O port mode)
		f_{EXCLK}	Clock input invalid (pin is I/O port mode)
	Subsystem clock	f_{XT}	Operation stopped (pin is I/O port mode)
	f_{RL}		Operation stopped
CPU			
Flash memory			
RAM			
Port (latch)			
16-bit timer/event counter		00	
		01	
		02	
8-bit timer/event counter		50	
		51	
		52	
8-bit timer		H0	
		H1	
		H2	
Real-time counter			
Watchdog timer			
Clock output			
Buzzer output Note			
A/D converter			
Serial interface	UART0, UART60, UART61 Note		
	CSI10, CSI11		
	IICA00 to IICA02		
	CEC		
Remote controller receiver			
Multiplier/divider			
Power-on-clear function			Operable
Low-voltage detection function			Operation stopped
External interrupt			

Note 78K0/KE2-C only

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 oscillation clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 oscillation clock
f_{RL}: Internal low-speed oscillation clock

Table 23-2. Hardware Statuses After Reset Acknowledgment (1/5)

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)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0 to P4, P5[Note 3], P6, P7, P12 to P14) (output latches)		00H
Port mode registers (PM0 to PM4, PM5[Note 3], PM6, PM7, PM12, PM14)		FFH
Port input mode register 7 (PIM7)		00H
Pull-up resistor option registers (PU0, PU1, PU3, PU4, PU5[Note 3], PU7, PU12, PU14)		00H
Port function register 4 (PF4)		00H
Port function register 7 (PF7)		00H
Internal expansion RAM size switching register (IXS)		0CH[Note 4]
Internal memory size switching register (IMS)		CFH[Note 4]
Clock operation mode select register (OSCCTL)		00H
Processor clock control register (PCC)		01H
Internal oscillation mode register (RCM)		80H
Main OSC control register (MOC)		80H
Main clock mode register (MCM)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		05H
16-bit timer/event counters 00, 01, 02	Timer counters 00, 01, 02 (TM00, TM01, TM02)	0000H
	Capture/compare registers 000, 001, 002, 010, 011, 012 (CR000, CR001, CR002, CR010, CR011, CR012)	0000H
	Mode control registers 00, 01, 02 (TMC00, TMC01, TMC02)	00H
	Prescaler mode registers 00, 01, 02 (PRM00, PRM01, PRM02)	00H
	Capture/compare control registers 00, 01[Note 3], 02[Note 3] (CRC00, CRC01[Note 3], CRC02[Note 3])	00H
	Timer output control registers 00, 01[Note 3], 02[Note 3] (TOC00, TOC01[Note 3], TOC02[Note 3])	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. 78K0/KE2-C only

4. The initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) after a reset release are constant (IMS = CFH, IXS = 0CH) in all the 78K0/Kx2-C products, regardless of the internal memory capacity. Therefore, after a reset is released, be sure to set the following values for each product.

78K0/KC2-C	78K0/KE2-C	IMS	IXS
μPD78F0760	μPD78F0763	C8H	0CH
μPD78F0761	μPD78F0764	CCH	0AH
μPD78F0762	μPD78F0765	CFH	

Table 23-2. Hardware Statuses After Reset Acknowledgment (2/5)

Hardware		After Reset Acknowledgment[Note 1]
8-bit timer/event counters 50, 51	Timer counters 50, 51 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	00H
	Timer clock selection registers 50, 51 (TCL50, TCL51)	00H
	Mode control registers 50, 51 (TMC50, TMC51)	00H
8-bit timers H0, H1	Compare registers 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	Mode registers (TMHMD0, TMHMD1)	00H
	Carrier control register 1 (TMCYC1)[Note 2]	00H
Real-time counter	Clock selection register (RTCCL)	00H
	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 selection register (CKS)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 3]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	A/D converter mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	08H
Serial interface UART0	Receive buffer register 0 (RXB0)	FFH
	Transmit shift register 0 (TXS0)	FFH
	Asynchronous serial interface operation mode register 0 (ASIM0)	01H
	Asynchronous serial interface reception error status register 0 (ASIS0)	00H
	Baud rate generator control register 0 (BRGC0)	1FH

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. 8-bit timer H1 only.

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

Table 23-2. Hardware Statuses After Reset Acknowledgment (3/5)

Hardware		Status After Reset Acknowledgment[Note 1]
Serial interfaces UART60, UART61[Note 2]	Receive buffer registers 60, 61[Note 2] (RXB60, RXB61[Note 2])	FFH
	Transmit buffer registers 60, 61[Note 2] (TXB60, TXB61[Note 2])	FFH
	Asynchronous serial interface operation mode registers 60, 61[Note 2] (ASIM60, ASIM61[Note 2])	01H
	Asynchronous serial interface reception error status registers 60, 61[Note 2] (ASIS60, ASIS61[Note 2])	00H
	Asynchronous serial interface transmission status registers 60, 61[Note 2] (ASIF60, ASIF61[Note 2])	00H
	Clock selection registers 60, 61[Note 2] (CKSR60, CKSR61[Note 2])	00H
	Baud rate generator control registers 60, 61[Note 2] (BRGC60, BRGC61[Note 2])	FFH
	Asynchronous serial interface control registers 60, 61[Note 2] (ASICL60, ASICL61[Note 2])	16H
	Input switch control register (ISC)	00H
Serial interfaces CSI10, CSI11	Transmit buffer registers 10, 11 (SOTB10, SOTB11)	00H
	Serial I/O shift registers 10, 11 (SIO10, SIO11)	00H
	Serial operation mode registers 10, 11 (CSIM10, CSIM11)	00H
	Serial clock selection registers 10, 11 (CSIC10, CSIC11)	00H
Serial interfaces IICA00, IICA01, IICA02	Shift registers 00, 01, 02 (IICA00, IICA01, IICA02)	00H
	Status registers 00, 01, 02 (IICAS00, IICAS01, IICAS02)	00H
	Flag registers 00, 01, 02 (IICAF00, IICAF01, IICAF02)	00H
	Control registers 00, 01, 02 (IICACTL00, IICACTL01, IICACTL02)	00H
	Control registers 10, 11, 12 (IICACTL10, IICACTL11, IICACTL12)	00H
	Low-level width setting registers 00, 01, 02 (IICAWL00, IICAWL01, IICAWL01)	FFH
	High-level width setting registers 00, 01, 02 (IICAWH00, IICAWH01, IICAWH02)	FFH
	Slave address registers 00, 01, 02 (SVA00, SVA01, SVA02)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. UART61 is 78K0/KE2-C only.

Table 23-2. Hardware Statuses After Reset Acknowledgment (4/5)

Hardware		Status After Reset Acknowledgment[Note 1]
CEC transmission/reception circuit	CEC transmission buffer register (CTXD)	00H
	CEC reception buffer register (CRXD)	00H
	CEC communication error status register (CECES)	00H
	CEC communication status register (CECS)	00H
	CEC communication error flag clear trigger register (CECFC)	00H
	CEC control register 0 (CECCTL0)	00H
	CEC local address setting register (CADR)	0000H
	CEC control register 1 (CECCTL1)	00H
	CEC transmission start bit width setting register (STATB)	0000H
	CEC transmission start bit low width setting register (STATL)	0000H
	CEC transmission logical 0 low width setting register (LGC0L)	0000H
	CEC transmission logical 1 low width setting register (LGC1L)	0000H
	CEC transmission data bit width setting register (DATB)	0000H
	CEC data bit reference width setting register (NOMP)	0000H
	CEC reception data sampling time setting register (NOMT)	0000H
	CEC reception start bit minimum low width setting register (STATLL)	0000H
	CEC reception start bit maximum low width setting register (STATLH)	0000H
	CEC reception start bit minimum bit width setting register (STATBL)	0000H
	CEC reception start bit maximum bit width setting register (STATBH)	0000H
	CEC reception logical 0 minimum low width setting register (LGC0LL)	0000H
	CEC reception logical 0 maximum low width setting register (LGC0LH)	0000H
	CEC reception logical 1 minimum low width setting register (LGC1LL)	0000H
	CEC reception logical 1 maximum low width setting register (LGC1LH)	0000H
	CEC reception data bit minimum bit width setting register (DATBL)	0000H
	CEC reception data bit maximum bit width setting register (DATBH)	0000H
Key interrupt	Key return mode register (KRM)	00H
Reset function	Reset control flag register (RESF)	00H[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 2]
	Low-voltage detection level selection register (LVIS)	00H[Note 2]

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
RESF	WDTRF bit	Cleared (0)	Cleared (0)	Set (1)	Held
	LVIRF bit			Held	Set (1)
LVIM		Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS					

Table 23-2. Hardware Statuses After Reset Acknowledgment (5/5)

Hardware		Status After Reset Acknowledgment[Note]
Remote controller receiver	Remote controller receive shift register (RMSR)	00H
	Remote controller receive data register (RMDR)	00H
	Remote controller shift register receive counter register (RMSCR)	00H
	Remote controller receive GPHS compare register (RMGPHS)	00H
	Remote controller receive GPHL compare register (RMGPHL)	00H
	Remote controller receive DLS compare register (RMDLS)	00H
	Remote controller receive DLL compare register (RMDLL)	00H
	Remote controller receive DH0S compare register (RMDH0S)	00H
	Remote controller receive DH0L compare register (RMDH0L)	00H
	Remote controller receive DH1S compare register (RMDH1S)	00H
	Remote controller receive DH1L compare register (RMDH1L)	00H
	Remote controller receive end width select register (RMER)	00H
	Remote controller receive interrupt status register (INTS)	00H
	Remote controller receive interrupt status clear register (INTC)	00H
	Remote controller receive control register (RMCN)	00H
	Remote controller receive data output control register (RMSW)	00H
Multiplier/divider	Remainder data register 0 (SDR0)	0000H
	Multiplication/division data register A0 (MDA0H, MDA0L)	0000H
	Multiplication/division data register B0 (MDB0)	0000H
	Multiplier/divider control register 0 (DMUC0)	00H
Interrupt	Request flag registers 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	Mask flag registers 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	Priority specification flag registers 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

23.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/Kx2-C. 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 set RESF to 00H.

Figure 23-5. Format of Reset Control Flag Register (RESF)

Address: FFACH After reset: 00H[Note] R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Note The value after reset varies depending on the reset source.

Caution Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 23-3.

Table 23-3. RESF Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held
LVIRF			Held	Set (1)

CHAPTER 24 POWER-ON-CLEAR CIRCUIT

24.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
  In the 1.59 V POC mode (option byte: POCMODE = 0), the reset signal is released when the supply voltage (V_{DD}) exceeds 1.59 V ±0.15 V.
  In the 2.7 V/1.59 V POC mode (option byte: POCMODE = 1), the reset signal is released when the supply voltage (V_{DD}) exceeds 2.7 V ±0.2 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V), generates internal reset signal when $V_{DD} < V_{POC}$.

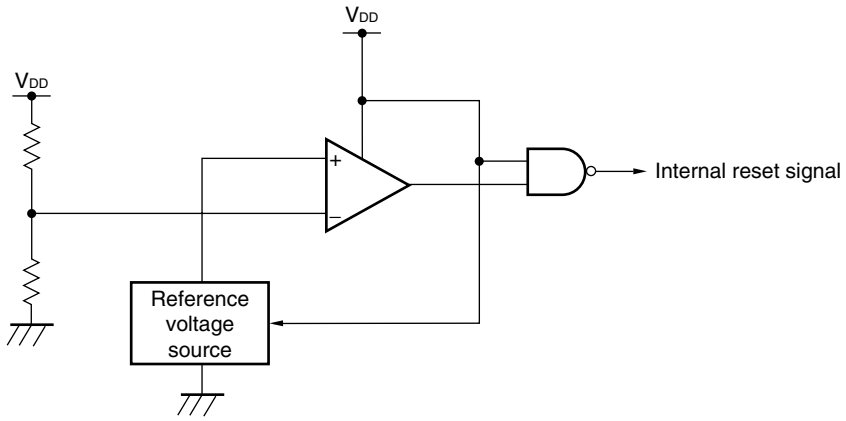
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark 78K0/Kx2-C 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) or low-voltage-detector (LVI). RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT or LVI.
For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

24.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 24-1.

Figure 24-1. Block Diagram of Power-on-Clear Circuit



24.3 Operation of Power-on-Clear Circuit

(1) In 1.59 V POC mode (option byte: POCMODE = 0)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POC} = 1.59 V ±0.15 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{POC}$.

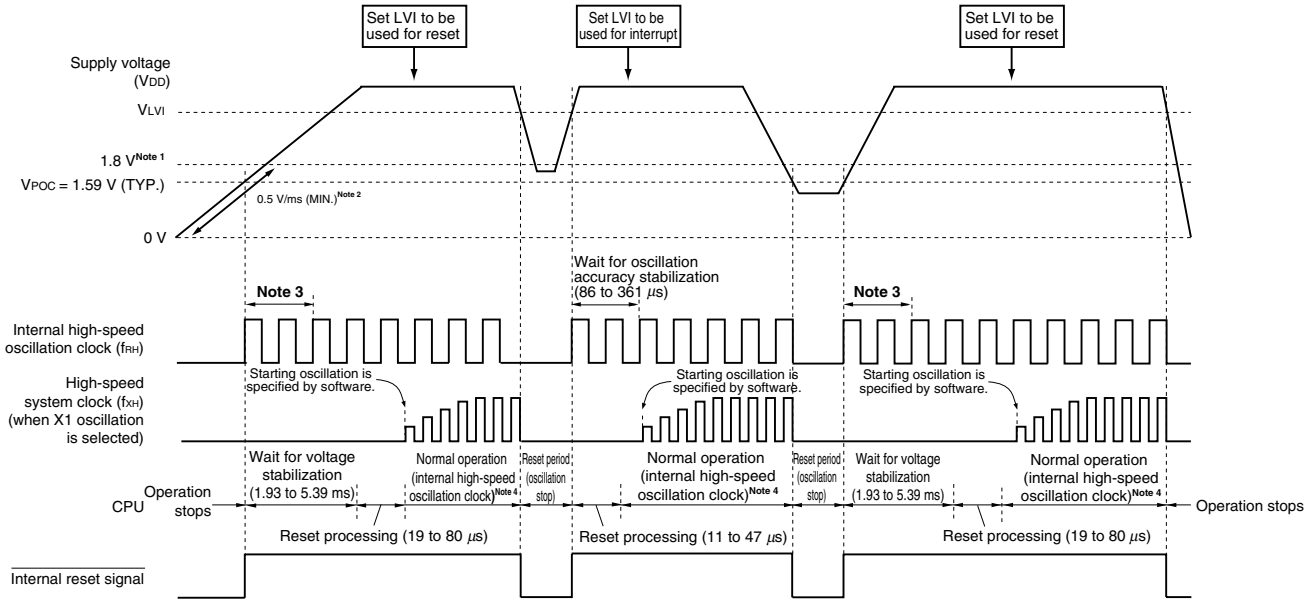
(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{DDPOC} = 2.7 V ±0.2 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{DDPOC}$.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 24-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. If the voltage rises to 1.8 V at a rate slower than 0.5 V/ms (MIN.) on power application, input a low level to the $\overline{\text{RESET}}$ pin after power application and before the voltage reaches 1.8 V, or set the 2.7 V/1.59 V POC mode by using an option byte (POCMODE = 1).

3. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

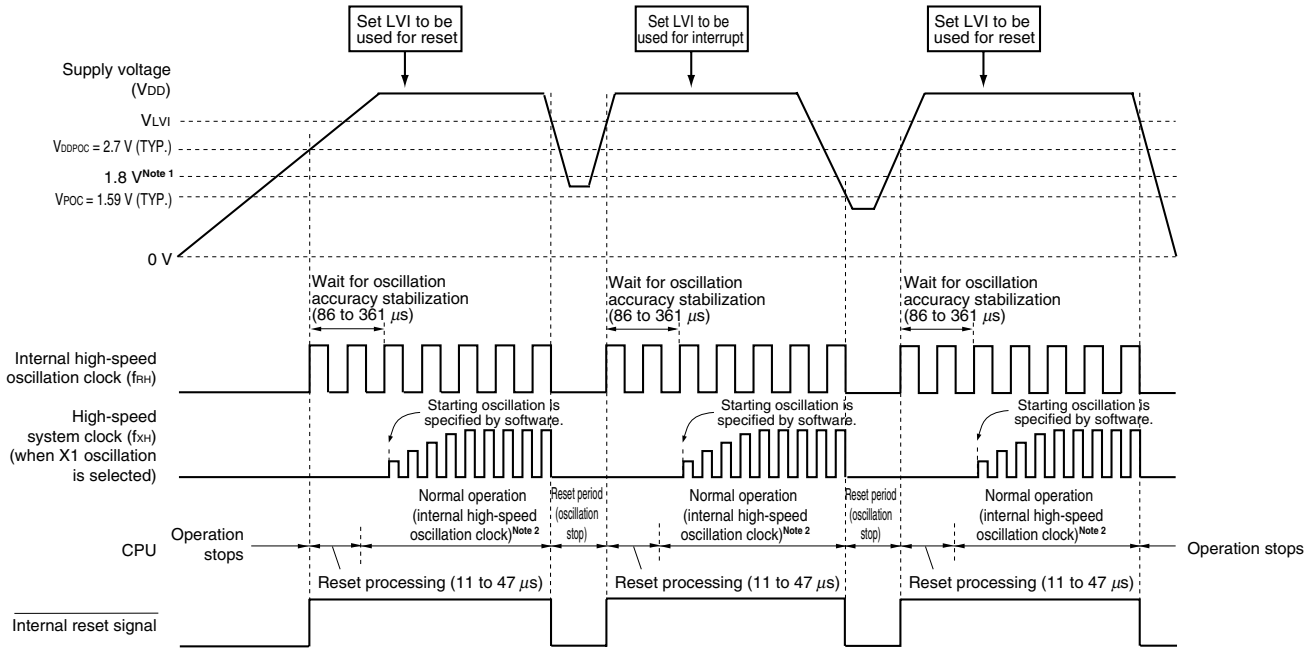
4. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 25 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

Figure 24-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. 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.

Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 25 LOW-VOLTAGE DETECTOR).

2. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the time the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) is within 1.93 to 5.39 ms, a power supply stabilization wait time of 0 to 5.39 ms occurs automatically before reset processing and the reset processing time becomes 19 to 80 μs.

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

24.4 Cautions for Power-on-Clear Circuit

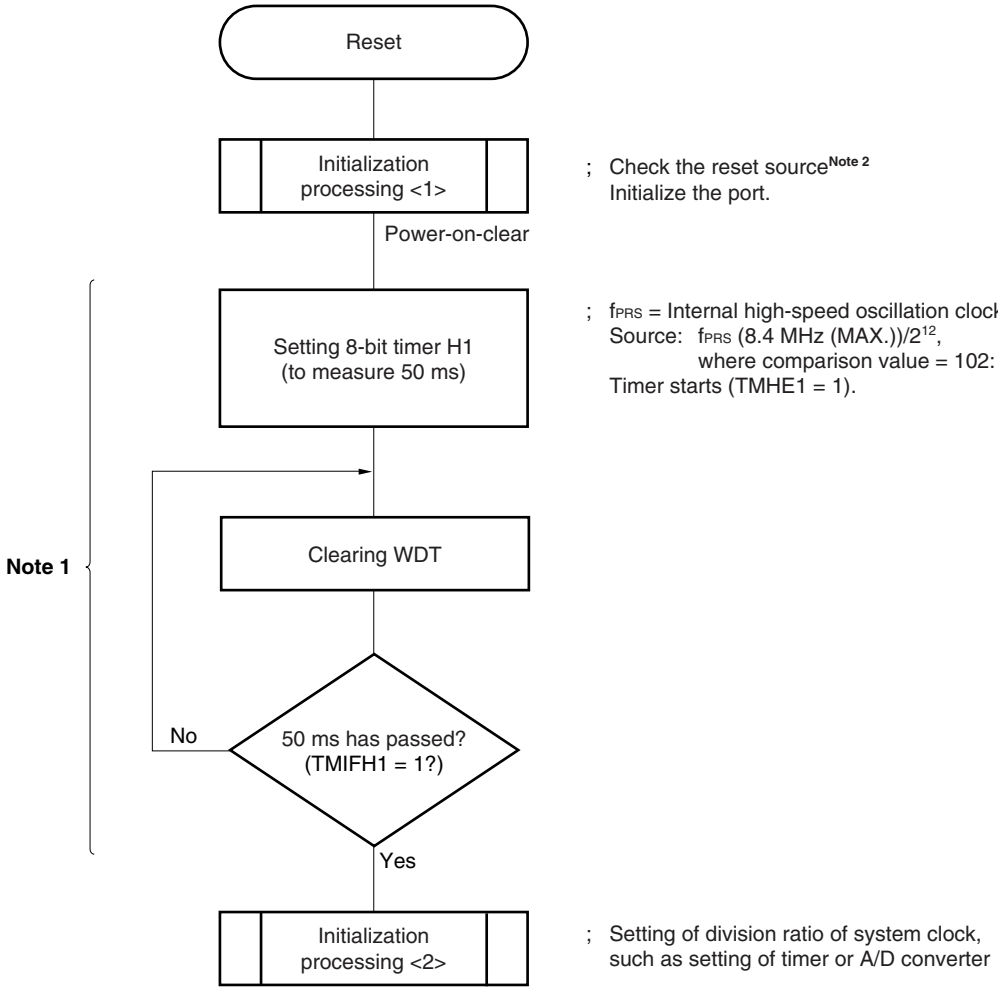
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 24-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

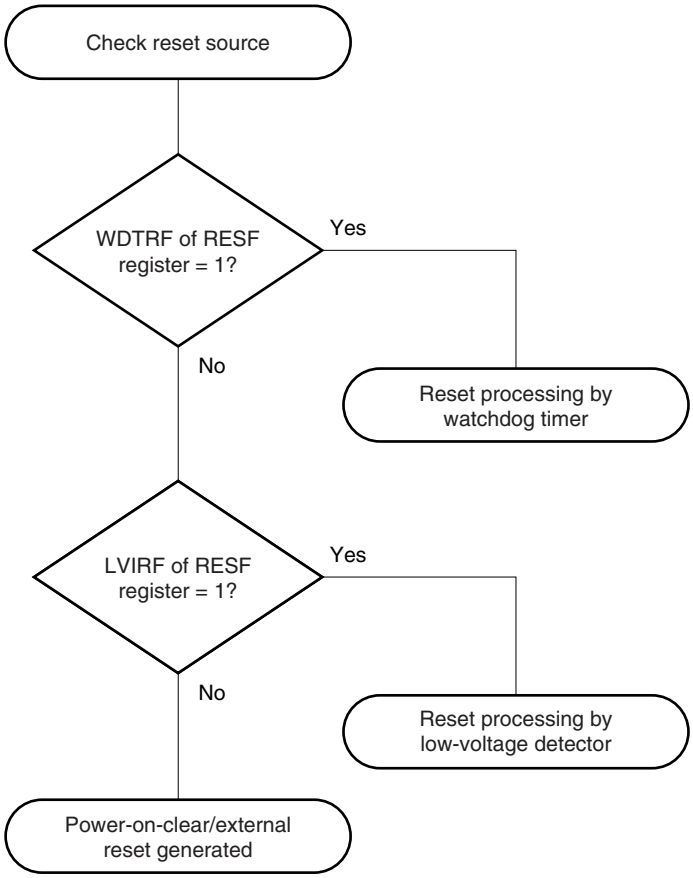


Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Figure 24-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



CHAPTER 25 LOW-VOLTAGE DETECTOR

25.1 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 (TYP.): fixed), and generates an internal reset or internal interrupt signal.
- The supply voltage (V_{DD}) or input voltage from an external input pin (EXLVI) can be selected by software.
- Reset or interrupt function can be selected by software.
- Detection levels (16 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

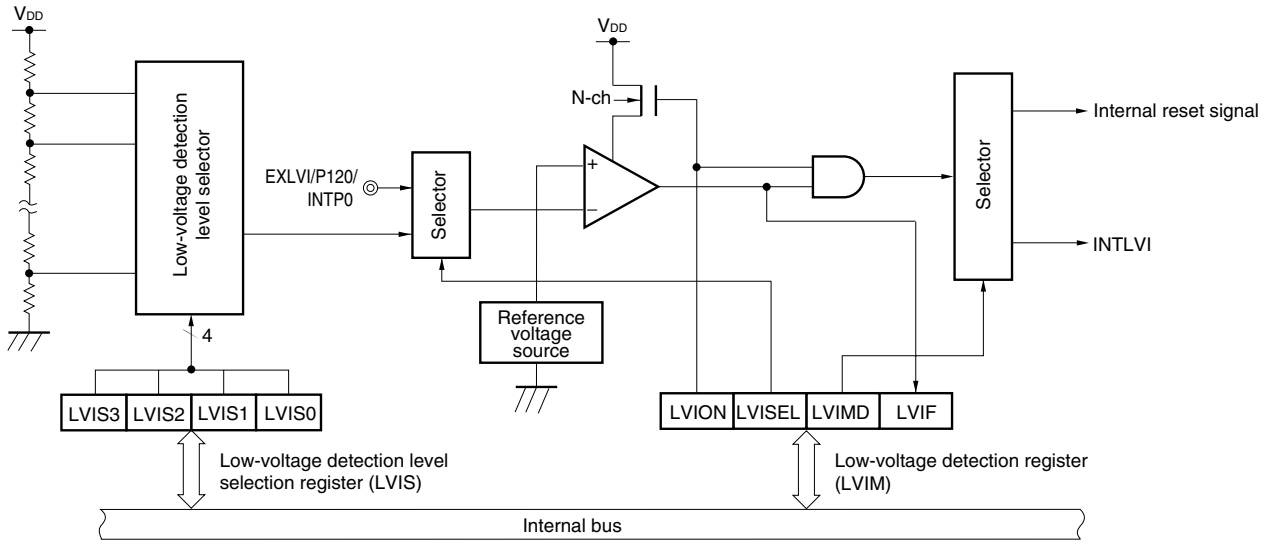
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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 23 RESET FUNCTION**.

25.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 25-1.

Figure 25-1. Block Diagram of Low-Voltage Detector



25.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level selection register (LVIS)
- Port mode register 12 (PM12)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 25-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFBEH 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 operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when 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. This bit is cleared to 00H upon a reset other than an LVI reset.

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 an operation stabilization time (10 μs (MAX.)) from when LVION is set to 1 until operation is stabilized. After operation has stabilized, 200 μs (MIN.) are required from when a state below LVI detection voltage has been entered, until LVIF is set (1).

Cautions 1. To stop LVI, follow either of the procedures below.

- **When using 8-bit memory manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

2. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

3. When using LVI as an interrupt, if LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

(2) Low-voltage detection level selection 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.

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 25-3. Format of Low-Voltage Detection Level Selection Register (LVIS)

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

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	0	0	0	V_{LVI0} (4.24 V±0.1 V)
0	0	0	1	V_{LVI1} (4.09 V±0.1 V)
0	0	1	0	V_{LVI2} (3.93 V±0.1 V)
0	0	1	1	V_{LVI3} (3.78 V±0.1 V)
0	1	0	0	V_{LVI4} (3.62 V±0.1 V)
0	1	0	1	V_{LVI5} (3.47 V±0.1 V)
0	1	1	0	V_{LVI6} (3.32 V±0.1 V)
0	1	1	1	V_{LVI7} (3.16 V±0.1 V)
1	0	0	0	V_{LVI8} (3.01 V±0.1 V)
1	0	0	1	V_{LVI9} (2.85 V±0.1 V)
1	0	1	0	V_{LVI10} (2.70 V±0.1 V)
1	0	1	1	V_{LVI11} (2.55 V±0.1 V)
1	1	0	0	V_{LVI12} (2.39 V±0.1 V)
1	1	0	1	V_{LVI13} (2.24 V±0.1 V)
1	1	1	0	V_{LVI14} (2.08 V±0.1 V)
1	1	1	1	V_{LVI15} (1.93 V±0.1 V)

Note The value of LVIS is not reset but retained as is, upon a reset by LVI. It is cleared to 00H upon other resets.

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2. Do not change the value of LVIS during LVI operation.

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI} = 1.21 V (TYP.)) is fixed. Therefore, setting of LVIS is not necessary.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM12 to FFH.

Figure 25-4. Format of Port Mode Register 12 (PM12)

Address: FF2CH 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)

25.4 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 (TYP.)), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

(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 (TYP.)). 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

25.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

- 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 an operation stabilization time (10 μs (MAX.)).
  - <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 25-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

  Cautions 1. <1> must always be executed. 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.

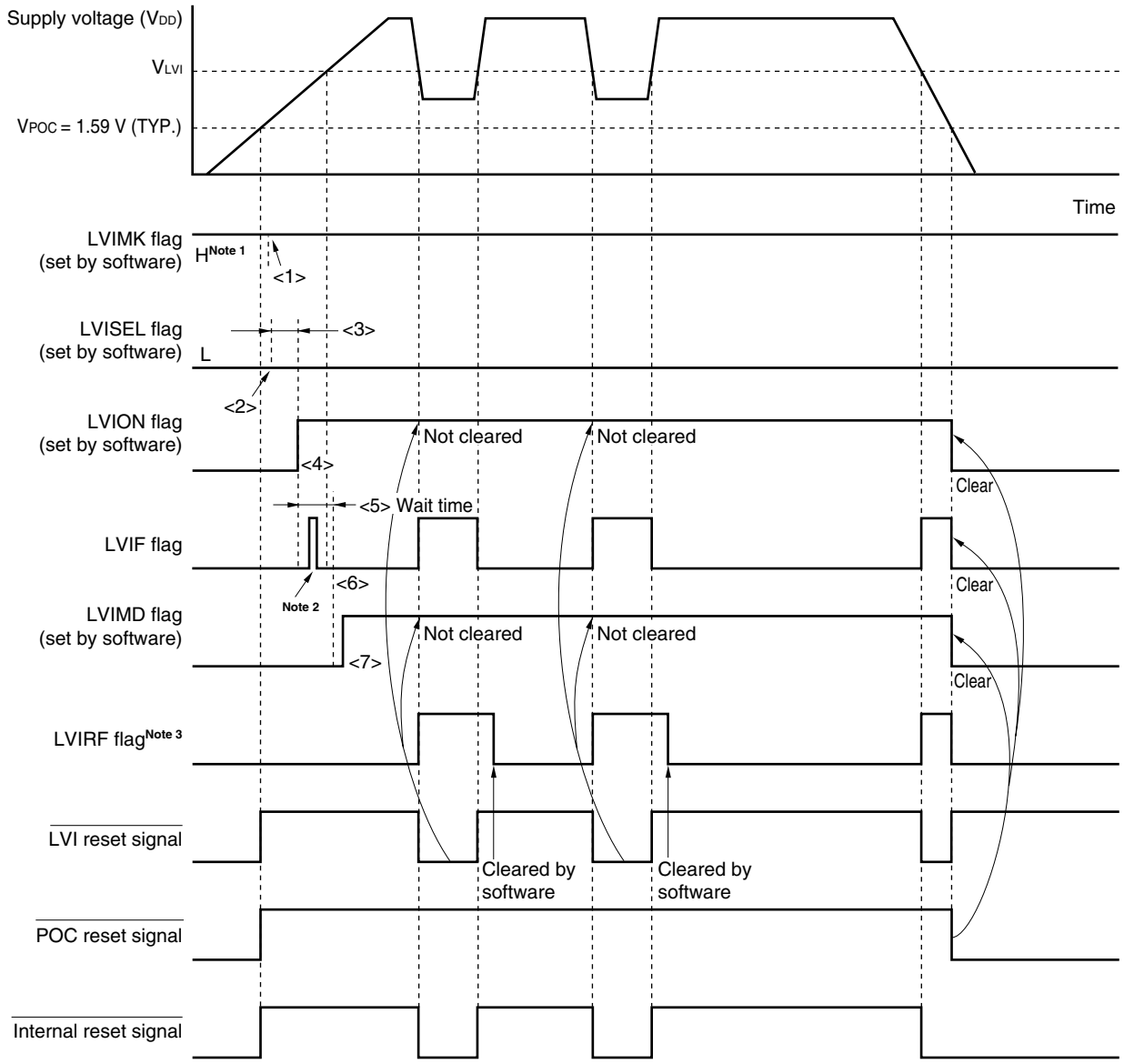
- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVIMD to 0 and then LVION to 0.

Figure 25-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

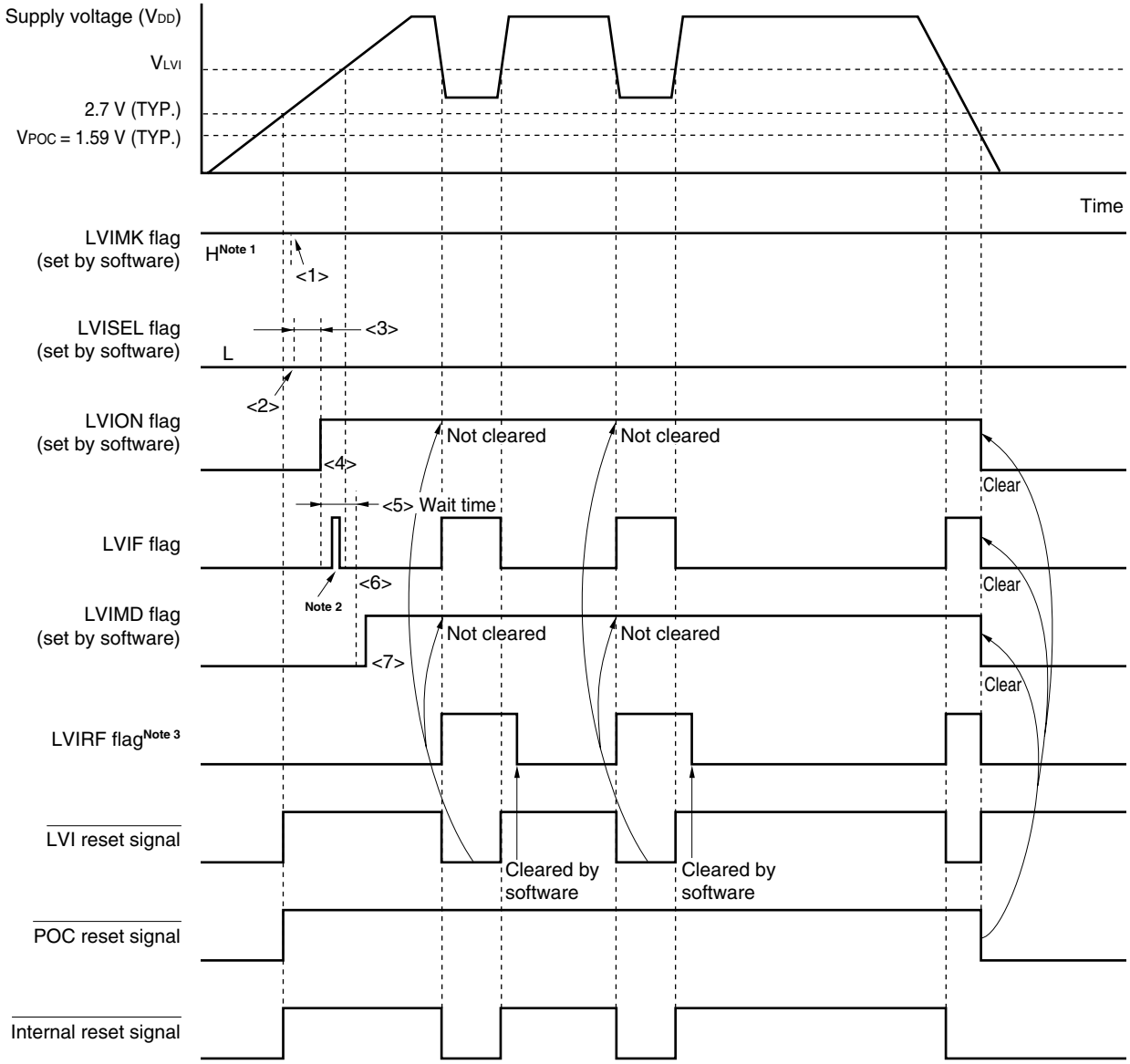
2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

Remark <1> to <7> in Figure 25-5 above correspond to <1> to <7> in the description of "When starting operation" in **25.4.1 (1) When detecting level of supply voltage (V_{DD})**.

Figure 25-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. 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 23 RESET FUNCTION**.

Remark <1> to <7> in Figure 25-5 above correspond to <1> to <7> in the description of "When starting operation" in **25.4.1 (1) When detecting level of supply voltage (V_{DD})**.

(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 an operation stabilization time (10 μs (MAX.)).
  - <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 25-6 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.

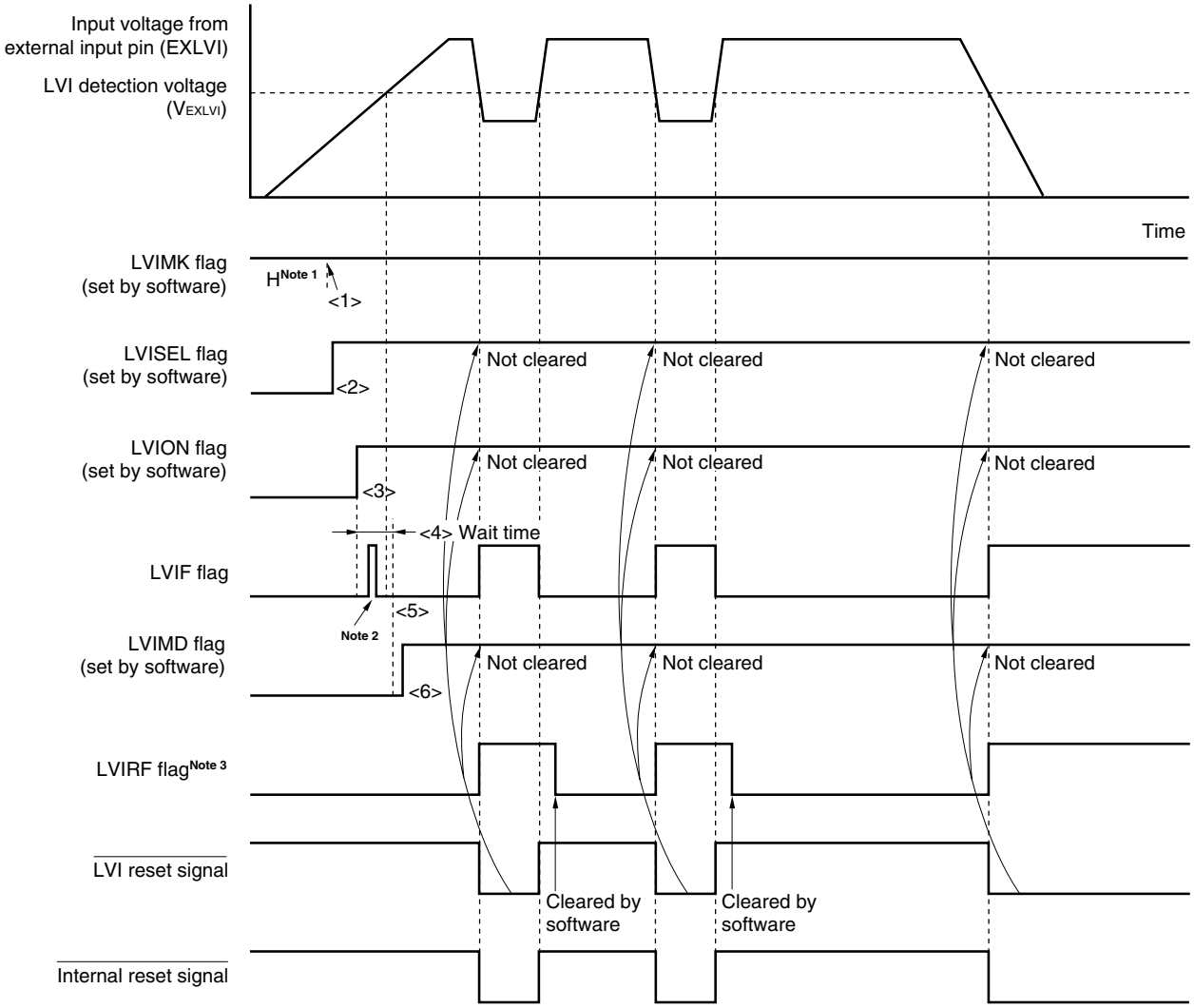
  Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.
  2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.
  3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVIMD to 0 and then LVION to 0.

Figure 25-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. 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 23 RESET FUNCTION**.

Remark <1> to <6> in Figure 25-6 above correspond to <1> to <6> in the description of "When starting operation" in **25.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

25.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

- 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 an operation stabilization time (10 μs (MAX.)).
  - <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> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <10> Execute the EI instruction (when vector interrupts are used).

  Figure 25-7 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <9> above.

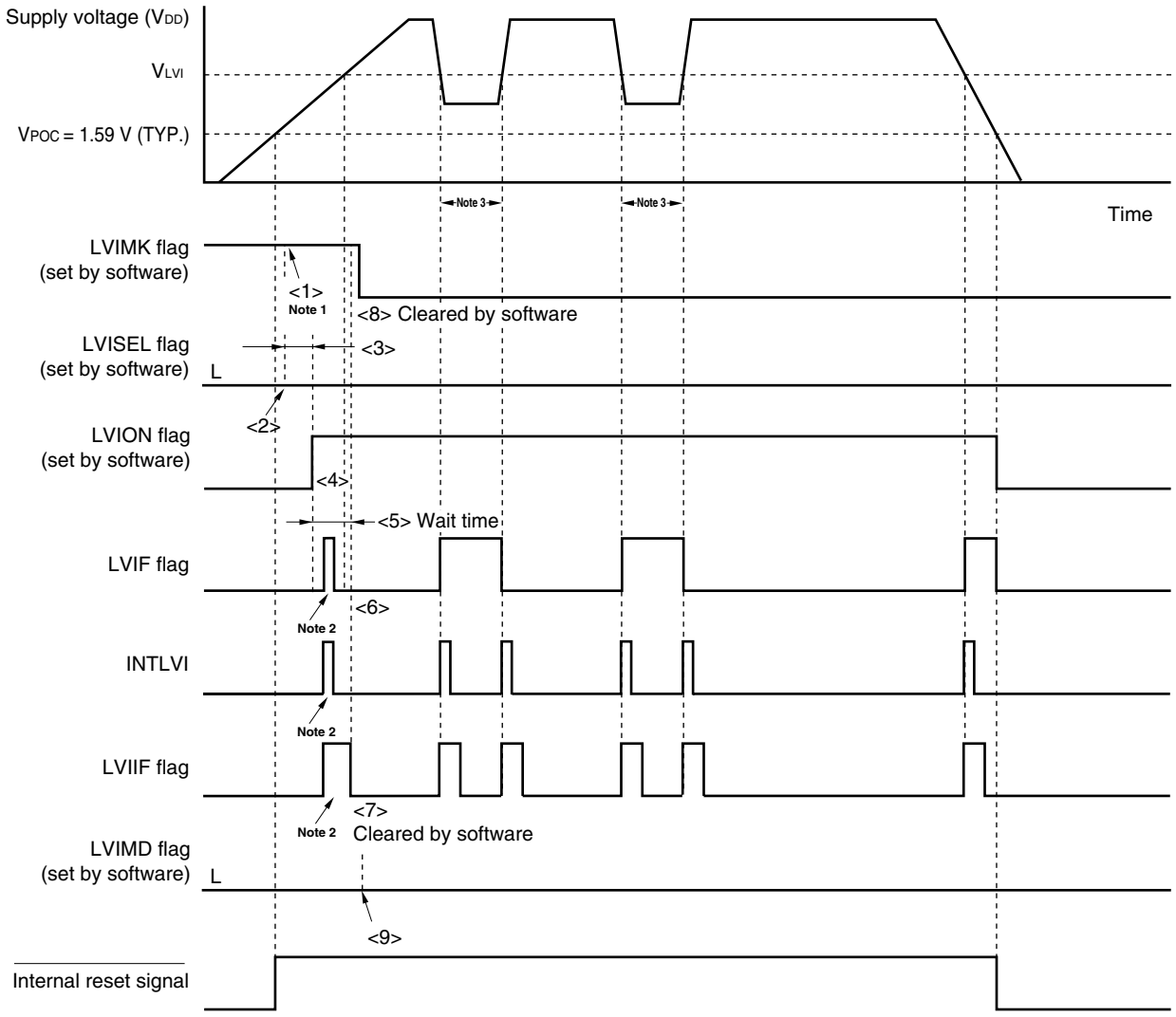
- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 25-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

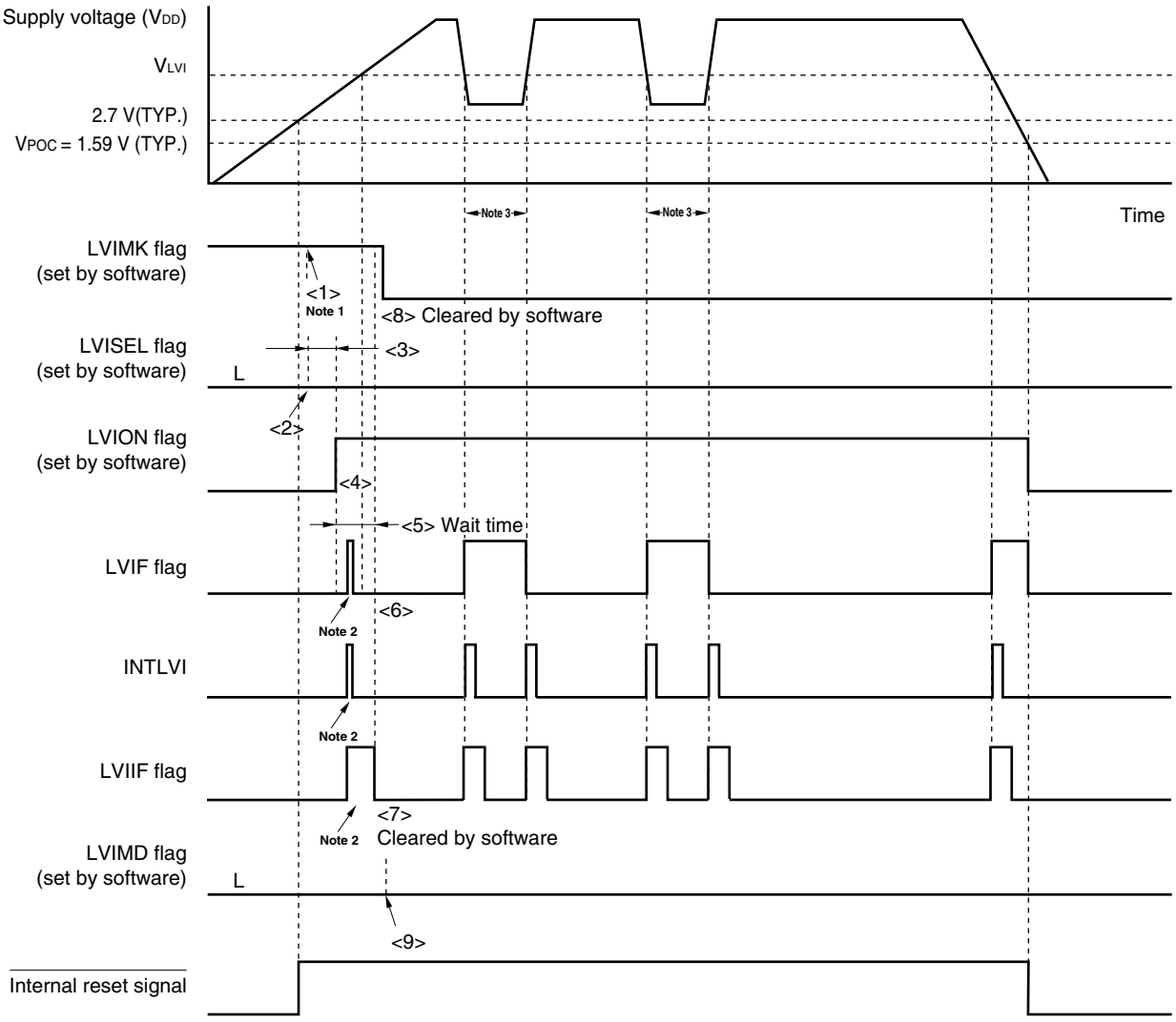
2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in Figure 25-7 above correspond to <1> to <9> in the description of "When starting operation" in **25.4.2 (1) When detecting level of supply voltage (V_{DD})**.

Figure 25-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 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 LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in Figure 25-7 above correspond to <1> to <9> in the description of "When starting operation" in **25.4.2 (1) When detecting level of supply voltage (V_{DD})**.

(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 an operation stabilization time (10 μs (MAX.)).
  - <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> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <9> Execute the EI instruction (when vector interrupts are used).

  Figure 25-8 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.

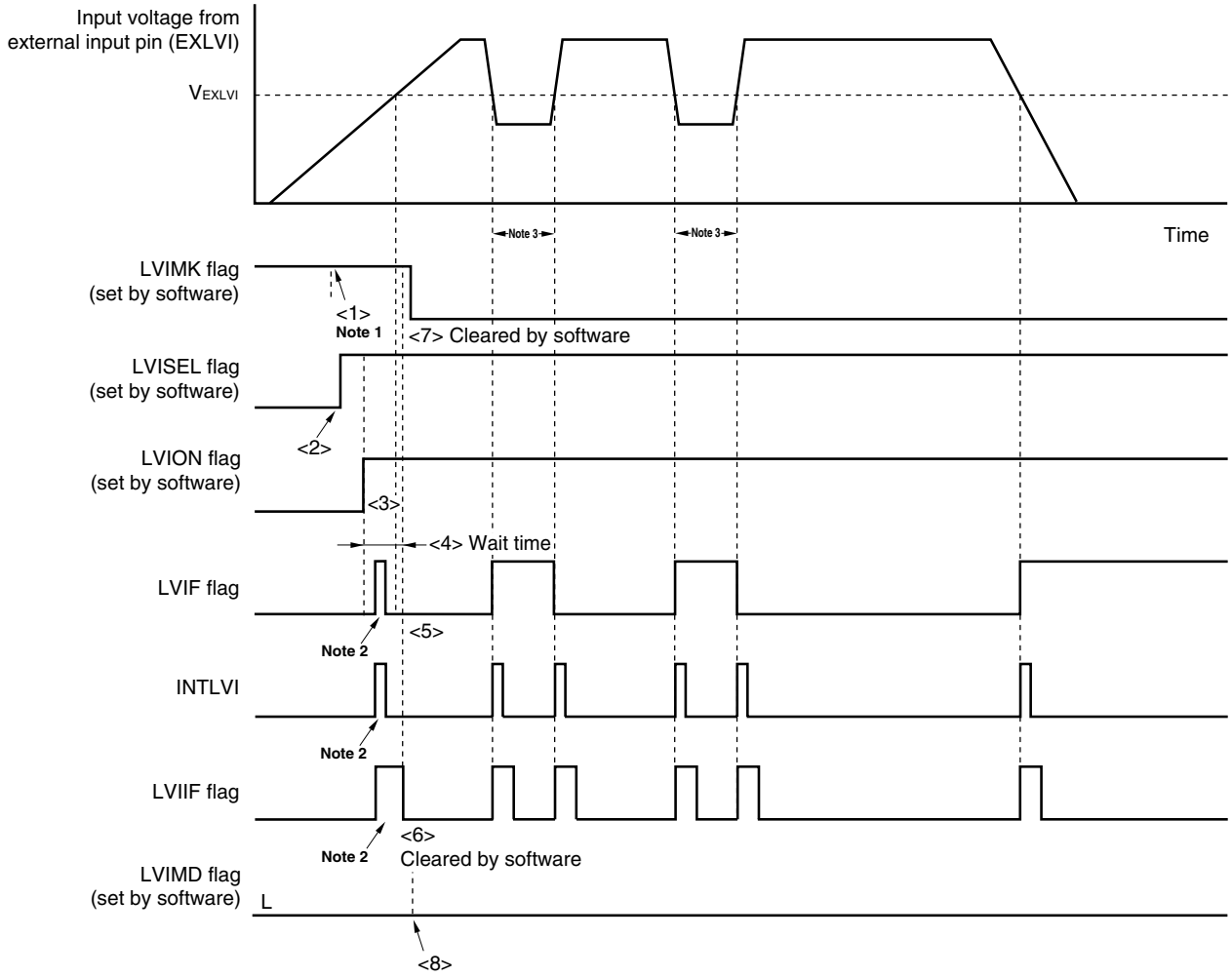
  Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 25-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



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 LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <8> in Figure 25-8 above correspond to <1> to <8> in the description of “When starting operation” in **25.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

25.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

(1) When used as reset

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 action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take (b) of action (2) below.

<Action>

(1) When used as reset

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 25-9**).

(2) When used as interrupt

(a) 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 0 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

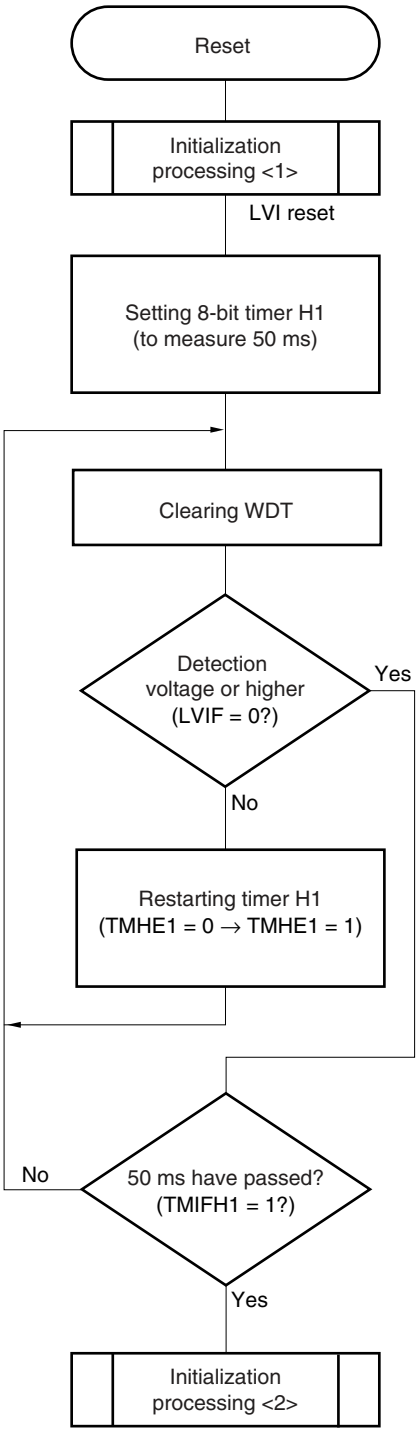
(b) In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, using the LVIF flag, and clear the LVIIF flag to 0.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 25-9. Example of Software Processing After Reset Release (1/2)

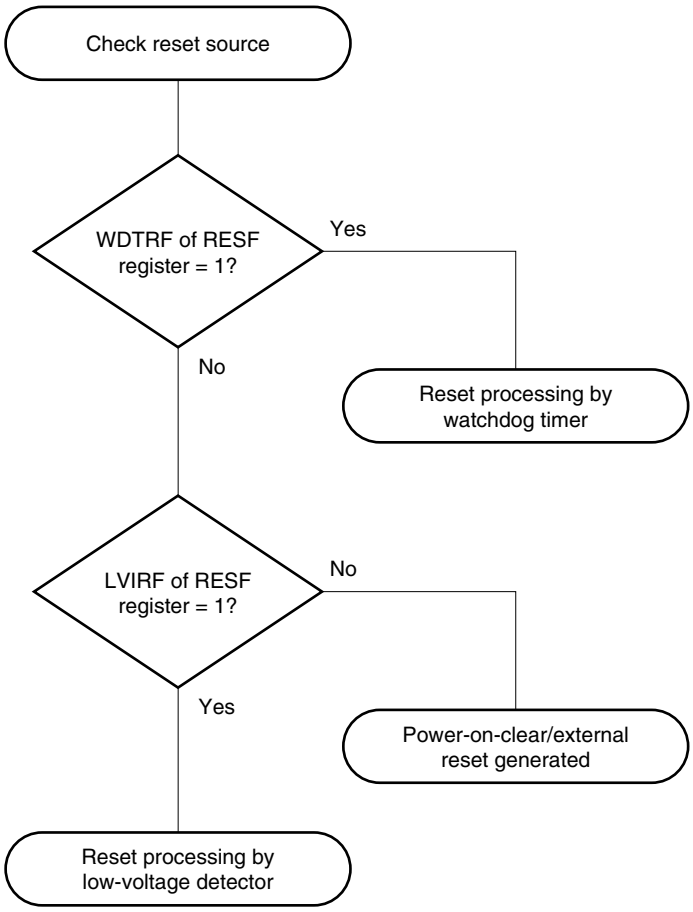
• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

Figure 25-9. Example of Software Processing After Reset Release (2/2)

• Checking reset source



CHAPTER 26 OPTION BYTE

26.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the 78K0/Kx2-C is an option byte area. When power is turned on or when the device is restarted from the reset status, the device automatically references the option bytes and sets specified functions. When using the product, be sure to set the following functions by using the option bytes.

When the boot swap operation is used during self-programming, 0080H to 0084H are switched to 1080H to 1084H. Therefore, set values that are the same as those of 0080H to 0084H to 1080H to 1084H in advance.

Caution Be sure to set 00H to 0082H and 0083H (0082H/1082H and 0083H/1083H when the boot swap function is used).

(1) 0080H/1080H

- Internal low-speed oscillator operation
  - Can be stopped by software
  - Cannot be stopped
- Watchdog timer interval time setting
- Watchdog timer counter operation
  - Enabled counter operation
  - Disabled counter operation
- Watchdog timer window open period setting

Caution Set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.

(2) 0081H/1081H

- Selecting POC mode
  - During 2.7 V/1.59 V POC mode operation (POCMODE = 1)
    The device is in the reset state upon power application and until the supply voltage reaches 2.7 V (TYP.). It is released from the reset state when the voltage exceeds 2.7 V (TYP.). After that, POC is not detected at 2.7 V but is detected at 1.59 V (TYP.).
    If the supply voltage rises to 1.8 V after power application at a pace slower than 0.5 V/ms (MIN.), use of the 2.7 V/1.59 V POC mode is recommended.
  - During 1.59 V POC mode operation (POCMODE = 0)
    The device is in the reset state upon power application and until the supply voltage reaches 1.59 V (TYP.). It is released from the reset state when the voltage exceeds 1.59 V (TYP.). After that, POC is detected at 1.59 V (TYP.), in the same manner as on power application.

Caution POCMODE can only be written by using a dedicated flash memory programmer. It cannot be set during self-programming or boot swap operation during self-programming (at this time, 1.59 V POC mode (default) is set). However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

(3) 0084H/1084H

○ On-chip debug operation control

- Disabling on-chip debug operation
- Enabling on-chip debug operation and erasing data of the flash memory in case authentication of the on-chip debug security ID fails
- Enabling on-chip debug operation and not erasing data of the flash memory even in case authentication of the on-chip debug security ID fails

Caution To use the on-chip debug function, set 02H or 03H to 0084H. Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot operation.

26.2 Format of Option Byte

The format of the option byte is shown below.

Figure 26-1. Format of Option Byte (1/2)

Address: 0080H/1080H[Note]

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC

WINDOW1	WINDOW0	Watchdog timer window open period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter/illegal access detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

LSROSC	Internal low-speed oscillator operation
0	Can be stopped by software (stopped when 1 is written to bit 1 (LSRSTOP) of RCM register)
1	Cannot be stopped (not stopped even if 1 is written to LSRSTOP bit)

Note Set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

3. If LSROSC = 0 (oscillation can be stopped by software), the count clock is not supplied to the watchdog timer in the HALT and STOP modes, regardless of the setting of bit 1 (LSRSTOP) of the internal oscillation mode register (RCM).
When 8-bit timer H1 operates with the internal low-speed oscillation clock, the count clock is supplied to 8-bit timer H1 even in the HALT/STOP mode.

4. Be sure to clear bit 7 to 0.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

2. ( ): f_{RL} = 264 kHz (MAX.)

Figure 26-1. Format of Option Byte (2/2)

Address: 0081H/1081H[Notes 1, 2]

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

POCMODE	POC mode selection
0	1.59 V POC mode (default)
1	2.7 V/1.59 V POC mode

Notes 1. POCMODE can only be written by using a dedicated flash memory programmer. It cannot be set during self-programming or boot swap operation during self-programming (at this time, 1.59 V POC mode (default) is set). However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

2. To change the setting for the POC mode, set the value to 0081H again after batch erasure (chip erasure) of the flash memory. The setting cannot be changed after the memory of the specified block is erased.

Caution Be sure to clear bits 7 to 1 to "0".

Address: 0082H/1082H, 0083H/1083H[Note]

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

Note Be sure to set 00H to 0082H and 0083H, as these addresses are reserved areas. Also set 00H to 1082H and 1083H because 0082H and 0083H are switched with 1082H and 1083H when the boot swap operation is used.

Address: 0084H/1084H[Note]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	On-chip debug operation control
0	0	Operation disabled
0	1	Setting prohibited
1	0	Operation enabled. Does not erase data of the flash memory in case authentication of the on-chip debug security ID fails.
1	1	Operation enabled. Erases data of the flash memory in case authentication of the on-chip debug security ID fails.

Note To use the on-chip debug function, set 02H or 03H to 0084H. Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot swap operation.

Remark For the on-chip debug security ID, see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

Here is an example of description of the software for setting the option bytes.

```
OPT     CSEG   AT 0080H
OPTION: DB     30H      ; Enables watchdog timer operation (illegal access detection operation),
                        ; Window open period of watchdog timer: 50%,
                        ; Overflow time of watchdog timer: 2^10/fRL,
                        ; Internal low-speed oscillator can be stopped by software.
        DB     00H      ; 1.59 V POC mode
        DB     00H      ; Reserved area
        DB     00H      ; Reserved area
        DB     00H      ; On-chip debug operation disabled
```

Remark Referencing of the option byte is performed during reset processing. For the reset processing timing, see **CHAPTER 23 RESET FUNCTION**.

CHAPTER 27 FLASH MEMORY

The 78K0/Kx2-C microcontrollers incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

27.1 Internal Memory Size Switching Register

Select the internal memory capacity using the internal memory size switching register (IMS).
IMS is set by an 8-bit memory manipulation instruction.
Reset signal generation sets IMS to CFH.

Caution Be sure to set each product to the values shown in Table 27-1 after a reset release.

Figure 27-1. Format of Internal Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
1	0	0	0	32 KB
1	1	0	0	48 KB
1	1	1	1	60 KB
Other than above				Setting prohibited

Caution To set the memory size, set IMS and then IXS. Set the memory size so that the internal ROM and internal expansion RAM areas do not overlap.

Table 27-1. Internal Memory Size Switching Register Settings

78K0/KC2-C	78K0/KE2-C	IMS
μPD78F0760	μPD78F0763	C8H
μPD78F0761	μPD78F0764	CCH
μPD78F0762	μPD78F0765	CFH

Remark The ROM and RAM capacities of the products with the on-chip debug function can be debugged according to the debug target products using IMS register. Set IMS according to the debug target products.

27.2 Internal Expansion RAM Size Switching Register

Select the internal expansion RAM capacity using the internal expansion RAM size switching register (IXS).
IXS is set by an 8-bit memory manipulation instruction.
Reset signal generation sets IXS to 0CH.

Cautions 1. Be sure to set each product to the values shown in Table 27-2 after a reset release.
2. A product that does not have an internal expansion RAM is not provided with IXS.

Figure 27-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	0	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
1	1	0	0	0 bytes
1	0	1	0	1024 bytes
1	0	0	0	2048 bytes
Other than above				Setting prohibited

Caution To set memory size, set IMS and then IXS. Set memory size so that the internal ROM area and internal expansion RAM area do not overlap.

Table 27-2. Internal Expansion RAM Size Switching Register (IXS) Settings

78K0/KC2-C	78K0/KE2-C	IXS
μPD78F0760	μPD78F0763	0CH
μPD78F0761	μPD78F0764	0AH
μPD78F0762	μPD78F0765	08H

Note A product that does not have an internal expansion RAM is not provided with IXS.

Remark The internal expansion RAM capacity of the products with the on-chip debug function can be debugged according to the debug target products using IXS register. Set IXS according to the debug target products.

27.3 Writing with Flash Memory Programmer

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the 78K0/Kx2-C microcontrollers have been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

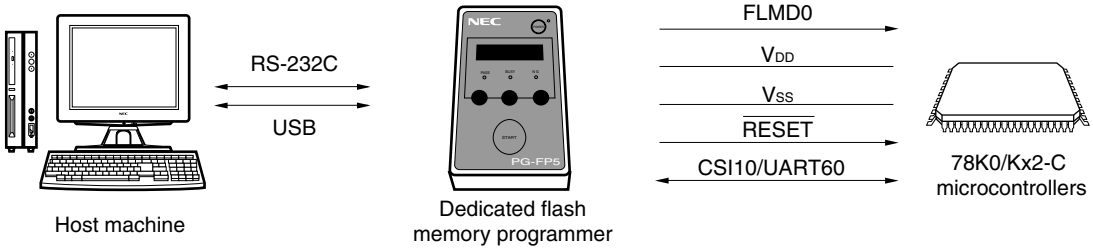
Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0/Kx2-C microcontrollers are mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

27.4 Programming Environment

The environment required for writing a program to the flash memory of the 78K0/Kx2-C microcontrollers are illustrated below.

Figure 27-3. 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 78K0/Kx2-C microcontrollers, CSI10 or UART60 is used for manipulation such as writing and erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

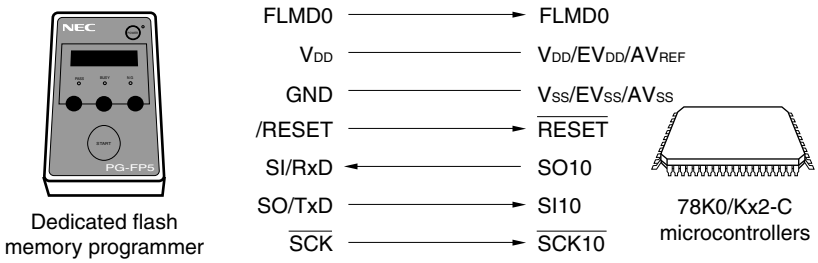
27.5 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0/Kx2-C microcontrollers is established by serial communication via CSI10 or UART60 of the 78K0/Kx2-C microcontrollers.

(1) CSI10

Transfer rate: 2.4 kHz to 2.5 MHz

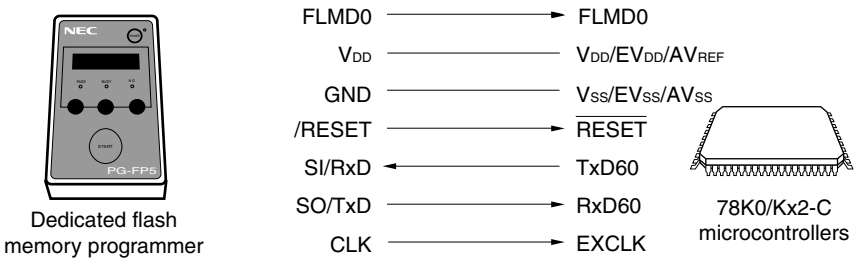
Figure 27-4. Communication with Dedicated Flash memory programmer (CSI10)



(2) UART60

Transfer rate: 115200 bps

Figure 27-5. Communication with Dedicated Flash memory programmer (UART60)



Remark With 78K0/KC2-C without an EV_{SS} pin, connect them to V_{SS}. With 78K0/KC2-C without an EV_{DD} pin, connect them to V_{DD}.

The dedicated flash memory programmer generates the following signals for the 78K0/Kx2-C microcontrollers. For details, refer to the user's manual for the PG-FP5 or FL-PR5.

Table 27-3. Pin Connection

Dedicated Flash memory programmer			78K0/KC2-C microcontrollers	78K0/KE2-C microcontrollers	Connection	
Signal Name	I/O	Pin Function	Pin Name	Pin Name	CSI10	UART60
FLMD0	Output	Mode signal	FLMD0	FLMD0	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, AV_{REF}	V_{DD}, EV_{DD}, AV_{REF}	◎	◎
GND	–	Ground	V_{SS}, AV_{SS}	V_{SS}, EV_{SS}, AV_{SS}	◎	◎
CLK	Output	Clock output to 78K0/Kx2-C microcontrollers	EXCLK/X2/P122	EXCLK/X2/P122	×Note 1	○Note 2
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	◎	◎
SI/RxD	Input	Receive signal	SO10/TxD60	SO10/TxD60	◎	◎
SO/TxD	Output	Transmit signal	SI10/RxD60	SI10/RxD60	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	$\overline{\text{SCK10}}$	◎	×

Notes 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

2. Only the X1 clock (f_X) or external main system clock (f_{EXCLK}) can be used when UART60 is used.

Remarks 1. ◎: Be sure to connect the pin.

○: The pin does not have to be connected if the signal is generated on the target board.

×: The pin does not have to be connected.

2. For the pins not to be connected the dedicated flash memory programmer, it is recommended to perform the processing described under the "Recommended Connection of Unused Pins" shown in **Table 2-3. Pin I/O Circuit Types (78K0/KC2-C)** or **Table 2-4. Pin I/O Circuit Types (78K0/KE2-C)**.

27.6 Connection of Pins on Board

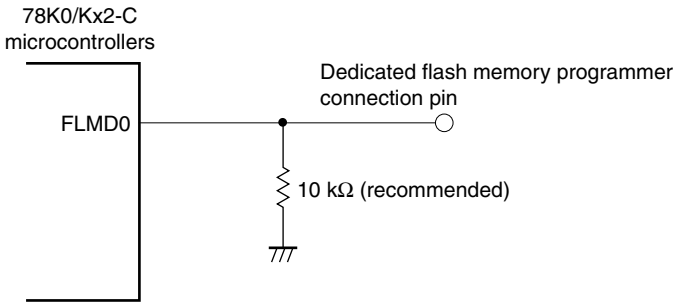
To write the flash memory on-board, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

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.

27.6.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. An FLMD0 pin connection example is shown below.

Figure 27-6. FLMD0 Pin Connection Example



27.6.2 Serial interface pins

The pins used by each serial interface are listed below.

Table 27-4. Pins Used by Each Serial Interface

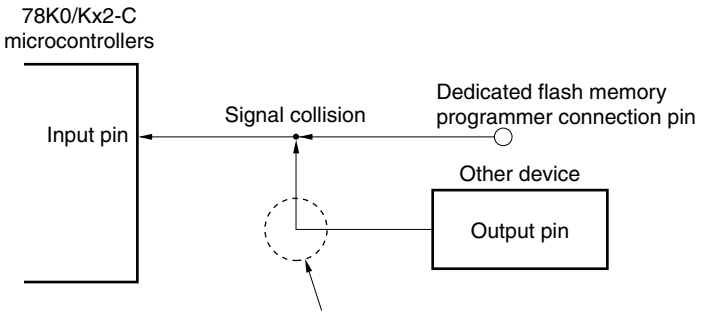
Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{SCK10}$
UART60	TxD60, RxD60

To connect the dedicated flash memory programmer to the pins of a serial interface that is connected to another device on the board, care must be exercised so that signals do not collide or that the other device does not malfunction.

(1) Signal collision

If the dedicated flash memory programmer (output) is connected to a pin (input) of a serial interface connected to another device (output), signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into an output high-impedance state.

Figure 27-7. Signal Collision (Input Pin of Serial Interface)

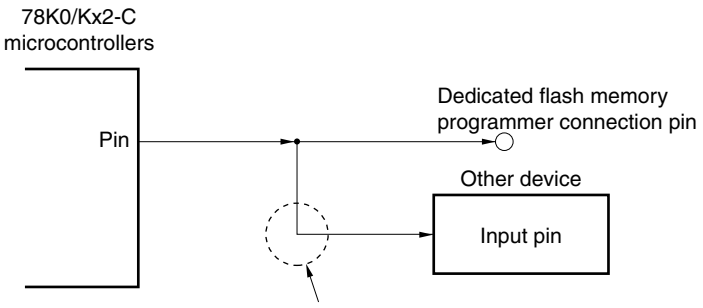


In the flash memory programming mode, the signal output by the device collides with the signal sent from the dedicated flash programmer. Therefore, isolate the signal of the other device.

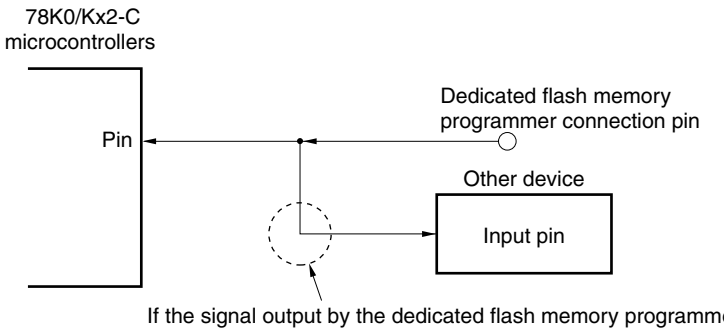
(2) Malfunction of other device

If the dedicated flash memory programmer (output or input) is connected to a pin (input or output) of a serial interface connected to another device (input), a signal may be output to the other device, causing the device to malfunction. To avoid this malfunction, isolate the connection with the other device.

Figure 27-8. Malfunction of Other Device



If the signal output by the 78K0/Kx2-C microcontrollers in the flash memory programming mode affects the other device, isolate the signal of the other device.



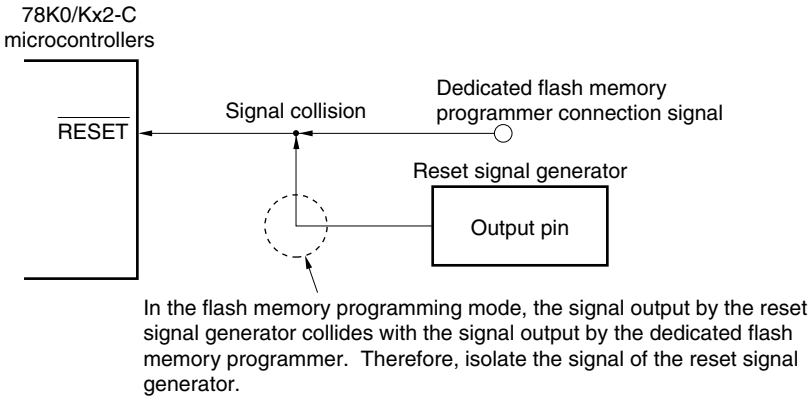
If the signal output by the dedicated flash memory programmer in the flash memory programming mode affects the other device, isolate the signal of the other device.

27.6.3 $\overline{\text{RESET}}$ pin

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, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 27-9. Signal Collision ($\overline{\text{RESET}}$ Pin)



27.6.4 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 EV_{DD}[Note] or EV_{SS}[Note] via a resistor.

Note With products without an EV_{SS} pin, connect them to V_{SS}. With products without an EV_{DD} pin, connect them to V_{DD}.

27.6.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF: recommended) in the same manner as during normal operation.

27.6.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode when using the on-board clock.

To input the operating clock from the dedicated flash memory programmer, however, connect CLK of the programmer to EXCLK/X2/P122.

Cautions 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

2. Only the X1 clock (f_X) or external main system clock (f_{EXCLK}) can be used when UART60 is used.

3. For the product with an on-chip debug function, connect P31/INTP2/OCD1A and P121/X1/OCD0A as follows when writing the flash memory with a flash memory programmer.

- **P31/INTP2/OCD1A: Connect to EV_{SS}[Note] via a resistor.**
- **P121/X1/OCD0A: Connect to V_{SS} via a resistor.**

Note With products without an EV_{SS} pin, connect them to V_{SS}.

27.6.7 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, 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, even when using the on-board supply voltage.

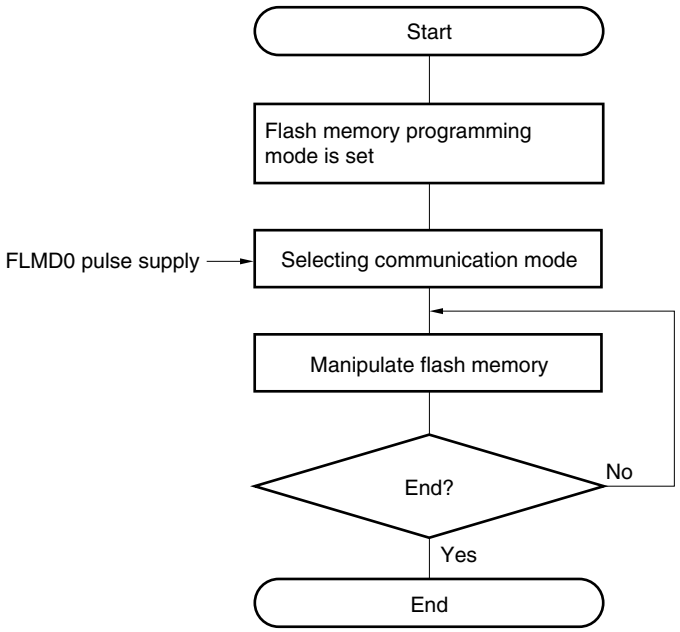
Supply the same other power supplies (EV_{DD}, EV_{SS}, AV_{REF}, and AV_{SS}) as those in the normal operation mode.

27.7 Programming Method

27.7.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 27-10. Flash Memory Manipulation Procedure



27.7.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0/Kx2-C microcontrollers in the flash memory programming mode. To set the mode, set the FLMD0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 27-11. Flash Memory Programming Mode

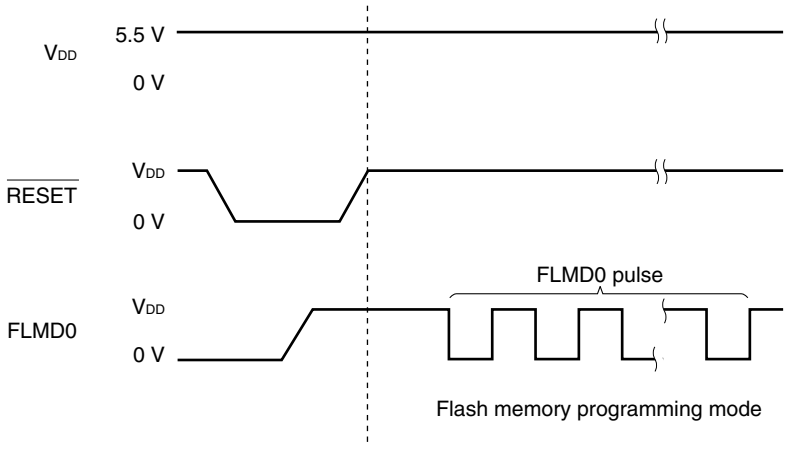


Table 27-5. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0 V	Normal operation mode
V_{DD}	Flash memory programming mode

27.7.3 Selecting communication mode

In the 78K0/Kx2-C microcontrollers, a communication mode is selected by inputting pulses to the FLMD0 pin after the dedicated flash memory programming mode is entered. These FLMD0 pulses are generated by the flash memory programmer.

The following table shows the relationship between the number of pulses and communication modes.

Table 27-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used	Peripheral Clock	Number of FLMD0 Pulses
	Port	Speed	Frequency	Multiply Rate			
UART (UART60)	UART-Ext-Osc	115,200 bps[Note 3]	2 to 20 MHz[Note 2]	1.0	TxD60, RxD60	f_X	0
	UART-Ext-FP5CK					f_{EXCLK}	3
3-wire serial I/O (CSI10)	CSI-Internal-OSC	2.4 kHz to 2.5 MHz	–		SO10, SI10, $\overline{SCK10}$	f_{RH}	8

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. The possible setting range differs depending on the voltage. For details, refer to the chapter of electrical specifications.

3. 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.

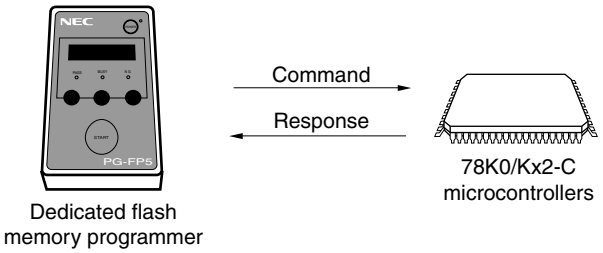
Caution When UART60 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash memory programmer after the FLMD0 pulse has been received.

Remark f_X: X1 clock
f_{EXCLK}: External main system clock
f_{RH}: Internal high-speed oscillation clock

27.7.4 Communication commands

The 78K0/Kx2-C microcontrollers communicate with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0/Kx2-C microcontrollers are called commands, and the signals sent from the 78K0/Kx2-C microcontrollers to the dedicated flash memory programmer are called response.

Figure 27-12. Communication Commands



The flash memory control commands of the 78K0/Kx2-C microcontrollers are listed in the table below. All these commands are issued from the programmer and the 78K0/Kx2-C microcontrollers perform processing corresponding to the respective commands.

Table 27-7. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	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	Status	Gets the current operating status (status data).
	Silicon Signature	Gets 78K0/Kx2-C information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0/Kx2-C version and 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.
	Oscillating Frequency Set	Specifies an oscillation frequency.

The 78K0/Kx2-C microcontrollers return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0/Kx2-C microcontrollers are listed below.

Table 27-8. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

27.8 Security Settings

The 78K0/Kx2-C microcontrollers support a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (0000H to 0FFFH) in the flash memory is prohibited by this setting. Execution of the batch erase (chip erase) command is also prohibited by this setting.

  Caution If a security setting that rewrites boot cluster 0 has been applied, the rewriting of boot cluster 0 and the batch erase (chip erase) will not be executed for the device.

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.

Prohibition of erasing blocks and writing is cleared by executing the batch erase (chip erase) command.

Table 27-9 shows the relationship between the erase and write commands when the 78K0/Kx2-C microcontroller security function is enabled.

Table 27-9. 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.

Table 27-10 shows how to perform security settings in each programming mode.

Table 27-10. 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		Cannot be disabled after set.

27.9 Flash Memory Programming by Self-Programming

The 78K0/Kx2-C microcontrollers support a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using a 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. To execute interrupt servicing, restore the normal operation mode after self-programming has been stopped, and execute the EI instruction. After the self-programming mode is later restored, self-programming can be resumed.

Remark For details of the self-programming function and the self-programming library, refer to **78K0 Microcontrollers Self Programming Library Type01 User's Manual (U18274E)**.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. Oscillation of the internal high-speed oscillator is started during self programming, regardless of the setting of the RSTOP flag (bit 0 of the internal oscillation mode register (RCM)). Oscillation of the internal high-speed oscillator cannot be stopped even if the STOP instruction is executed.

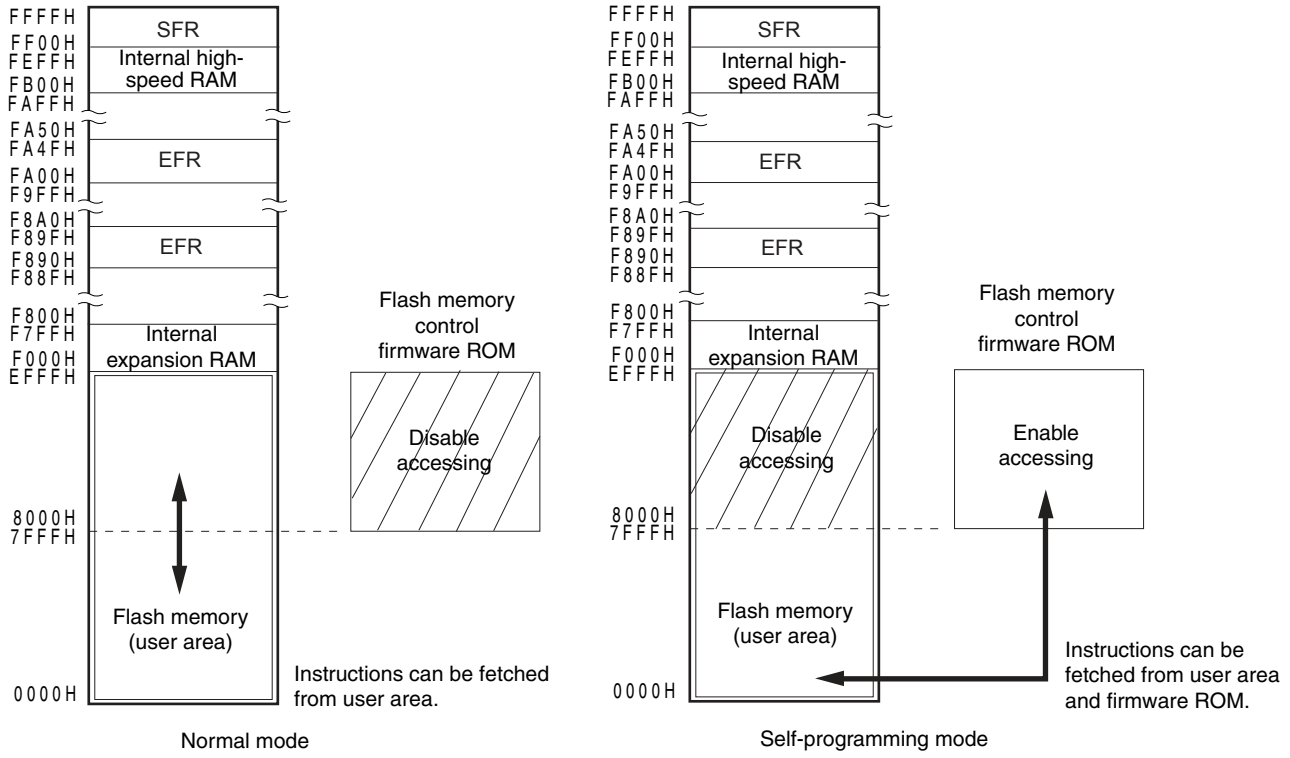
3. Input a high level to the FLMD0 pin during self-programming.

4. Be sure to execute the DI instruction before starting self-programming.
The self-programming function checks the interrupt request flags (IF0L, IF0H, IF1L, and IF1H). If an interrupt request is generated, self-programming is stopped.

5. Self-programming is also stopped by an interrupt request that is not masked even in the DI status. To prevent this, mask the interrupt by using the interrupt mask flag registers (MK0L, MK0H, MK1L, and MK1H).

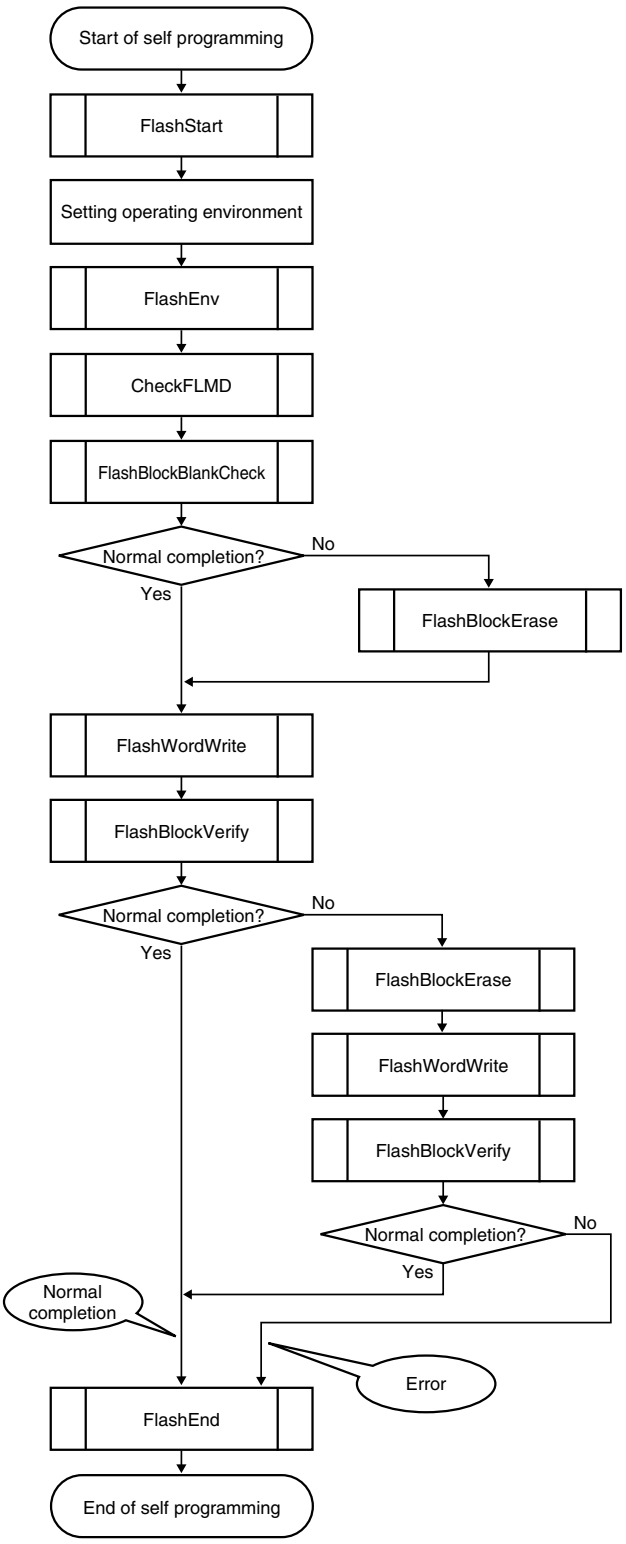
Caution 6. Allocate the entry program for self-programming in the common area of 0000H to 7FFFH.

Figure 27-13. Operation Mode and Memory Map for Self-Programming (μPD78F0762 and 78F0765)



The following figure illustrates a flow of rewriting the flash memory by using a self-programming library.

Figure 27-14. Flow of Self Programming (Rewriting Flash Memory)



Remark For details of the self-programming library, refer to **78K0 Microcontrollers Self Programming Library Type01 User's Manual (U18274E)**.

27.9.1 Boot swap function

If rewriting the boot area has failed during self-programming due to a power failure or some other cause, the data in the boot area may be lost and the program may not be restarted by resetting.

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 78K0/Kx2-C microcontrollers, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0.

As a result, even if a power failure occurs while the boot programming area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

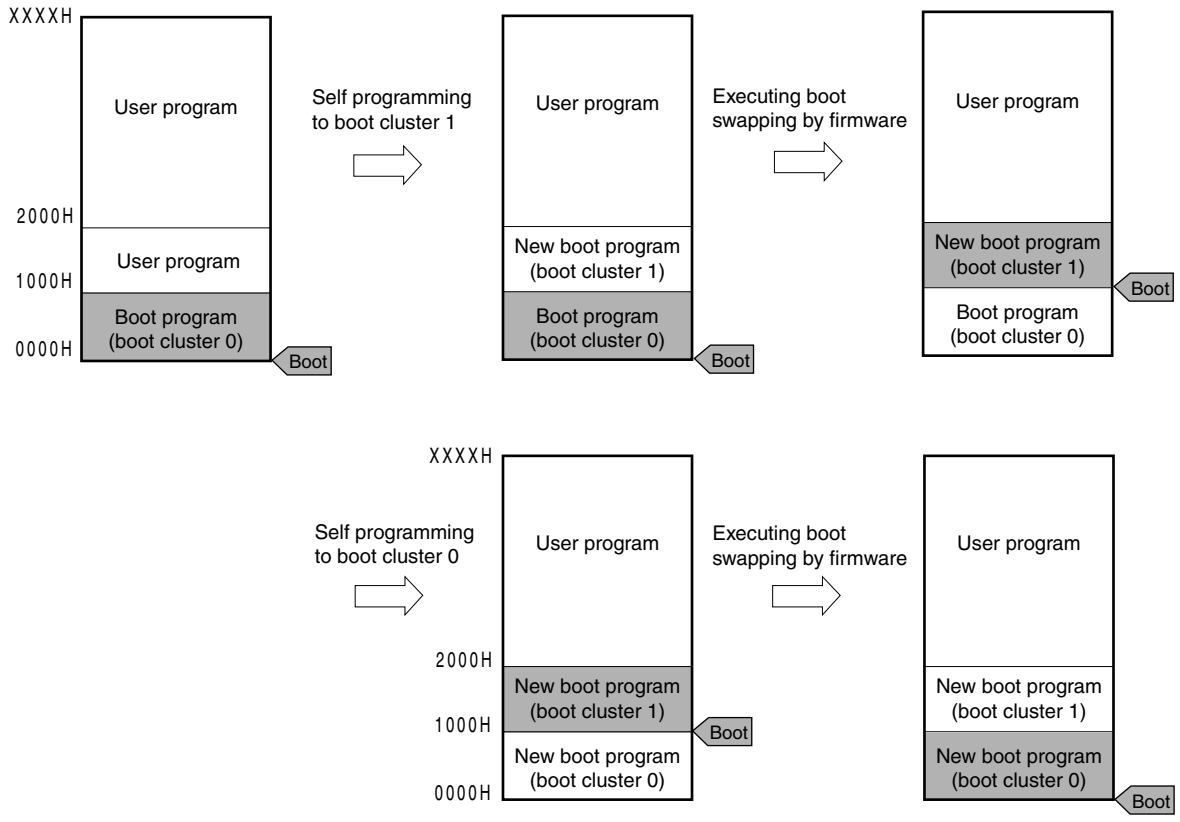
If the program has been correctly written to boot cluster 0, restore the original boot area by using the set information function of the firmware of the 78K0/Kx2-C microcontrollers.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Boot cluster 0 (0000H to 0FFFH): Original boot program area
Boot cluster 1 (1000H to 1FFFH): Area subject to boot swap function

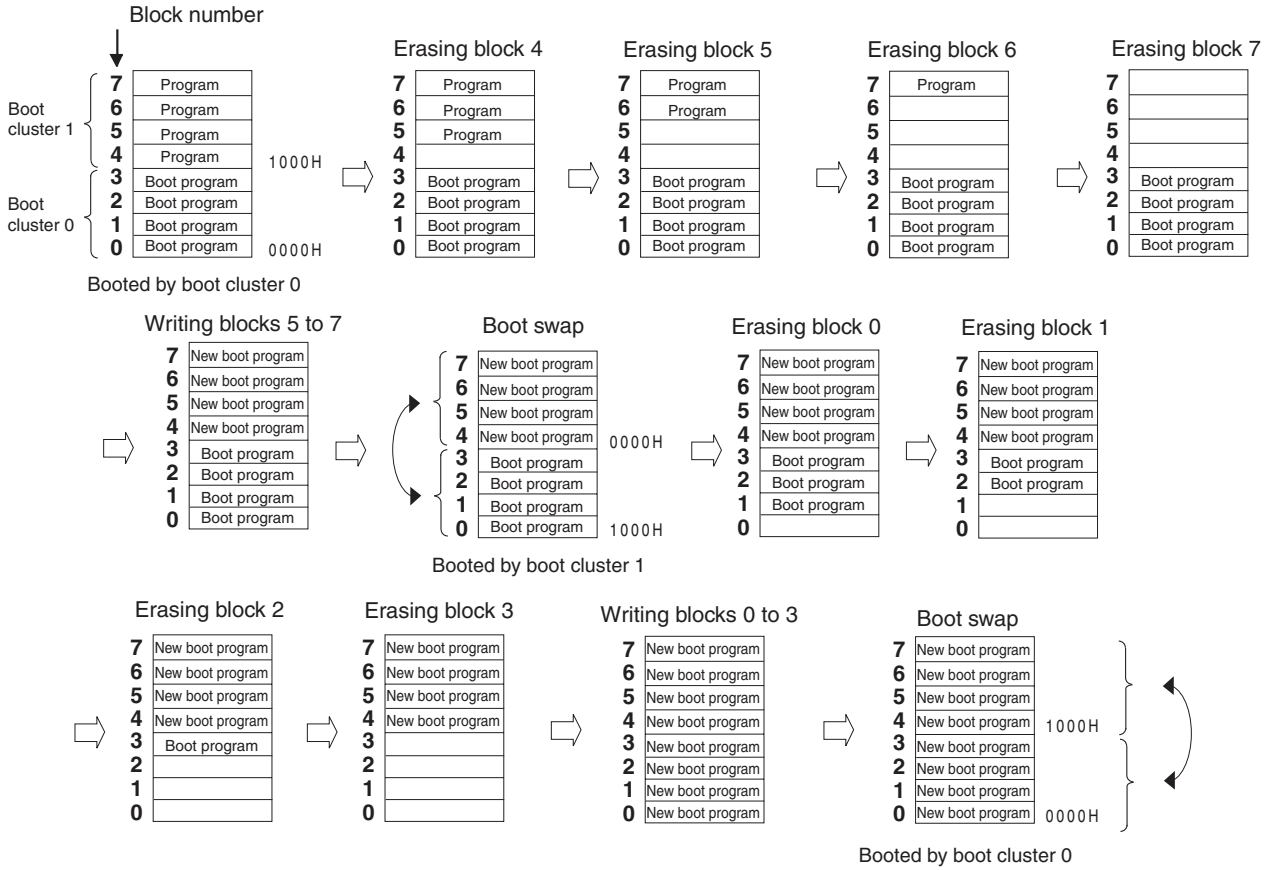
Caution When executing boot swapping, do not use the E.P.V command with the dedicated flash memory programmer.

Figure 27-15. Boot Swap Function



Remark Boot cluster 1 becomes 0000H to 0FFFH when a reset is generated after the boot flag has been set.

Figure 27-16. Example of Executing Boot Swapping



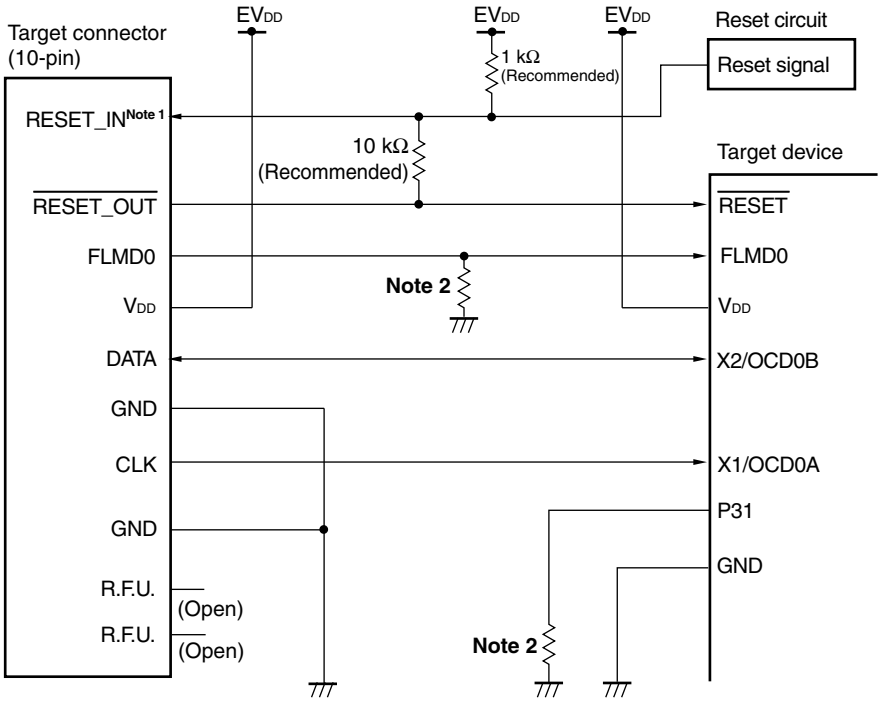
CHAPTER 28 ON-CHIP DEBUG FUNCTION

28.1 Connecting QB-MINI2 to 78K0/Kx2-C

The 78K0/Kx2-C uses the V_{DD}, FLMD0, $\overline{RESET}$, OCD0A/X1 (or OCD1A/P31), OCD0B/X2 (or OCD1B/P32), and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2). Whether OCD0A/X1 and OCD1A/P31, or OCD0B/X2 and OCD1B/P32 are used can be selected.

Caution The 78K0/Kx2-C 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 28-1. Connection Example of QB-MINI2 and 78K0/Kx2-C (When OCD0A/X1 and OCD0B/X2 Are Used)



Notes 1. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **QB-MINI2 User's Manual (U18371E)**.

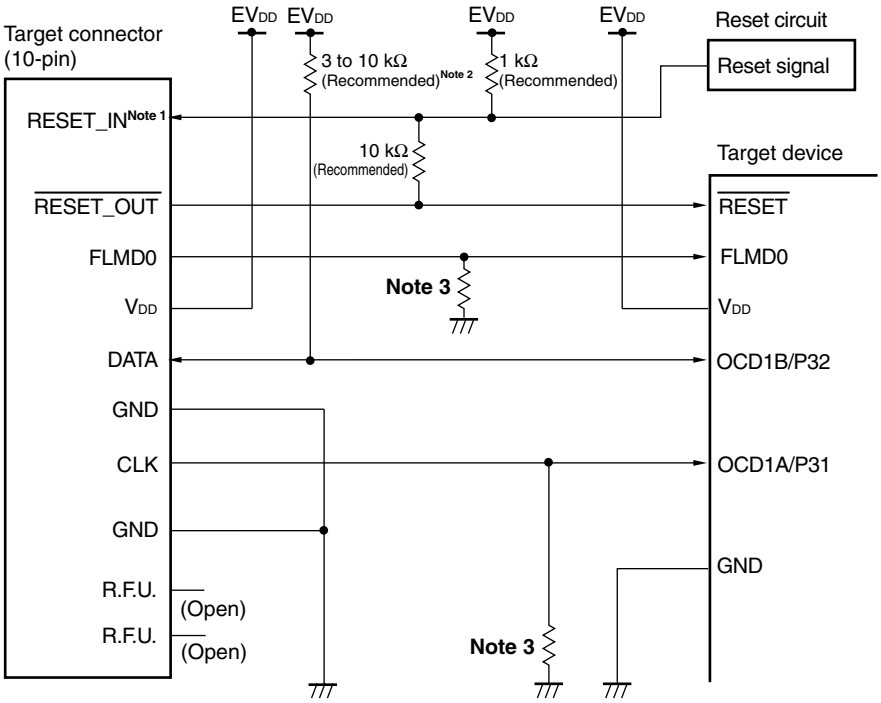
2. Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Cautions 1. Input the clock from the OCD0A/X1 pin during on-chip debugging.

2. Control the OCD0A/X1 and OCD0B/X2 pins by externally pulling down the OCD1A/P31 pin or by using an external circuit using the P130 pin (that outputs a low level when the device is reset).

Remark With 78K0/KC2-C not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Figure 28-2. Connection Example of QB-MINI2 and 78K0/Kx2-C (When OCD1A/P31 and OCD1B/P32 Are Used)



Notes 1. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **QB-MINI2 User's Manual (U18371E)**.

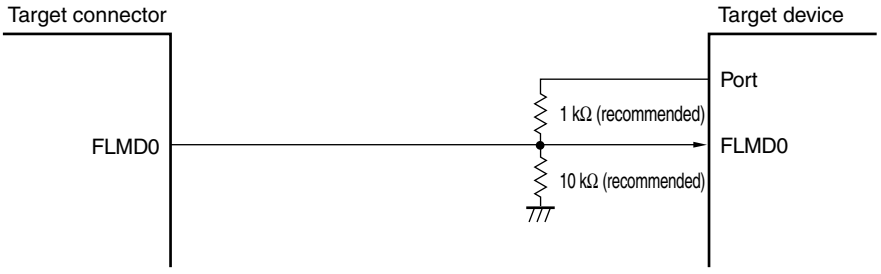
2. This is the processing of the pin when OCD1B/P32 is set as the input port (to prevent the pin from being left opened when not connected to QB-MINI2).

3. Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Remark With 78K0/KC2-C not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Connect the FLMD0 pin as follows when performing self programming by means of on-chip debugging.

Figure 28-3. Connection of FLMD0 Pin for Self Programming by Means of On-Chip Debugging



Caution When using the port that controls the FLMD0 pin, make sure that it satisfies the values of the high-level output current and FLMD0 supply voltage (minimum value: 0.8V_{DD}) stated in CHAPTER 30 ELECTRICAL SPECIFICATIONS.

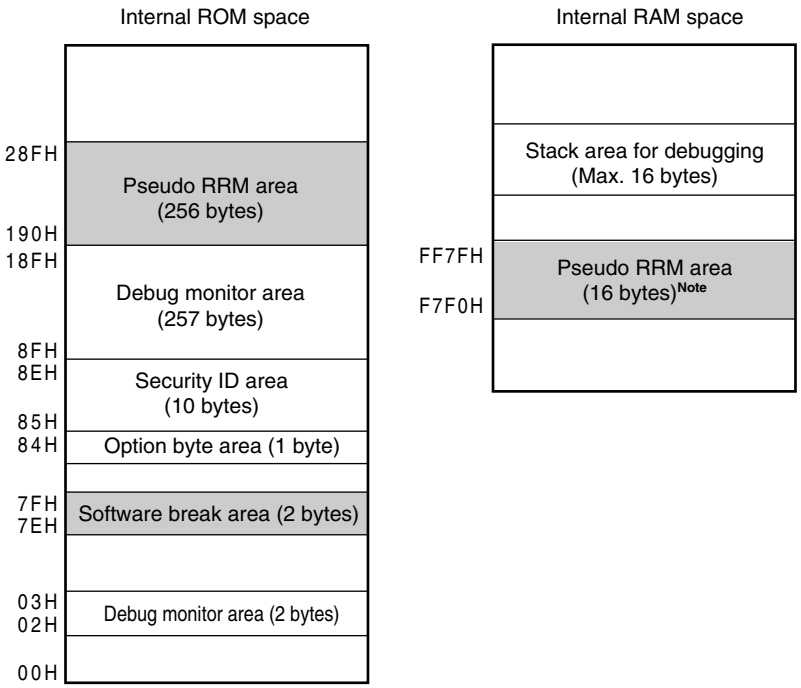
28.2 Reserved Area Used by QB-MINI2

QB-MINI2 uses the reserved areas shown in Figure 28-4 below to implement communication with the 78K0/Kx2-C, or each debug function. The shaded reserved areas are used for the respective debug functions to be used, and the other areas are always used for debugging. These reserved areas can be secured by using user programs and compiler options.

When using a boot swap operation during self programming, set the same value to boot cluster 1 beforehand.

For details on reserved area, refer to **QB-MINI2 User's Manual (U18371E)**.

Figure 28-4. Reserved Area Used by QB-MINI2



Note With products not incorporated the internal expansion RAM (μPD78F0760 and 78F0763), it is not necessary to secure this area.

Remark Shaded reserved areas: Area used for the respective debug functions to be used
Other reserved areas: Areas always used for debugging

CHAPTER 29 INSTRUCTION SET

This chapter lists each instruction set of the 78K0/Kx2 microcontrollers in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

29.1 Conventions Used in Operation List

29.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Uppercase letters and the symbols #, !, $ and [ ] are keywords and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register identifiers r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for specification.

Table 29-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol[Note]
sfrp	Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, see **Table 3-7 Special Function Register List**.

29.1.2 Description of operation column

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
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: Higher 8 bits and lower 8 bits of 16-bit register
∧: Logical product (AND)
∨: Logical sum (OR)
$\veebar$: Exclusive logical sum (exclusive OR)
$\overline{\ \ }$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

29.1.3 Description of flag operation column

(Blank): Not affected
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is restored

29.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9	A ← (addr16)	
		!addr16, A	3	8	9	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL + byte]	2	8	9	A ← (HL + byte)	
		[HL + byte], A	2	8	9	(HL + byte) ← A	
		A, [HL + B]	1	6	7	A ← (HL + B)	
		[HL + B], A	1	6	7	(HL + B) ← A	
		A, [HL + C]	1	6	7	A ← (HL + C)	
		[HL + C], A	1	6	7	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ (sfr)	
		A, !addr16	3	8	10	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10	A ↔ (HL + B)	
		A, [HL + C]	2	8	10	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
		!addr16, AX	3	10	12	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r	× × ×
		r, A	2	4	–	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	× × ×
		r, A	2	4	–	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + C	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	× × ×
		A, r Note 3	2	4	–	A, CY ← A – r	× × ×
		r, A	2	4	–	r, CY ← r – A	× × ×
		A, saddr	2	4	5	A, CY ← A – (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	× × ×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	× × ×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	× × ×
		r, A	2	4	–	r, CY ← r – A – CY	× × ×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	× × ×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	× × ×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	× × ×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×
		A, r Note 3	2	4	–	A ← A ∧ r	×
		r, A	2	4	–	r ← r ∧ A	×
		A, saddr	2	4	5	A ← A ∧ (saddr)	×
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×
		A, [HL]	1	4	5	A ← A ∧ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×
		A, r Note 3	2	4	–	A ← A ∨ r	×
		r, A	2	4	–	r ← r ∨ A	×
		A, saddr	2	4	5	A ← A ∨ (saddr)	×
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×
		A, [HL]	1	4	5	A ← A ∨ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ∨ (HL + C)	×
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×
		A, r Note 3	2	4	–	A ← A ⊻ r	×
		r, A	2	4	–	r ← r ⊻ A	×
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ⊻ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ⊻ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ⊻ (HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9	A – (addr16)	× × ×
		A, [HL]	1	4	5	A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit	×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit	×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit	×
		CY, PSW. bit	3	–	7	CY ← CY ⊻ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit	×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1	
		sfr.bit	3	–	8	sfr.bit ← 1	
		A.bit	2	4	–	A.bit ← 1	
		PSW.bit	2	–	6	PSW.bit ← 1	× × ×
		[HL].bit	2	6	8	(HL).bit ← 1	
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0	
		sfr.bit	3	–	8	sfr.bit ← 0	
		A.bit	2	4	–	A.bit ← 0	
		PSW.bit	2	–	6	PSW.bit ← 0	× × ×
		[HL].bit	2	6	8	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15 – 11 ← 00001, PC10 – 0 ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (addr5 + 1), PCL ← (addr5), SP ← SP – 2	
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)H, (SP – 3) ← (PC + 1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP – 3, IE ← 0	
	RET		1	6	–	PCH ← (SP + 1), PCL ← (SP), SP ← SP + 2	
	RETI		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
	RETB		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1	
		rp	1	4	–	(SP – 1) ← rpH, (SP – 2) ← rpL, SP ← SP – 2	
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R R R
		rp	1	4	–	rpH ← (SP + 1), rpL ← (SP), SP ← SP + 2	
	MOVW	SP, #word	4	–	10	SP ← word	
		SP, AX	2	–	8	SP ← AX	
		AX, SP	2	–	8	AX ← SP	
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16	
		$addr16	2	6	–	PC ← PC + 2 + jdisp8	
		AX	2	8	–	PCH ← A, PCL ← X	
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

29.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand / First Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except “r = A”

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand / First Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand / First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

CHAPTER 30 ELECTRICAL SPECIFICATIONS

Target Products 78K0/KC2-C: μPD78F0762, μPD78F0761, μPD78F0760
78K0/KE2-C: μPD78F0765, μPD78F0764, μPD78F0763

Cautions 1. The 78K0R/Kx2-C 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. The pins mounted depend on the product as follows.

(1) Port functions

Port	78K0/KC2-C	78K0/KE2-C
Port 0	P00, P01	P00 to P06
Port 1	P10 to P17	P10 to P17
Port 2	P20 to P27	P20 to P27
Port 3	P30 to P33	P30 to P33
Port 4	P40, P41	P40 to P43
Port 5	–	P50 to P53
Port 6	P60 to P63	P60 to P63
Port 7	P70 to P75	P70 to P77
Port 12	P120 to P124	P120 to P124
Port 13	P130	P130
Port 14	P140	P140, P141

(The remaining table is on the next page.)

(2) Non-port functions

Port		78K0/KC2-C	78K0/KE2-C
Power supply, ground		V_{DD}, AV_{REF}, V_{SS}, AV_{SS}	V_{DD}, EV_{DD}, V_{SS}, EV_{SS}, AV_{REF}, AV_{SS}
Regulator		REGC	REGC
Reset		$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
Clock oscillation		X1, X2, XT1,XT2, EXCLK	X1, X2, XT1,XT2, EXCLK
Real-time counter		RTC1HZ, RTCCL, RTCDIV	RTC1HZ, RTCCL, RTCDIV
Writing to flash memory		FLMD0	FLMD0
Interrupt		INTP0 to INTP6	INTP0 to INTP6
Key interrupt		KR0 to KR3	KR0 to KR7
Timer	TM00	TI000, TI010, TO00	TI000, TI001, TI002, TI010, TI011, TI012, TO00, TO01, TO02
	TM50	TI50, TO50	TI50, TO50
	TM51	TI51, TO51	TI51, TO51
	TMH0	TOH0	TOH0
	TMH1	TOH1	TOH1
Serial interface	UART0	RxD0, TxD0	RxD0, TxD0
	UART6	RxD60, TxD60	RxD60, RxD61, TxD60, TxD61
	IICA00, IICA01, IICA02	SDAA0 to SDAA2, SCLA0 to SCLA2	SDAA0 to SDAA2, SCLA0 to SCLA2
	CSI10	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK10}}$, SI10, SO10
	CSI11	$\overline{\text{SCK11}}$, SI11, SO11	$\overline{\text{SCK11}}$, SI11, SO11, $\overline{\text{SSI11}}$
Remote controller receiver		RIN, ROUT	RIN, ROUT
CEC transmission/reception		CECIN, CECIO, CECOUT	CECIN, CECIO, CECOUT
A/D converter		ANI0 to ANI7	ANI0 to ANI7
Clock output		PCL	PCL
Buzzer output		–	BUZ
Low-voltage detector (LVI)		EXLVI	EXLVI
On-chip debug function		OCD0A, OCD1A, OCD0B, OCD1B	OCD0A, OCD1A, OCD0B, OCD1B

<R>

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	EV_{DD}		−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	EV_{SS}		−0.5 to +0.3	V
	AV_{REF}		−0.5 to V_{DD} + 0.3[Note]	V
	AV_{SS}		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}		−0.5 to +3.6 and −0.5 to V_{DD}	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P20 to P27, P30 to P33, P40 to P43, P50 to P53, P120 to P124, P140, P141, X1, X2, XT1, XT2, $\overline{\text{RESET}}$, FLMD0	−0.3 to V_{DD} + 0.3[Note]	V
	V_{I2}	P60 to P63, P70 to P77 (N-ch open drain)	−0.3 to +6.5	V
Output voltage	V_O		−0.3 to V_{DD} + 0.3[Note]	V
Analog input voltage	V_{AN}	ANI0 to ANI7	−0.3 to AV_{REF} + 0.3[Note] and −0.3 to V_{DD} + 0.3[Note]	V

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P120 to P124, P130, P140, P141	−10	mA
		Total of all pins −80 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	−25	mA
			P05, P06, P10 to P17, P30 to P33, P50 to P53	−55	mA
	I_{OH2}	Per pin	P20 to P27	−0.5	mA
		Total of all pins		−2	mA
Output current, low	I_{OL1}	Per pin	P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P77, P120 to P124, P130, P140, P141	30	mA
		Total of all pins 200 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	60	mA
			P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P77,	140	mA
	I_{OL2}	Per pin	P20 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		−40 to +85	°C
		In flash memory programming mode		−40 to +85	
Storage temperature	T_{stg}			−65 to +150	°C

Cautions 1. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

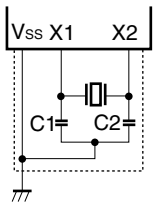
2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator, Crystal resonator		X1 clock oscillation frequency (f_X)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Internal Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

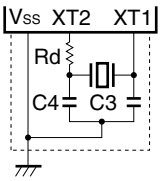
Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
8 MHz internal oscillator	Internal high-speed oscillation clock frequency (f_{RH})[Note]	2.5 V ≤ V_{DD} ≤ 5.5 V	7.6	8.0	8.4	MHz
		1.8 V ≤ V_{DD} < 2.5 V	6.75	8.0	8.4	
240 kHz internal oscillator	Internal low-speed oscillation clock frequency (f_{RL})	2.6 V ≤ V_{DD} ≤ 5.5 V	216	240	264	kHz
		1.8 V ≤ V_{DD} < 2.6 V	192	240	264	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Remark RSTS: Bit 7 of the internal oscillation mode register (RCM)

XT1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	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 figure to avoid an adverse effect from wiring capacitance.

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

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

<R>**Recommended oscillator circuit constants**

(1) X1 oscillation: Ceramic resonator (AMPH = 0, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd.	CSTCC2M00G56-R0	SMD	2.0	Internal (47)	Internal (47)	0	1.8	5.5
	CSTCR4M00G55-R0	SMD	4.0	Internal (39)	Internal (39)	0		
	CSTLS4M00G56-B0	Lead		Internal (47)	Internal (47)	0		
	CSTCR4M19G55-R0	SMD	4.194	Internal (39)	Internal (39)	0		
	CSTLS4M19G56-B0	Lead		Internal (47)	Internal (47)	0		
	CSTCR4M91G55-R0	SMD	4.915	Internal (39)	Internal (39)	0		
	CSTLS4M91G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR5M00G55-R0	SMD	5.0	Internal (39)	Internal (39)	0		
	CSTLS5M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR6M00G53-R0	SMD	6.0	Internal (15)	Internal (15)	0		
	CSTLS6M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE8M00G55-R0	SMD	8.0	Internal (33)	Internal (33)	0		
	CSTLS8M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE8M38G55-R0	SMD	8.388	Internal (33)	Internal (33)	0		
	CSTLS8M38G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE10M0G52-R0	SMD	10.0	Internal (10)	Internal (10)	0		
	CSTLS10M0G53-B0	Lead		Internal (15)	Internal (15)	0		

(2) X1 oscillation: Ceramic resonator (AMPH = 1, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd.	CSTCE12M0G55-R0	SMD	12.0	Internal (33)	Internal (33)	0	1.8	5.5
	CSTCE16M0V53-R0	SMD	16.0	Internal (15)	Internal (15)	0		
	CSTLS16M0X51-B0	Lead		Internal (5)	Internal (5)	0		
	CSTCE20M0V53-R0	SMD	20.0	Internal (15)	Internal (15)	0		
	CSTLS20M0X51-B0	Lead		Internal (5)	Internal (5)	0		

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 78K0/KC2-C, 78K0/KE2-C so that the internal operation conditions are within the specifications of the DC and AC characteristics.

(3) XT1 oscillation: Crystal resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Load Capacitance CL (pF)	Recommended Circuit Constants			Oscillation Voltage Range	
					C3 (pF)	C4 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
SEIKO INSTRUMENTS INC	SSP-T7-F	SMD	32.768	7	10	12	0	1.8	5.5
CITIZEN FINETECH MIYOTA CO., LTD.	CMR200T	SMD	32.768	9	15	12	0	1.8	5.5

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 78K0/KC2-C, 78K0/KE2-C so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (1/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P120, P130, P140, P141		4.0 V ≤ V_{DD} ≤ 5.5 V			–3	mA
				2.7 V ≤ V_{DD} < 4.0 V			–2.5	mA
				1.8 V ≤ V_{DD} < 2.7 V			–1.0	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141		4.0 V ≤ V_{DD} ≤ 5.5 V			–20.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			–5.0	mA
		Total of P05, P06, P10 to P17, P30 to P33, P50 to P53		4.0 V ≤ V_{DD} ≤ 5.5 V			–30.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			–19.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			–10.0	mA
		Total of all the pins above[Note 3]		4.0 V ≤ V_{DD} ≤ 5.5 V			–50.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			–29.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			–15.0	mA
	I_{OH2}	Per pin for P20 to P27	AV_{REF} = V_{DD}	4.0 V ≤ V_{DD} ≤ 5.5 V			–100	μA
				2.7 V ≤ V_{DD} < 4.0 V			–100	μA
				1.8 V ≤ V_{DD} < 2.7 V			–100	μA
Output current, low[Note 2]	I_{OL1}	Per pin for P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P120, P130, P140, P141		4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
				2.7 V ≤ V_{DD} < 4.0 V			5.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Per pin for P60 to P63, P70 to P77		4.0 V ≤ V_{DD} ≤ 5.5 V			15.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			5.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141		4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P77		4.0 V ≤ V_{DD} ≤ 5.5 V			45.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			20.0	mA
		Total of all the pins above[Note 3]		4.0 V ≤ V_{DD} ≤ 5.5 V			65.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			50.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			29.0	mA
	I_{OL2}	Per pin for P20 to P27	AV_{REF} = V_{DD}	4.0 V ≤ V_{DD} ≤ 5.5 V			400	μA
				2.7 V ≤ V_{DD} < 4.0 V			400	μA
				1.8 V ≤ V_{DD} < 2.7 V			400	μA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where the duty factor is 50%, 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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (2/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P02 to P04, P12, P13, P15, P40, P42, P52, P53, P121 to P124			0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P05, P06, P10, P11, P14, P16, P17, P30 to P33, P41, P43, P50, P51, P120, P140, P141, $\overline{\text{RESET}}$, EXCLK			0.8V_{DD}		V_{DD}	V
	V_{IH3}	P60 to P63, P70, P71, P73 to P77			0.7V_{DD}		6.0	V
	V_{IH4}	P20 to P27	AV_{REF} = V_{DD}		0.7AV_{REF}		AV_{REF}	V
	V_{IH5}	P72	Normal mode		0.7V_{DD}		6.0	V
			CEC input buffer V_{DD} = 3.3 V±10%		2.0		3.6	V
Input voltage, low	V_{IL1}	P02 to P04, P12, P13, P15, P40, P42, P52, P53, P60 to P63, P70, P71, P73 to P77, P121 to P124			0		0.3V_{DD}	V
	V_{IL2}	P00, P01, P05, P06, P10, P11, P14, P16, P17, P30 to P33, P41, P43, P50, P51, P120, P140, P141, $\overline{\text{RESET}}$, EXCLK			0		0.2V_{DD}	V
	V_{IL3}	P20 to P27	AV_{REF} = V_{DD}		0		0.3AV_{REF}	V
	V_{IL4}	P72	Normal mode		0		0.3V_{DD}	V
			CEC input buffer V_{DD} = 3.3 V±10%		0		0.8	V
Output voltage, high	V_{OH1}	P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P120, P130, P140, P141	I_{OH1} = –3.0 mA	4.0 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.7			V
			I_{OH1} = –2.5 mA	2.7 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.5			V
			I_{OH1} = –1.0 mA	1.8 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27	AV_{REF} = V_{DD}, I_{OH2} = –100 μA		V_{DD} – 0.5			V
	V_{OH3}	P72	CECIO mode[Note], V_{DD} = 3.3 V±10%, I_{OH} = –12.0 μA		2.5		3.6	V

Note When a pull-up resistor and a diode are connected.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (3/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output voltage, low	V_{OL1}	P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P120, P130, P140, P141	I_{OL1} = 8.5 mA	4.0 V ≤ V_{DD} ≤ 5.5 V			0.7	V
			I_{OL1} = 5.0 mA	2.7 V ≤ V_{DD} ≤ 5.5 V			0.7	V
			I_{OL1} = 2.0 mA	1.8 V ≤ V_{DD} ≤ 5.5 V			0.5	V
			I_{OL1} = 1.0 mA	1.8 V ≤ V_{DD} ≤ 5.5 V			0.5	V
			I_{OL1} = 0.5 mA	1.8 V ≤ V_{DD} ≤ 5.5 V			0.4	V
	V_{OL2}	P20 to P27	AV_{REF} = V_{DD}, I_{OL2} = 400 μA				0.4	V
	V_{OL3}	P60 to P63, P70, P71, P73 to P77	I_{OL1} = 15.0 mA	4.0 V ≤ V_{DD} ≤ 5.5 V			2.0	V
			I_{OL1} = 5.0 mA	4.0 V ≤ V_{DD} ≤ 5.5 V			0.4	V
			I_{OL1} = 5.0 mA	2.7 V ≤ V_{DD} < 4.0 V			0.6	V
			I_{OL1} = 3.0 mA	2.7 V ≤ V_{DD} ≤ 5.5 V			0.4	V
			I_{OL1} = 2.0 mA	1.8 V ≤ V_{DD} ≤ 5.5 V			0.4	V
	V_{OL4}	P72	Normal mode, V_{DD} = 3.3 V±10%, I_{OL1} = 3 mA				0.4	V
			CECIO mode, V_{DD} = 3.3 V±10%, I_{OL1} = 3.3 mA				0.6	V
Input leakage current, high	I_{LIH1}	P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P77, P120, P140, P141, FLMD0, $\overline{\text{RESET}}$	V_I = V_{DD}				1	μA
	I_{LIH2}	P20 to P27	V_I = AV_{REF} = V_{DD}				1	μA
	I_{LIH3}	P121 to 124 (X1, X2, XT1, XT2)	V_I = V_{DD}	I/O port mode			1	μA
				OSC mode			20	μA
	I_{LIH4}	P72	Normal mode, V_I = V_{DD}				1	μA
			CECIO mode, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_I = 3.6 V				1.8	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (4/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, low	I_{LIL1}	P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70, P71, P73 to P77, P120, P140, P141, FLMD0, $\overline{\text{RESET}}$	V_I = V_{SS}			–1	μA
	I_{LIL2}	P20 to P27	V_I = V_{SS}, AV_{REF} = V_{DD}			–1	μA
	I_{LIL3}	P121 to 124 (X1, X2, XT1, XT2)	V_I = V_{SS} I/O port mode			–1	μA
			V_I = V_{SS} OSC mode			–20	μA
	I_{LIH4}	P72	Normal mode, V_I = V_{SS}			–1	μA
			CECIO mode, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_I = V_{SS}			–1.8	μA
Pull-up resistor	R_U	P00 to P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P120, P140, P141	V_I = V_{SS}	10	20	100	kΩ
		P72	CECIO mode[Note], V_I = V_{SS}	18	26	28.6	kΩ
FLMD0 supply voltage	V_{IL}	In normal operation mode		0		0.2V_{DD}	V
	V_{IH}	In self-programming mode		0.8V_{DD}		V_{DD}	V

Note When a pull-up resistor and a diode are connected.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (5/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode Note 1	f_{XH} = 20 MHz, V_{DD} = 5.0 V Note 2	Square wave input		3.0	5.0	mA
				Resonator connection		3.5	5.5	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V Note 2	Square wave input		1.6	3.0	mA
				Resonator connection		2.3	3.4	mA
			f_{XH} = 10 MHz, V_{DD} = 3.0 V Notes 2, 3	Square wave input		1.5	2.9	mA
				Resonator connection		2.2	3.3	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V Notes 2, 3	Square wave input		0.9	1.7	mA
				Resonator connection		1.3	2.0	mA
			f_{XH} = 5 MHz, V_{DD} = 2.0 V Notes 2, 3	Square wave input		0.7	1.4	mA
				Resonator connection		1.0	1.6	mA
			f_{RH} = 8 MHz, V_{DD} = 5.0 V Note 4			1.4	2.3	mA
			f_{SUB} = 32.768 kHz, V_{DD} = 3.0 V Note 5	Resonator connection		6.7	26	μA
			f_{SUB} = 32.768 kHz, V_{DD} = 3.0 V, When XT1 oscillator, CEC receive, remote control receive, and RTC are operated Note 5	Resonator connection		7.55	26	μA

Notes 1. Total current flowing into the internal power supply (V_{DD}, EV_{DD}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum values include the peripheral operation current. However, not including the current flowing into the A/D converter, watchdog timer, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. Not including the operating current of the 8 MHz internal oscillator, 240 kHz internal oscillator, and XT1 oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

3. When AMPH (bit 0 of clock operation mode select register (OSCCTL)) = 0.

4. Not including the operating current of the X1 oscillator, XT1 oscillator, and 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

5. Not including the operating current of the X1 oscillation, 8 MHz internal oscillator, 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

Remarks 1. f_{XH}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency or external subsystem clock frequency)

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (6/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2}	HALT mode Note 1	f_{XH} = 20 MHz, V_{DD} = 5.0 V Note 2	Square wave input		0.8	2.8	mA
				Resonator connection		1.5	4.0	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V Notes 2, 3	Square wave input		0.4	1.4	mA
				Resonator connection		1.0	1.7	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V Notes 2, 3	Square wave input		0.2	0.7	mA
				Resonator connection		0.5	1.0	mA
			f_{RH} = 8 MHz, V_{DD} = 5.0 V Note 4			0.4	1.2	mA
			f_{SUB} = 32.768 kHz, V_{DD} = 3.0 V Note 5	Resonator connection		2.4	22	μA
			f_{SUB} = 32.768 kHz, V_{DD} = 3.0 V, When XT1 oscillator, CEC receive, remote control receive, and RTC are operated Note 5	Resonator connection		3.47	22	μA
	I_{DD3} Note 6	STOP mode				1	20	μA
			T_A = –40 to +70 °C			1	10	μA
			When XT1 oscillator, CEC receive, remote control receive, and RTC are operated			3.47	22	μA

Notes 1. Total current flowing into the internal power supply (V_{DD}, EV_{DD}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum values include the peripheral operation current. However, not including the current flowing into the A/D converter, watchdog timer, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. Not including the operating current of the 8 MHz internal oscillator, 240 kHz internal oscillator, and XT1 oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

3. When AMPH (bit 0 of clock operation mode select register (OSCCTL)) = 0.

4. Not including the operating current of the X1 oscillator, XT1 oscillator, and 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

5. Not including the operating current of the X1 oscillation, 8 MHz internal oscillator, 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

6. Not including the operating current of the 240 kHz internal oscillator and XT1 oscillation, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

Remarks 1. f_{XH}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency or external subsystem clock frequency)

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (7/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> A/D converter operating current	I_{ADC} Note 1	ADCS = 1, 2.3 V ≤ AV_{REF} ≤ V_{DD}		0.86	1.9	mA
Watchdog timer operating current	I_{WDT} Note 2	During 240 kHz internal low-speed oscillation clock operation		5	10	μA
LVI operating current	I_{LVI} Note 3			9	18	μA

Notes 1. Current flowing only to the A/D converter (AV_{REF}). The current value of the 78K0/Kx2-C microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

2. Current flowing only to the watchdog timer (including the operating current of the 240 kHz internal oscillator). The current value of the 78K0/Kx2-C microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.

3. Current flowing only to the LVI circuit. The current value of the 78K0/Kx2-C microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

AC Characteristics

(1) Basic operation (1/2)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	2.7 V ≤ V_{DD} ≤ 5.5 V	0.1		16	μs
			1.8 V ≤ V_{DD} < 2.7 V	0.4		16	μs
		Subsystem clock (f_{SUB}) operation		114	122	125	μs
Peripheral hardware clock frequency	f_{PRS}	f_{PRS} = f_{XH} (XSEL = 1) Note 1	2.7 V ≤ V_{DD} ≤ 5.5 V			20	MHz
			1.8 V ≤ V_{DD} < 2.7 V			5	MHz
		f_{PRS} = f_{RH} (XSEL = 0) Note 2	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6		8.4	MHz
			1.8 V ≤ V_{DD} < 2.7 V	6.75		8.4	MHz
External main system clock frequency	f_{EXCLK}	2.7 V ≤ V_{DD} ≤ 5.5 V		2.0		20.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V		2.0		5.0	MHz
External main system clock input high-level width, low-level width	t_{EXCLKH}, t_{EXCLKL}	2.7 V ≤ V_{DD} ≤ 5.5 V		24			ns
		1.8 V ≤ V_{DD} < 2.7 V		96			ns

Notes 1. Characteristics of the main system clock frequency. Set the division clock to be set by a peripheral function to f_{XH}/2 (10 MHz) or less. The multiplier/divider, however, can operate on f_{XH} (20 MHz).

2. Characteristics of the main system clock frequency. Set the division clock to be set by a peripheral function to f_{RH}/2 (5 MHz) or less.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(1) Basic operation (2/2)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

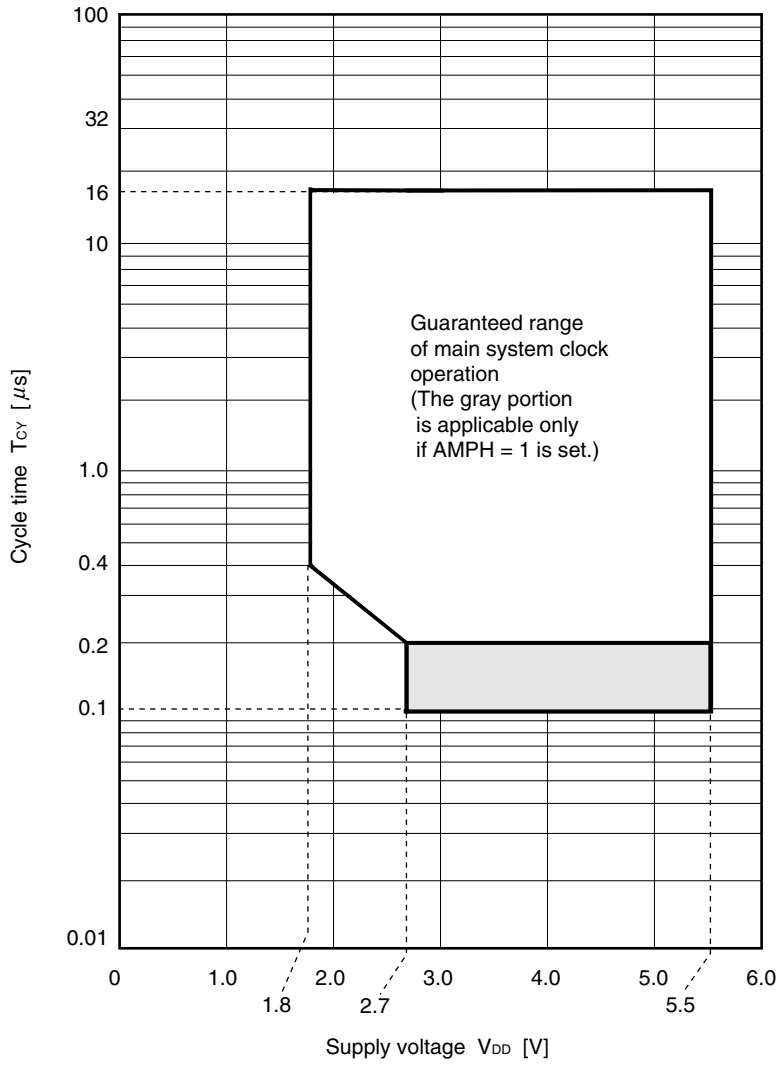
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
TI000, TI010, TI001, TI011, TI002, TI012 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} < 5.5 V	$2/f_{sam}$ + 0.1[Note]			μs
		2.7 V ≤ V_{DD} < 4.0 V	$2/f_{sam}$ + 0.2[Note]			μs
		1.8 V ≤ V_{DD} < 2.7 V	$2/f_{sam}$ + 0.5[Note]			μs
TI50, TI51 input frequency	f_{TI5}	2.7 V ≤ V_{DD} ≤ 5.5 V			10	MHz
		1.8 V ≤ V_{DD} < 2.7 V			5	MHz
TI50, TI51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	2.7 V ≤ V_{DD} ≤ 5.5 V	50			ns
		1.8 V ≤ V_{DD} < 2.7 V	100			ns
Interrupt input high-level width, low-level width	t_{INIH}, t_{INIL}		1			μs
Key interrupt input low-level width	t_{KR}		250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}		10			μs

Note Selection of f_{sam} = f_{PRS}, f_{PRS}/4, f_{PRS}/256, or f_{PRS}, f_{PRS}/16, f_{PRS}/64 is possible using bits 0 and 1 (PRM000, PRM001 or PRM010, PRM011 or PRM020, PRM021) of prescaler mode registers 00, 01, and 02 (PRM00, PRM01, and PRM02). Note that when selecting the TI000, TI001, or TI002 valid edge as the count clock, f_{sam} = f_{PRS}.

Remark f_{sam}: sampling clock frequency

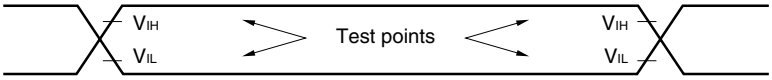
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

T_{CY} vs. V_{DD} (Main System Clock Operation)

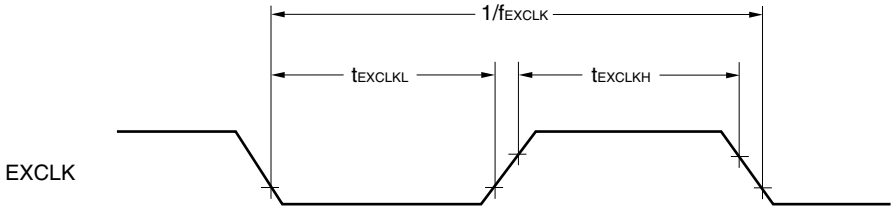


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

AC Timing Test Points

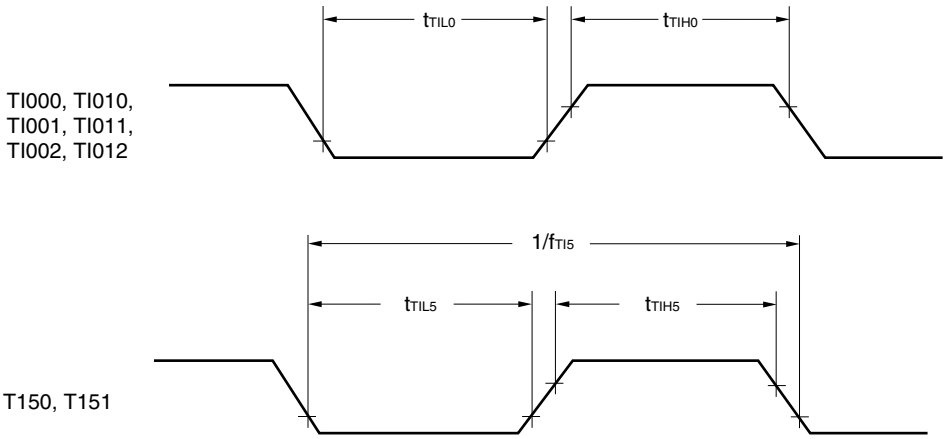


External Main System Clock Timing

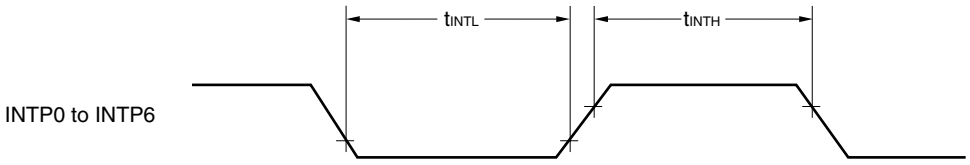


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

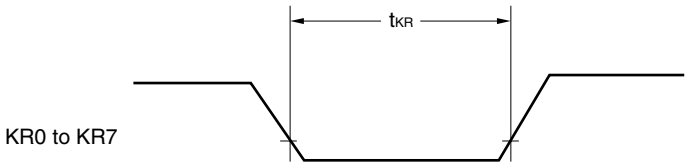
TI Timing



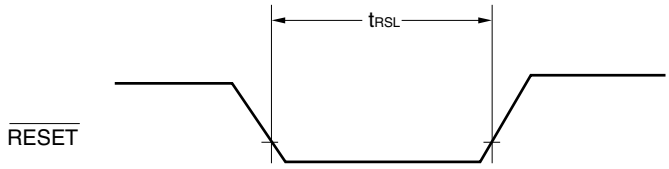
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(2) Serial interface

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) UART60, UART 61 (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

(b) UART0 (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

(c) IICA00, IICA01, IICA02

Parameter	Symbol	Conditions	Standard Mode		High-Speed Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{SCL}	Fast mode: 3.5 MHz ≤ f_{PRS} Normal mode: 1 MHz ≤ f_{PRS}	0	100	0	400	kHz
Setup time of restart condition	$t_{SU:STA}$		4.7	–	0.6	–	μs
Hold time[Note 1]	$t_{HD:STA}$		4.0	–	0.6	–	μs
Hold time when 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, 3]	$t_{HD:DAT}$		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.

3. Data hold time changes depending on the setting of IICA low-level width setting register (IICAWL0n).

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(d) CSI10 (master mode, $\overline{SCK10}$... internal clock output) (communication at same potential: CMOS output, normal schmitt input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	$4.0\ V \le V_{DD} \le 5.5\ V$	160			ns
		$2.7\ V \le V_{DD} < 4.0\ V$	250			ns
		$1.8\ V \le V_{DD} < 2.7\ V$	500			ns
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}	$4.0\ V \le V_{DD} \le 5.5\ V$	$t_{KCY1}/2 - 15$ Note 1			ns
		$2.7\ V \le V_{DD} < 4.0\ V$	$t_{KCY1}/2 - 25$ Note 1			ns
		$1.8\ V \le V_{DD} < 2.7\ V$	$t_{KCY1}/2 - 50$ Note 1			ns
SI10 setup time (to $\overline{SCK10}\uparrow$)	t_{SIK1}	$4.0\ V \le V_{DD} \le 5.5\ V$	55			ns
		$2.7\ V \le V_{DD} < 4.0\ V$	80			ns
		$1.8\ V \le V_{DD} < 2.7\ V$	170			ns
SI10 hold time (from $\overline{SCK10}\uparrow$)	t_{KSI1}		30			ns
Delay time from $\overline{SCK10}\downarrow$ to SO10 output	t_{KSO1}	C = 50 pF Note 2			40	ns

Notes 1. This value is when high-speed system clock (f_{XH}) is used.

2. C is the load capacitance of the $\overline{SCK10}$ and SO10 output lines.

(e) CSI10 (slave mode, $\overline{SCK10}$... external clock input) (communication at same potential: CMOS output, normal schmitt input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}			400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}			$t_{KCY2}/2$			ns
SI10 setup time (to $\overline{SCK10}\uparrow$)	t_{SIK2}			80			ns
SI10 hold time (from $\overline{SCK10}\uparrow$)	t_{KSI2}			50			ns
Delay time from $\overline{SCK10}\downarrow$ to SO10 output	t_{KSO2}	C = 50 pF Note	$2.7\ V \le V_{DD} \le 5.0\ V$			120	ns
			$1.8\ V \le V_{DD} < 2.7\ V$			165	ns

Note C is the load capacitance of the SO10 output line.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(f) CSI11 (master mode, $\overline{SCK11}$... internal clock output) (communication at different potential: N-ch open-drain output, Schmitt input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK11}$ cycle time	t_{KCY1}	R_U = 1 kΩ, C = 50 pF[Note 1]	4.0 V ≤ V_{DD} ≤ 5.5 V	300			ns
			2.7 V ≤ V_{DD} < 4.0 V	350			ns
			1.8 V ≤ V_{DD} < 2.7 V	550			ns
$\overline{SCK11}$ high-level width	t_{KH1}	R_U = 1 kΩ, C = 50 pF[Note 1]	4.0 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2-75$[Note 2]			ns
			2.7 V ≤ V_{DD} < 4.0 V	$t_{KCY1}/2-85$[Note 2]			ns
			1.8 V ≤ V_{DD} < 2.7 V	$t_{KCY1}/2-110$[Note 2]			ns
$\overline{SCK11}$ low-level width	t_{KL1}	R_U = 1 kΩ, C = 50 pF[Note 1]	4.0 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2-15$[Note 2]			ns
			2.7 V ≤ V_{DD} < 4.0 V	$t_{KCY1}/2-25$[Note 2]			ns
			1.8 V ≤ V_{DD} < 2.7 V	$t_{KCY1}/2-50$[Note 2]			ns
SI11 setup time (to $\overline{SCK11}$↑)[Note 3]	t_{SIK1}	R_U = 1 kΩ, C = 50 pF[Note 1]	4.0 V ≤ V_{DD} ≤ 5.5 V	120			ns
			2.7 V ≤ V_{DD} < 4.0 V	145			ns
			1.8 V ≤ V_{DD} < 2.7 V	235			ns
SI11 hold time (from $\overline{SCK11}$↑)[Note 3]	t_{KSI1}	R_U = 1 kΩ, C = 50 pF[Note 1]		30			ns
Delay time from $\overline{SCK11}$↓ to SO11 output[Note 3]	t_{KSO1}	R_U = 1 kΩ, C = 50 pF[Note 1]				105	ns
SI11 setup time (to $\overline{SCK11}$↓)[Note 4]	t_{SIK1}	R_U = 1 kΩ, C = 50 pF[Note 1]	4.0 V ≤ V_{DD} ≤ 5.5 V	50			ns
			2.7 V ≤ V_{DD} < 4.0 V	80			ns
			1.8 V ≤ V_{DD} < 2.7 V	170			ns
SI11 hold time (from $\overline{SCK11}$↓)[Note 4]	t_{KSI1}	R_U = 1 kΩ, C = 50 pF[Note 1]		30			ns
Delay time from $\overline{SCK11}$↑ to SO11 output[Note 4]	t_{KSO1}	R_U = 1 kΩ, C = 50 pF[Note 1]				40	ns

Notes 1. R_U is the external pull-up resistance of the $\overline{SCK11}$. C is the load capacitance of the $\overline{SCK11}$ and SO11 output line.

2. This value is when high-speed system clock (f_{XH}) is used.

3. When DAP11 = 0, CKP11 = 0 or when DAP11 = 1, CKP11 = 1.

4. When DAP11 = 0, CKP11 = 1 or when DAP11 = 1, CKP11 = 0.

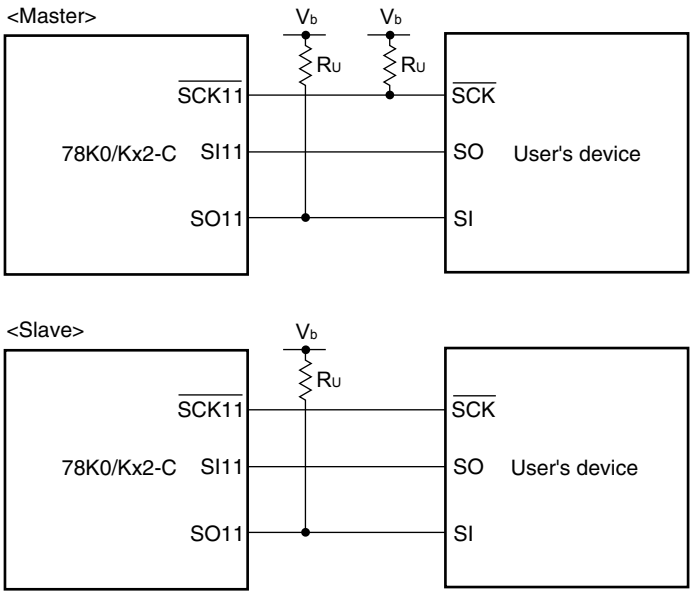
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(g) CSI11 (slave mode, $\overline{SCK11}$... external clock input) (communication at different potential: N-ch open-drain output, Schmitt input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK11}$ cycle time	t_{KCY2}	$2.7\ V \le V_{DD} \le 5.5\ V$		400			ns
		$1.8\ V \le V_{DD} < 2.7\ V$		500			ns
$\overline{SCK11}$ high-/low-level width	t_{KH2}, t_{KL2}			$t_{KCY2}/2$			ns
SI11 setup time (to $\overline{SCK11}\uparrow$)	t_{SIK2}			80			ns
SI11 hold time (from $\overline{SCK11}\uparrow$)	t_{KSI2}			50			ns
Delay time from $\overline{SCK11}\downarrow$ to SO11 output	t_{KSO2}	$R_U = 1\ k\Omega$, $C = 50\ pF$[Note]	$2.7\ V \le V_{DD} \le 5.5\ V$			185	ns
			$1.8\ V \le V_{DD} < 2.7\ V$			230	ns

Note R_U is the external pull-up resistance of the SO11. C is the load capacitance of the SO11 output line.

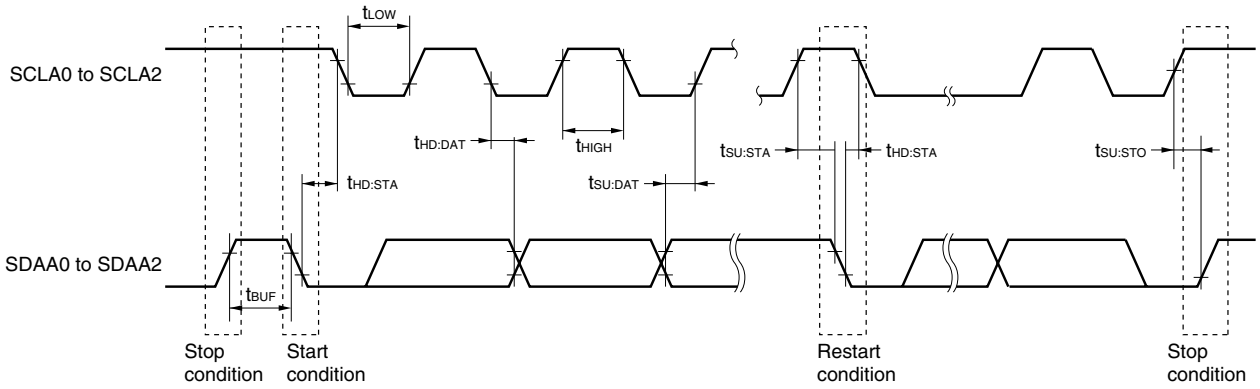
CSI mode connection diagram (communication at different potential)



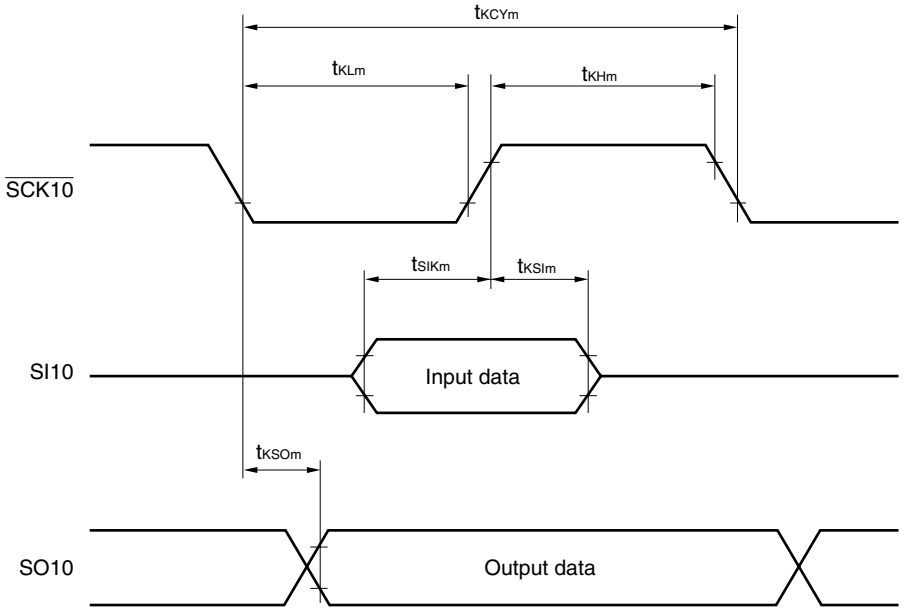
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Serial Transfer Timing

IICA0 to IICA2:



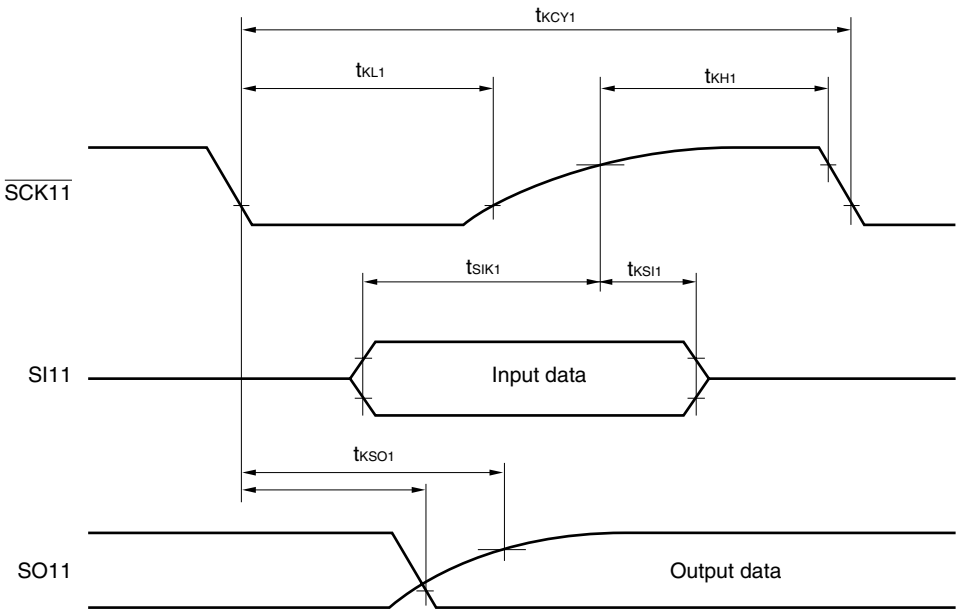
CSI10:



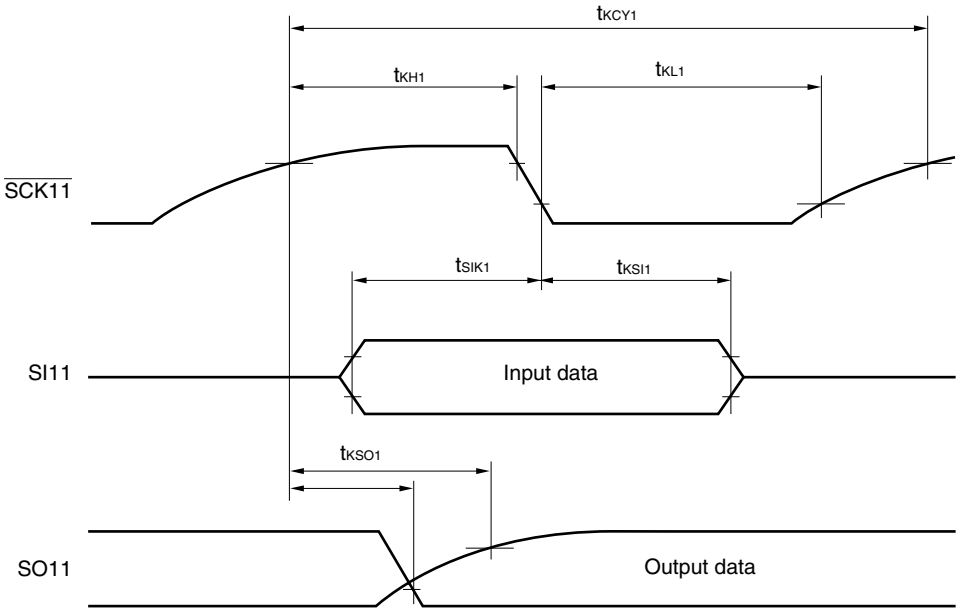
Remark m = 1, 2

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

CSI11: master mode (communication at different potential)
(When DAP11 = 0, CKP11 = 0 or when DAP11 = 1, CKP11 = 1)

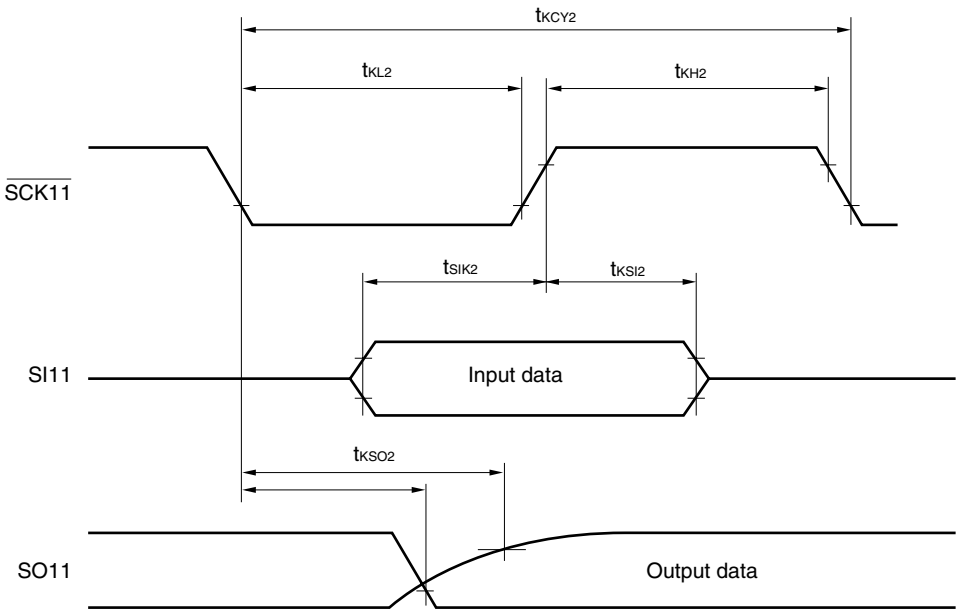


CSI11: master mode (communication at different potential)
(When DAP11 = 0, CKP11 = 1 or when DAP11 = 1, CKP11 = 0)

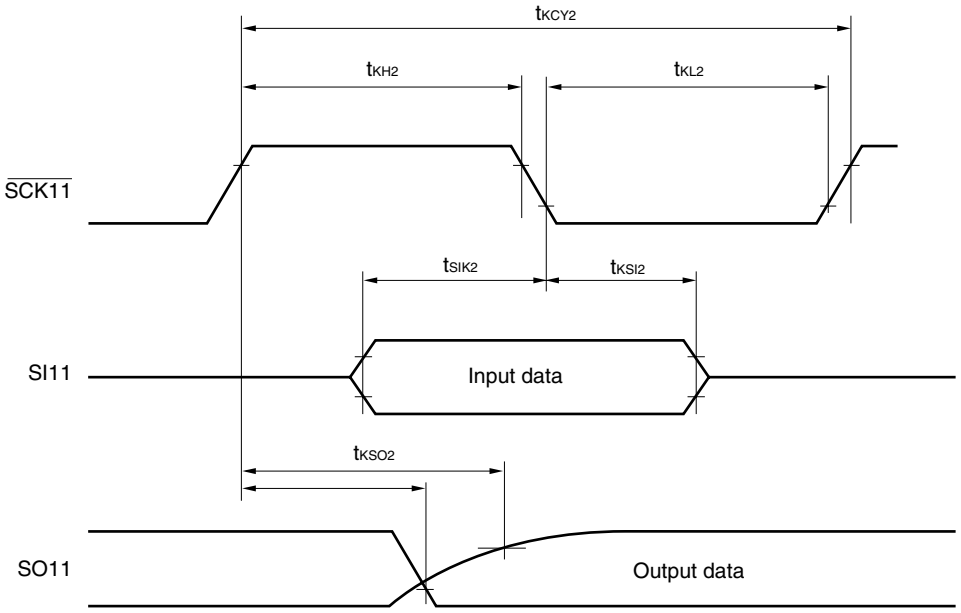


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

CSI11: slave mode (communication at different potential)
(When DAP11 = 0, CKP11 = 0 or when DAP11 = 1, CKP11 = 1)



CSI11: slave mode (communication at different potential)
(When DAP11 = 0, CKP11 = 1 or when DAP11 = 1, CKP11 = 0)



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

<R> **CEC Transmission/Reception Circuit**

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 3.63 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input hysteresis width	IHYS	P72	V_{DD} = 3.3 V±10%		0.4		V
Rise time of CECIO	t_R	P72	CECIO mode[Note] C_b = 1600 pF, R_b = 27 kΩ			250	μs
			CECIO mode[Note] C_b = 7700 pF, R_b = 3 kΩ			250	μs
Fall time of CECIO	t_F	P72	CECIO mode[Note] C_b = 1600 pF, R_b = 27 kΩ			50	μs
			CECIO mode[Note] C_b = 7700 pF, R_b = 3 kΩ			50	μs

Note When a pull-up resistor and a diode are connected. (PF72 = 1, PU72 = 1)

Remark C_b[F]: Communication line load capacitance, R_b[Ω]:Communication line pull-up resistance

A/D Converter Characteristics

(T_A = –40 to +85°C, 2.3 V ≤ AV_{REF} ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 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	4.0 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} < 4.0 V			±0.6	%FSR
		2.3 V ≤ V_{DD} < 2.7 V			±1.2	%FSR
Conversion time	tCONV	4.0 V ≤ V_{DD} ≤ 5.5 V	6.1		66.6	μs
		2.7 V ≤ V_{DD} < 4.0 V	13.2		66.6	μs
		2.3 V ≤ V_{DD} < 2.7 V	27.0		66.6	μs
Zero-scale error[Notes 1, 2]	EZS	4.0 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} < 4.0 V			±0.6	%FSR
		2.3 V ≤ V_{DD} < 2.7 V			±0.6	%FSR
Full-scale error[Notes 1, 2]	EFS	4.0 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} < 4.0 V			±0.6	%FSR
		2.3 V ≤ V_{DD} < 2.7 V			±0.6	%FSR
Integral non-linearity error[Note 1]	ILE	4.0 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ V_{DD} < 4.0 V			±4.5	LSB
		2.3 V ≤ V_{DD} < 2.7 V			±6.5	LSB
Differential non-linearity error[Note 1]	DLE	4.0 V ≤ V_{DD} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ V_{DD} < 4.0 V			±2.0	LSB
		2.3 V ≤ V_{DD} < 2.7 V			±2.0	LSB
Analog input voltage	VAIN		AVSS		AVREF	V

Notes 1. Excludes quantization error (±1/2 LSB).

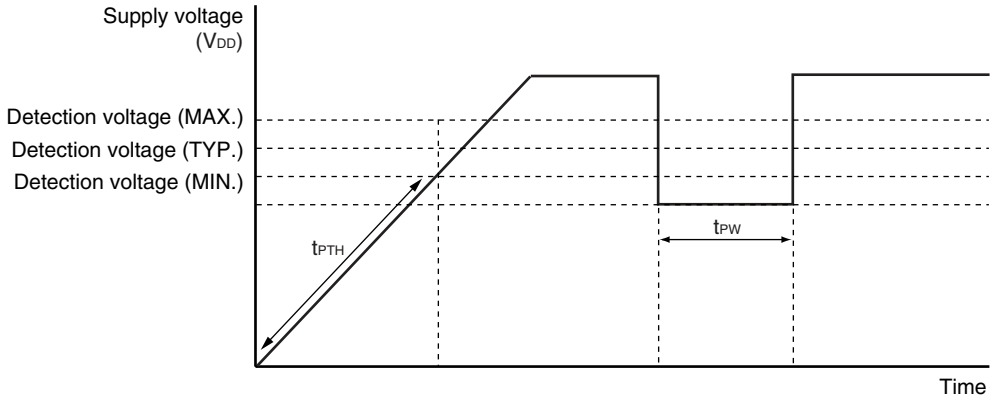
2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

1.59 V POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		1.44	1.59	1.74	V
Power supply voltage rise inclination	t_{PTH}	V_{DD}: 0 V → change inclination of V_{POC}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs

1.59 V POC Circuit Timing



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

2.7 V POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage When Power Supply Voltage Is Turned On	V_{DDPOC}	POCMODE (option byte) = 1	2.50	2.70	2.90	V

Remark The operations of the POC circuit are as described below, depending on the POCMODE (option byte) setting.

Option Byte Setting	POC Mode	Operation
POCMODE = 0	1.59 V mode operation	A reset state is retained until V_{POC} = 1.59 V (TYP.) is reached after the power is turned on, and the reset is released when V_{POC} is exceeded. After that, POC detection is performed at V_{POC}, similarly as when the power was turned on. The power supply voltage must be raised at a time of t_{PUP1} or t_{PUP2} when POCMODE is 0.
POCMODE = 1	2.7 V/1.59 V mode operation	A reset state is retained until V_{DDPOC} = 2.7 V (TYP.) is reached after the power is turned on, and the reset is released when V_{DDPOC} is exceeded. After that, POC detection is performed at V_{POC} = 1.59 V (TYP.) and not at V_{DDPOC}. The use of the 2.7 V/1.59 V POC mode is recommended when the rise of the voltage, after the power is turned on and until the voltage reaches 1.8 V, is more relaxed than t_{PTH}.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

LVI Circuit Characteristics (T_A = –40 to +85°C, 1.8 ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

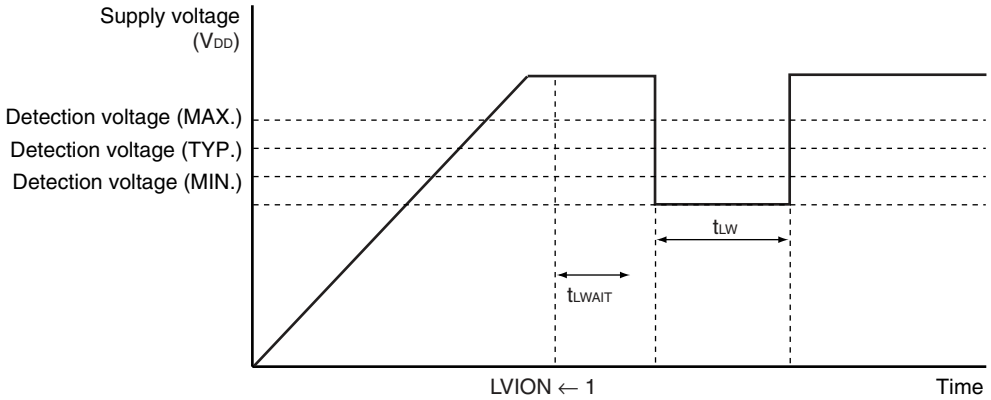
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.14	4.24	4.34	V
		V_{LVI1}		4.99	4.09	4.19	V
		V_{LVI2}		3.83	3.93	4.03	V
		V_{LVI3}		3.68	3.78	3.88	V
		V_{LVI4}		3.52	3.62	3.72	V
		V_{LVI5}		3.37	3.47	3.57	V
		V_{LVI6}		3.22	3.32	3.42	V
		V_{LVI7}		3.06	3.16	3.26	V
		V_{LVI8}		2.91	3.01	3.11	V
		V_{LVI9}		2.75	2.85	2.95	V
		V_{LVI10}		2.60	2.70	2.80	V
		V_{LVI11}		2.45	2.55	2.65	V
		V_{LVI12}		2.29	2.39	2.49	V
		V_{LVI13}		2.14	2.24	2.34	V
		V_{LVI14}		1.98	2.08	2.18	V
		V_{LVI15}		1.83	1.93	20.3	V
	External input pin[Note 1]	EXLVI	EXLVI < V_{DD}, 1.8 V ≤ V_{DD} ≤ 5.5 V		1.21		V
Minimum pulse width		t_{LW}		200			μs
Operation stabilization wait time[Note 2]		t_{LWAIT}				10	μs

Notes 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

Remark $V_{LVI(n-1)}$ > V_{LVIn}: n = 1 to 4

LVI Circuit Timing



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

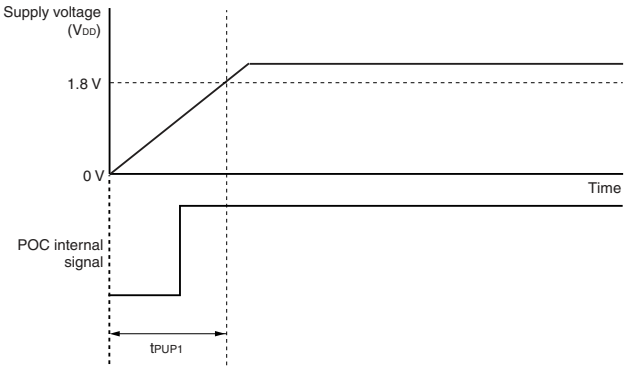
Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) (V_{DD}: 0 V → 1.8 V)[Note]	t_{PUP1}	LVI default start function stopped is set (POCMODE (option byte) = 0), when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) (releasing $\overline{\text{RESET}}$ input → V_{DD}: 1.8 V)[Note]	t_{PUP2}	LVI default start function stopped is set (POCMODE (option byte) = 0), when $\overline{\text{RESET}}$ input is used			1.9	ms

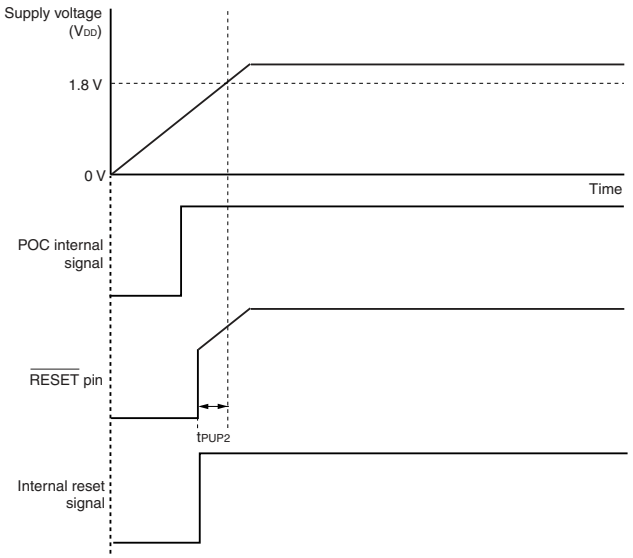
Note Make sure to raise the power supply in a shorter time than this.

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)

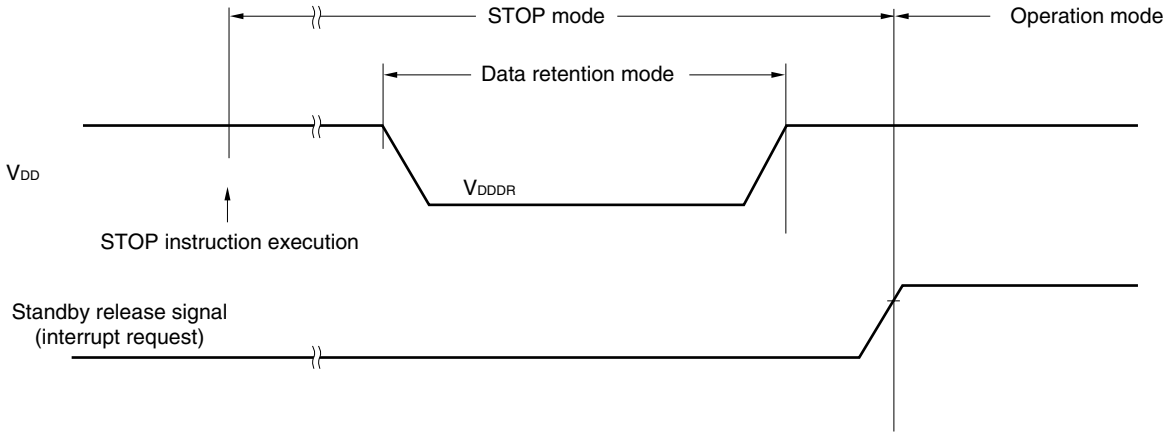


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

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}		1.44[Note]		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

• Basic characteristics

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}					4.5	11.0	mA
Number of rewrites per chip	C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 1]	• When using flash memory programmer and Renesas Electronics self programming library[Note 2] • For program update	Retention: 15 years	1000			Times
			• When using Renesas Electronics EEPROM emulation library[Note 3] • The rewritable ROM size: 4 KB • For data update	Retention: 5 years	10000			Times

Notes 1. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

2. The sample library specified by the **78K0/Kx2 Flash Memory Self Programming User’s Manual** (Document No.: **U17516E**) is excluded.

3. The sample program specified by the **78K0/Kx2 EEPROM Emulation Application Note** (Document No.: **U17517E**) is excluded.

Remarks 1. f_{XP}: Main system clock oscillation frequency

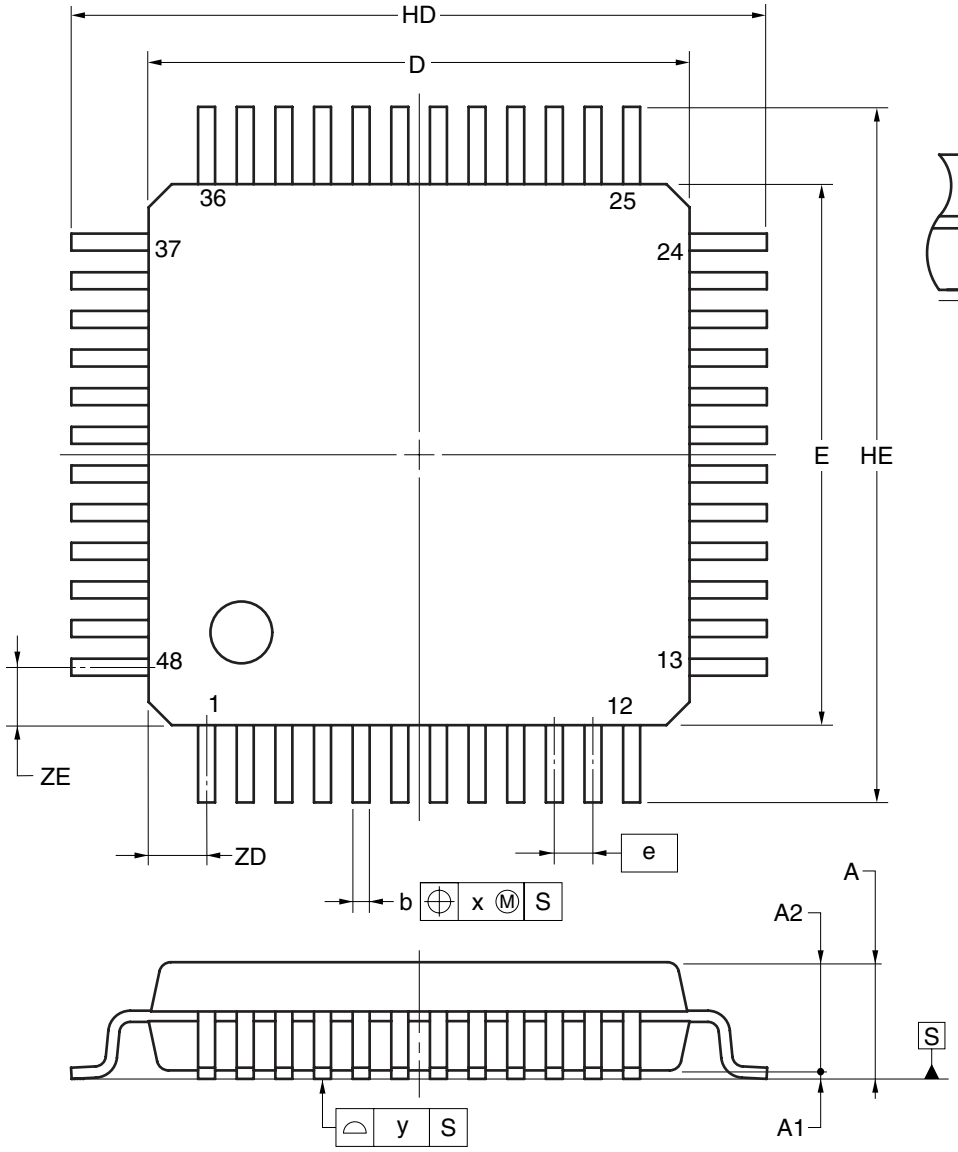
2. For serial write operation characteristics, refer to 78K0/Kx2 Flash Memory Programming (Programmer) Application Note (Document No.: U17739E).

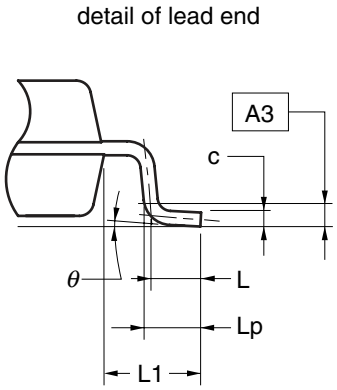
CHAPTER 31 PACKAGE DRAWINGS

31.1 78K0/KC2-C

• μPD78F0760GA-GAM-AX, 78F0761GA-GAM-AX, 78F0762GA-GAM-AX

48-PIN PLASTIC LQFP (FINE PITCH) (7x7)





(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	1.60 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	0.75
ZE	0.75

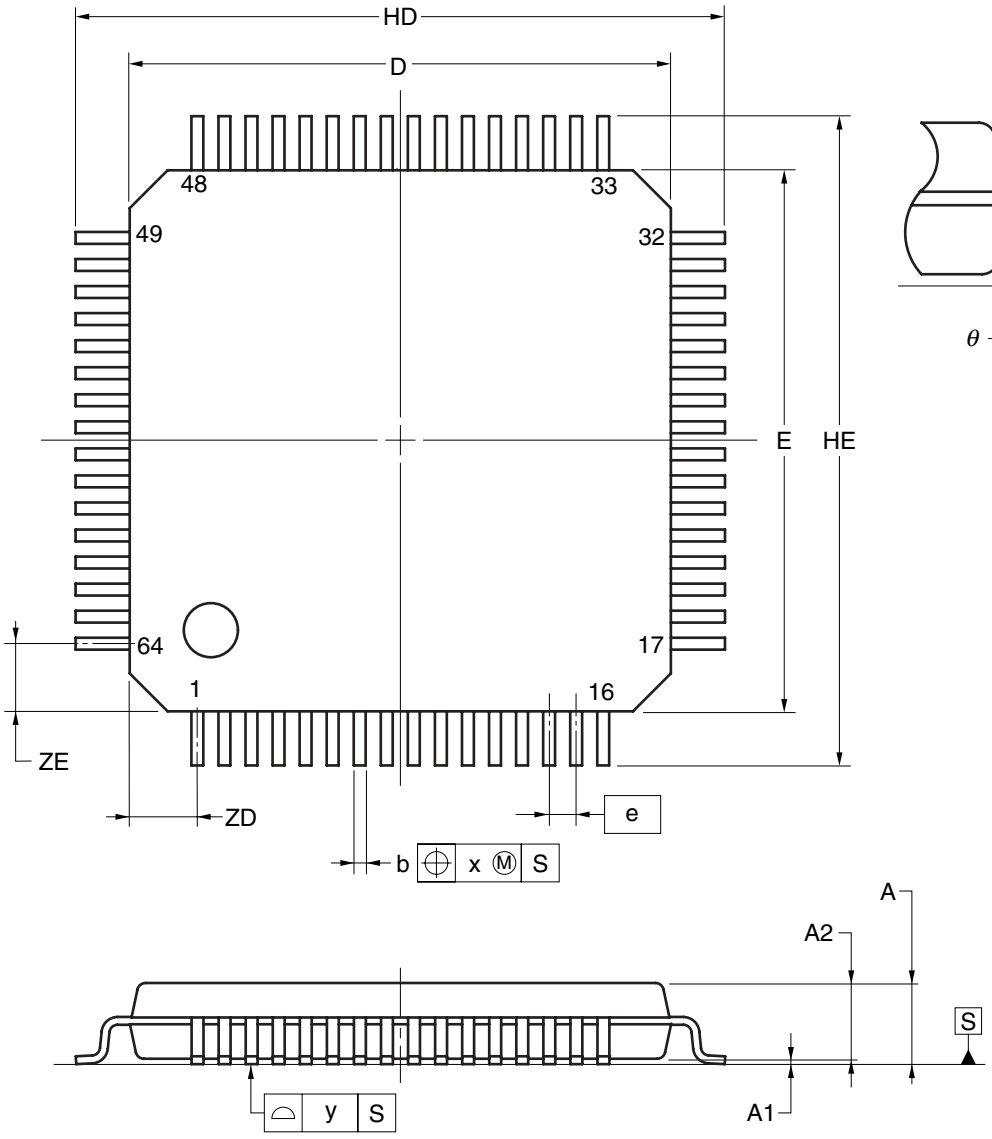
P48GA-50-GAM

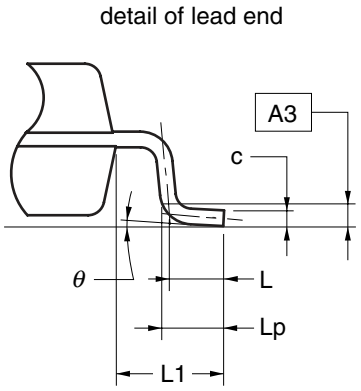
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

31.2 78K0/KE2-C

• μPD78F0763GB-GAH-AX, 78F0764GB-GAH-AX, 78F0765GB-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.

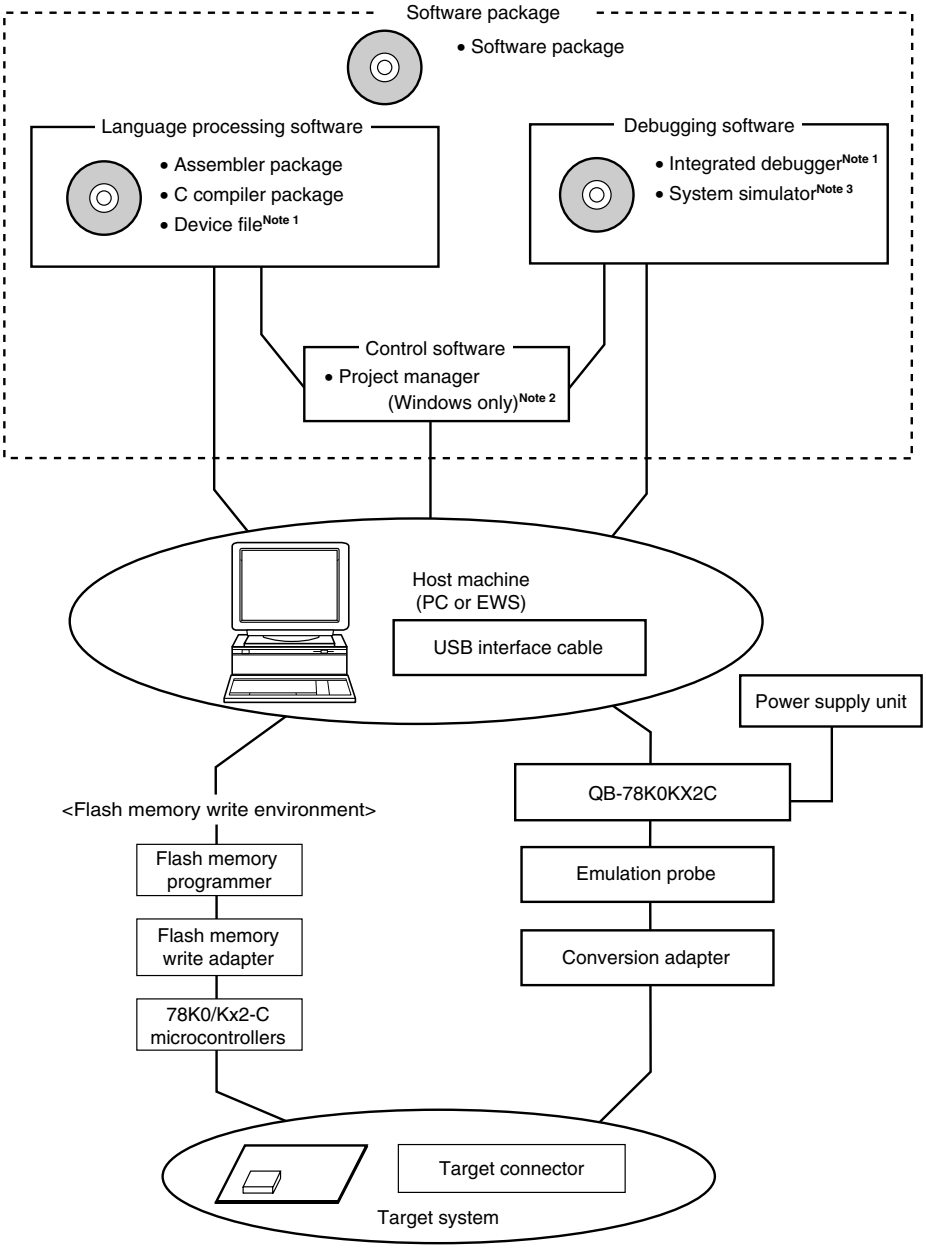
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0/Kx2-C microcontrollers.

Figure A-1 shows the development tool configuration.

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator QB-78K0KX2C



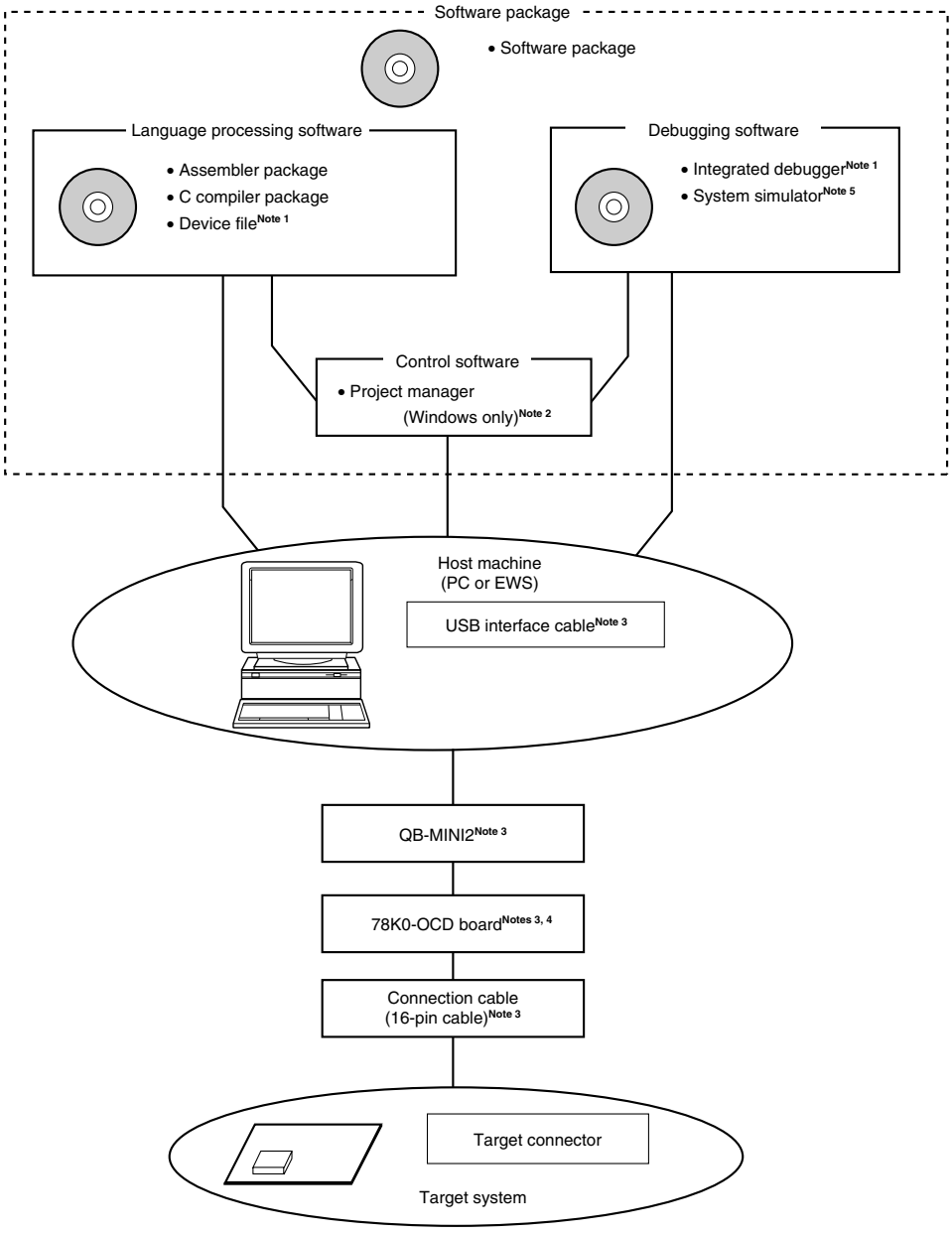
Notes 1. Download the device file for 78K0/Kx2-C microcontrollers (DF780765) and the integrated debugger ID78K0-QB from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

2. The project manager PM+ is included in the assembler package. The PM+ is only used for Windows™.

3. This is an instruction simulation version included in the software package.

Figure A-1. Development Tool Configuration (2/2)

(2) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file for 78K0/Kx2-C microcontrollers (DF780765) and the integrated debugger ID78K0-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.

3. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. In addition, download the software for operating the QB-MINI2 from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

4. This is used only when using QB-MINI2 as an on-chip debug emulator.

5. This is an instruction simulation version included in the software package.

A.1 Software Package

SP78K0 78K0 microcontroller software package	Development tools (software) common to the 78K0 microcontrollers are combined in this package.

A.2 Language Processing Software

RA78K0[Note 1] 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 (DF780765). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (PM+) on Windows. PM+ is included in assembler package.
CC78K0[Note 1] C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file. **<Precaution when using CC78K0 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 (PM+) on Windows. PM+ is included in assembler package.
DF780765[Note 2] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0, CC78K0, ID78K0-QB, and SM+ for 78K0). The corresponding OS and host machine differ depending on the tool to be used.

Notes 1. If the versions of RA78K0 and CC78K0 are Ver.4.00 or later, different versions of RA78K0 and CC78K0 can be installed on the same machine.

2. The DF780765 can be used in common with the RA78K0, CC78K0, ID78K0-QB, and SM+ for 78K0. Download the DF780765 from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

A.3 Flash Memory Programming Tools

A.3.1 When using flash memory programmer PG-FP5 and FL-PR5

PG-FP5, FL-PR5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-78F0515GA-8EU-RX, FA-78F0537GB-UEU-RX Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use.

Remarks 1. FL-PR5, FA-78F0515GA-8EU-RX, and FA-78F0537GB-UEU-RX are products of Naito Densei Machida Mfg. Co., Ltd.

TEL: +81-42-750-4172 Naito Densei Machida Mfg. Co., Ltd.

2. Use the latest version of the flash memory programming adapter.

A.3.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0/Kx2-C microcontrollers. When using this as flash memory programmer, it should be used in combination with a connection cable (16-pin cable) and a USB interface cable that is used to connect the host machine.
Target connector specifications	16-pin general-purpose connector (2.54 mm pitch)

Remark Download the software for operating the QB-MINI2 from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

A.4 Debugging Tools (Hardware)

A.4.1 When using in-circuit emulator

QB-78K0KX2C In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0/Kx2-C microcontrollers. It supports to the integrated debugger (ID78K0-QB). This emulator should be used in combination with a power supply unit and emulation probe, and the USB is used to connect this emulator to the host machine.

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0/Kx2-C. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. When using this as on-chip debug emulator, it should be used in combination with a connection cable (16-pin cable), a USB interface cable that is used to connect the host machine, and the 78K0-OCD board.
Target connector specifications	16-pin general-purpose connector (2.54 mm pitch)

Remark Download the software for operating the QB-MINI2 from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

A.5 Debugging Tools (Software)

ID78K0-QB Integrated debugger	This debugger supports the in-circuit emulators for the 78K0 microcontrollers. The ID78K0-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file (DF780765).
SM+ for 78K0 System simulator	System simulator is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of system simulator allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. System simulator should be used in combination with the device file (DF780765).

APPENDIX B REVISION HISTORY

B.1 Major Revisions in This Edition

Page	Description	Classification
Major Revisions in Rev.2.00		
CHAPTER 4 PORT FUNCTIONS		
p.142	Addition of **Figure 4-36. Relationship Between PU7 Register and PF7 Register**	(c)
p.143	Addition of **Figure 4-38. Relationship Between PU7 Register and PF7 Register**	(c)
CHAPTER 16 SERIAL INTERFACES IICA00, IICA01, AND IICA02		
p.485	Change of **16.5.4 Acknowledge ($\overline{\text{ACK}}$)**	(c)
p.496	Change of **16.5.13 Wakeup function**	(c)
CHAPTER 17 CEC TRANSMISSION/RECEPTION CIRCUIT		
p.557	Addition of **Note** to **Figure 17-12. Format of CEC Communication Error Status Register (CECES) (2/2)**	(c)
CHAPTER 30 ELECTRICAL SPECIFICATIONS		
p.767	Change of **(2) Non-port functions**	(a)
pp.772, 773	Addition of **Recommended oscillator circuit constants**	(b)
p.780	Change of conditions of A/D converter operating current in **DC Characteristics (7/7)**	(b)
APPENDIX B REVISION HISTORY		
p.809	Addition of **B.2 Revision History of Preceding Editions**	(c)
Major Revisions in Rev.2.01		
CHAPTER 16 SERIAL INTERFACES IICA00, IICA01, AND IICA02		
p.475	Change of **Figure 16-6. Format of IICA Status Register 0n (IICAS0n) (3/3)**	(c)
CHAPTER 17 CEC TRANSMISSION/RECEPTION CIRCUIT		
p.540	Change of description of **17.1 Functions of CEC Transmission/Reception Circuit**	(c)
CHAPTER 30 ELECTRICAL SPECIFICATIONS		
p.793	Addition of **CEC Transmission/Reception circuit Characteristics**	(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

<R>B.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

Edition	Description	Chapter
1st edition	Change URL of Renesas Electronics website	Throughout
	Change of status of μPD78F0760, 78F0761, 78F0762, 78F0763, 78F0764, and 78F0765 from under development to mass production	
	Change power supply voltage to V_{DD} = 1.8 to 5.5 V	
	Change of **Figure 12-3. Format of A/D Converter Mode Register (ADM)**	CHAPTER 12 A/D CONVERTER
	Change of **Table 12-2. A/D Conversion Time Selection**	
	Change of **Table 12-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	
	Change of description of ACKERR bit in **Figure 17-12. Format of CEC Communication Error Status Register (CECES) (2/2)**	CHAPTER 17 CEC TRANSMISSION/RECEPTION CIRCUIT
	Change of **Figure 25-3. Format of Low-Voltage Detection Level Selection Register (LVIS)**	CHAPTER 25 LOW-VOLTAGE DETECTOR
	Addition of specifications of V_{DD} = 1.8 to 5.5 V	CHAPTER 30 ELECTRICAL SPECIFICATIONS
	Change of specifications from target specifications to formal specifications	
	Addition of normal mode to Input leakage current, high in **DC Characteristics (3/7)** and Input leakage current, low in **DC Characteristics (4/7)**	
	Deletion of description of input hysteresis width, rise time of CECIO, and fall time of CECIO of **DC Characteristics (4/7)**	
	Change of specifications of supply current in **DC Characteristics (5/7)** and **DC Characteristics (6/7)**	
	Change of specifications of **(d) CSI10 (master mode, $\overline{\text{SCK10}}$... internal clock output) (communication at same potential: CMOS output, normal schmitt input)**	
	Change of specifications of **(e) CSI10 (slave mode, $\overline{\text{SCK10}}$... external clock input) (communication at same potential: CMOS output, normal schmitt input)**	
	Change of specifications of **(f) CSI11 (master mode, $\overline{\text{SCK11}}$... internal clock output) (communication at different potential: N-ch open-drain output, Schmitt input)**	
	Change of specifications of **(g) CSI11 (slave mode, $\overline{\text{SCK11}}$... external clock input) (communication at different potential: N-ch open-drain output, Schmitt input)**	
	Change of timing chart of CSI11 in **Serial Transfer Timing**	
	Change of specifications of **A/D Converter Characteristics** to AV_{REF} = 2.3 to 5.5 V	
	Change of **Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)**	
	Addition of chapter	APPENDIX A DEVELOPMENT TOOLS
	Addition of chapter	APPENDIX B REVISION HISTORY

78K0/Kx2-C User's Manual: Hardware

Publication Date: Rev.0.01 Aug 1, 2009
Rev.2.01 Sep 30, 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.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1

78K0/Kx2-C

Renesas Electronics Corporation

R01UH0014EJ0201