User's Manual

8

78K0/Fx2-L

User's Manual: Hardware

8-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.2.03 Jun 2012

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

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

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
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.

(2012.4)

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/Fx2-L microcontrollers and design and develop application systems and programs for these devices.
The target products are as follows.

- 78K0/FY2-L: μPD78F0854, 78F0855, 78F0856
- 78K0/FA2-L: μPD78F0857, 78F0858, 78F0859
- 78K0/FB2-L: μPD78F0864, 78F0865

Purpose

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

Organization

The manual for the 78K0/Fx2-L microcontrollers is separated into two parts: this manual and the instructions edition (common to the 78K0 microcontrollers).

78K0/Fx2-L 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/Fx2-L User's Manual	U19856E
78K/0 Series Instructions User's Manual	U12326E
78K0 Microcontrollers User's Manual Self Programming Library Type 01	U18274E
78K0 Microcontrollers Self Programming Library Type 01 Ver. 3.10 Operating Precautions (Notification Document)	ZUD-CD-09-0122-E
78K0 Microcontrollers User's Manual EEPROM™ Emulation Library Type 01	U18275E
78K0 Microcontrollers EEPROM Emulation Library Type 01 Ver.2.10 Operating Precautions (Notification Document)	ZUD-CD-09-0165-E

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

Document Name		Document No.
QB-78K0FX2L In-Circuit Emulator		To be prepared
QB-MINI2 On-Chip Debug Emulator with Programming Function		U18371E
QB-Programmer Programming GUI	Operation	U18527E

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

Document Name	Document No.
PG-FP5 Flash Memory Programmer	U18865E

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

Documents Related to Development Tools (Software)

Document Name		Document No.
RA78K0 Ver.3.80 Assembler Package User's Manual Note 1	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
78K0 Assembler Package RA78K0 Ver.4.01 Operating Precautions (Notification Document) Note 1		ZUD-CD-07-0181-E
CC78K0 Ver.3.70 C Compiler User's Manual Note 2	Operation	U17201E
	Language	U17200E
78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions (Notification Document) Note 2		ZUD-CD-07-0103-E
SM+ System Simulator User's Manual	Operation	U18601E
	User Open Interface	U18212E
ID78K0-QB Ver.2.94 Integrated Debugger User's Manual	Operation	U18330E
ID78K0-QB Ver.3.00 Integrated Debugger User's Manual	Operation	U18492E
PM plus Ver.5.20 Note 3 User's Manual		U16934E
PM+ Ver.6.30 Note 4 User's Manual		U18416E

Notes 1. This document is installed into the PC together with the tool when installing RA78K0 Ver. 4.01. For descriptions not included in "78K0 Assembler Package RA78K0 Ver. 4.01 Operating Precautions", refer to the user's manual of RA78K0 Ver. 3.80.

2. This document is installed into the PC together with the tool when installing CC78K0 Ver. 4.00. For descriptions not included in "78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions", refer to the user's manual of CC78K0 Ver. 3.70.

3. PM plus Ver. 5.20 is the integrated development environment included with RA78K0 Ver. 3.80.

4. PM+ Ver. 6.30 is the integrated development environment included with RA78K0 Ver. 4.01. Software tool (assembler, C compiler, debugger, and simulator) products of different versions can be managed.

Other Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE – Products and Packages –	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www2.renesas.com/pkg/en/mount/index.html).

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

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

- 78K0 CPU core
- I/O ports, ROM and RAM capacities

Item / Products	I/O Ports	Program Memory (Flash Memory)	Data Memory (Internal High-Speed RAM)
78K0/FY2-L (16 pins)	11 (CMOS I/O: 9, CMOS input: 2)	4 KB to 16 KB	384 bytes to 768 bytes
78K0/FA2-L (20 pins)	15 (CMOS I/O: 13, CMOS input: 2)	8 KB and 16 KB	512 bytes and 768 bytes
78K0/FB2-L (30 pins)	24 (CMOS I/O: 22, CMOS input: 2)		

- Low power consumption (V_{DD} = 3.0 V, T_A = –40 to +85°C)
  - Internal high-speed oscillator operation mode: 220 μA (TYP.) (f_{CPU} = 1 MHz operation)
  - STOP mode: 0.65 μA (TYP.) (f_{IL} = 30 kHz operation)
- Clock
  - High-speed system clock … Selected from the following three sources
    - Ceramic/crystal resonator: 2 to 20 MHz
    - External clock: 2 to 20 MHz
    - Internal high-speed oscillator:
      4 MHz ±2% (–20 to +70°C),
      4 MHz ±3% (–40 to +85°C),
      or 8 MHz ±3% (–40 to +85°C)
  - Internal low-speed oscillator 30 kHz (TYP.) … Watchdog timer, timer clock in intermittent operation

- Timer
  - 16-bit timer X … PWM output, operation in conjunction with an external signal, synchronous output of up to four channels (available only in 78K0/FB2-L), A/D conversion trigger generation
  - 16-bit timer/event counter … PPG output, capture input, external event counter input
  - 8-bit timer H1 … PWM output, operable with low-speed internal oscillation clock
  - 8-bit timer/event counter 51 … PWM output, external event counter input
  - Watchdog timer … Operable with internal low-speed oscillation clock

Item / Products	16-bit Timer	16-bit Timer/ Event Counter	8-bit Timer	Watchdog Timer
78K0/FY2-L (16 pins)	1 ch	1 ch	Timer H1: 1 ch Timer 51: 1 ch	1 ch
78K0/FA2-L (20 pins)				
78K0/FB2-L (30 pins)	2 ch			

- Serial interface
  - UART6 … Asynchronous 2-wire serial interface
  - IICA … Clock synchronous 2-wire serial interface, multimaster supported, standby can be released upon address match in slave mode
  - CSI11 … Clock synchronous 3-wire serial interface, operable as SPI in slave mode

Item / Products	UART6	IICA	CSI11
78K0/FY2-L (16 pins)	1 ch	1 ch	–
78K0/FA2-L (20 pins)			
78K0/FB2-L (30 pins)			1 ch

- Multiplier (8 bits × 8 bits = 16 bits, 16 bits × 16 bits = 32 bits, 1-clock operation)
- 10-bit resolution A/D conversion
  - 78K0/FY2-L: 4 ch
  - 78K0/FA2-L: 6 ch
  - 78K0/FB2-L: 9 ch
- Comparator
  - 78K0/FY2-L: 1 ch
  - 78K0/FA2-L: 3 ch
  - 78K0/FB2-L: 3 ch
- Power-on-clear (POC) circuit
- Low-voltage detector (LVI) circuit (An interrupt/reset (selectable) is generated when the detection voltage is reached))
  - Detection voltage: Selectable from sixteen levels between 1.91 and 4.22 V
- Single-power-supply flash memory
  - Flash self programming enabled
  - Software protection function: Protected from outside party copying (no flash reading command)
- Safety function
  - Watchdog timer operated by clock independent from CPU
    … A hang-up can be detected even if the system clock stops
  - Supply voltage drop detectable by LVI
    … Appropriate processing can be executed before the supply voltage drops below the operation voltage
  - Equipped with option byte function
    … Important system operation settings set in hardware
- On-chip debug function …Available to control for the target device, and to reference memory
- Assembler and C language supported
- Enhanced development environment
  - Support for full-function emulator (IECUBE), and simplified emulator (MINICUBE2)
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature:
  - (A) grade products: T_A = –40 to +85°C
  - (A2) grade products: T_A = –40 to +125°C

1.2 Ordering Information

[Part Number]

μPD78F08xy ΔΔ × - ××× -×

	Semiconductor	
G	Lead-free	Product contains no lead in any area (Terminal finish is Ni/Pd/Au plating)

	Quality Grade
A	Special (T_A = –40 to +85°C)
A2	Special (T_A = –40 to +125°C)

xy	ΔΔ - xxx	Package Type
54, 55, 56 (FY2-L)	MA-FAA	16-pin plastic SSOP (5.72 mm (225))
57, 58, 59 (FA2-L)	MC-CAA	20-pin plastic SSOP (7.62 mm (300))
64, 65 (FB2-L)	MC-CAB	30-pin plastic SSOP (7.62 mm (300))

xy	High-speed RAM Capacity	Flash Memory Capacity
54, 57	384 bytes	4 KB
55, 58, 64	512 bytes	8 KB
56, 59, 65	768 bytes	16 KB

	Product Type
F	Flash memory version

[Example of Part Number]

μPD78F08 54 MA-A-FAA-G

- Lead-free
- Special
- 16-pin plastic SSOP (5.72 mm (225))
- High-speed RAM: 384 bytes, flash Memory: 4 KB
- Flash memory version

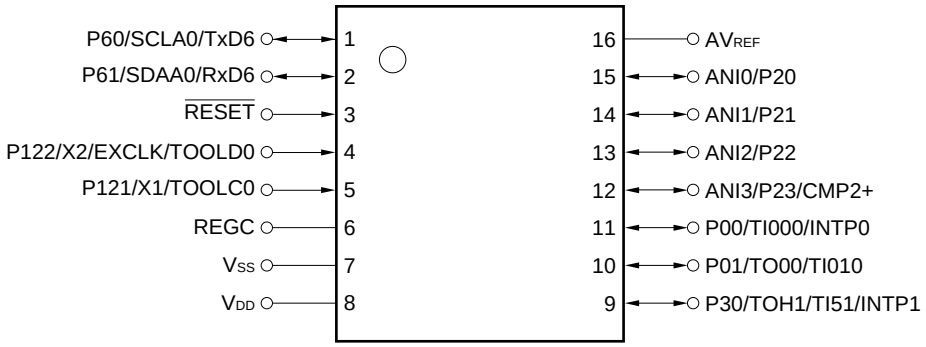
[List of Part Number]

78K0/Fx2-L microcontrollers	Package	Part Number
78K0/FY2-L	16-pin plastic SSOP (5.72 mm (225))	μPD78F0854MAA-FAA-G, 78F0855MAA-FAA-G, 78F0856MAA-FAA-G, 78F0854MAA2-FAA-G, 78F0855MAA2-FAA-G, 78F0856MAA2-FAA-G
78K0/FA2-L	20-pin plastic SSOP (7.62 mm (300))	μPD78F0857MCA-CAA-G, 78F0858MCA-CAA-G, 78F0859MCA-CAA-G, 78F0857MCA2-CAA-G, 78F0858MCA2-CAA-G, 78F0859MCA2-CAA-G
78K0/FB2-L	30-pin plastic SSOP (7.62 mm (300))	μPD78F0864MCA-CAB-G, 78F0865MCA-CAB-G, 78F0864MCA2-CAB-G, 78F0865MCA2-CAB-G

1.3 Pin Configuration (Top View)

1.3.1 78K0/FY2-L (16 pins)

- 16-pin plastic SSOP (5.72 mm (225))



ANI0 to ANI3:	Analog Input	RESET:	Reset
AVREF:	Analog Reference Voltage	RxD6:	Receive Data
		SCLA0:	Serial Clock Input/Output
CMP2+:	Comparator Input	SDAA0:	Serial Data Input/Output
EXCLK:	External Clock Input (Main System Clock)	TI000, TI010, TI51:	Timer Input
		TO00, TOH1:	Timer Output
INTP0, INTP1:	External Interrupt Input	TOOLC0:	Clock Input for Tool
		TOOLD0:	Data Input/Output for Tool
P00, P01:	Port 0	TxD6:	Transmit Data
P20 to P23:	Port 2	VDD:	Power Supply
P30:	Port 3	VSS:	Ground
P60, P61:	Port 6	X1, X2:	Crystal Oscillator (Main System Clock)
P121, P122:	Port 12		
REGC:	Regulator Capacitance		

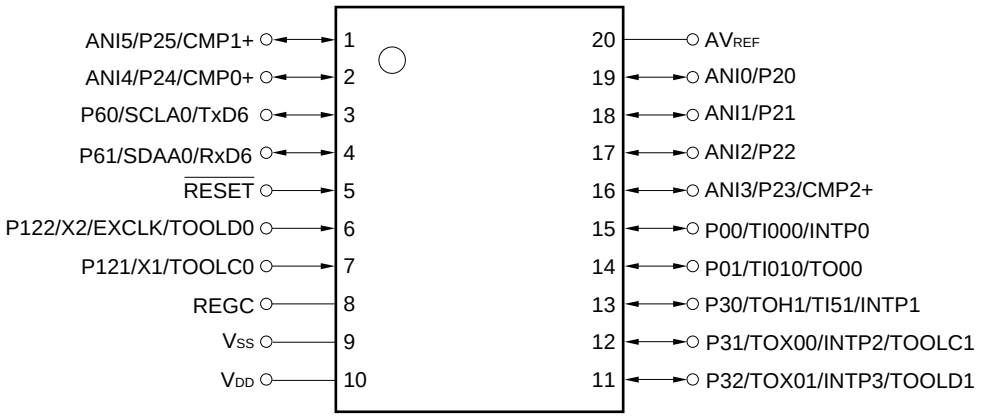
Cautions 1. VSS functions alternately as the ground potential of the A/D converter. Be sure to connect VSS to a stabilized GND (= 0 V).

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

3. ANI0/P20, ANI1/P21, ANI2/P22, and ANI3/P23/CMP2+ are set in the analog input mode after release of reset.

1.3.2 78K0/FA2-L (20 pins)

- 20-pin plastic SSOP (7.62 mm (300))



ANI0 to ANI5:	Analog Input	$\overline{\text{RESET}}$:	Reset
AVREF:	Analog Reference Voltage	RxD6:	Receive Data
		SCLA0:	Serial Clock Input/Output
CMP0+ to CMP2+:	Comparator Input	SDAA0:	Serial Data Input/Output
EXCLK:	External Clock Input (Main System Clock)	TI000, TI010, TI51:	Timer Input
		TO00, TOH1:	Timer Output
INTP0 to INTP3:	External Interrupt Input	TOOLC0, TOOLC1:	Clock Input for Tool
		TOOLD0, TOOLD1:	Data Input/Output for Tool
P00, P01:	Port 0	TOX00, TOX01:	Timer Output
P20 to P25:	Port 2	TxD6:	Transmit Data
P30 to P32:	Port 3	VDD:	Power Supply
P60, P61:	Port 6	VSS:	Ground
P121, P122:	Port 12	X1, X2:	Crystal Oscillator (Main System Clock)
REGC:	Regulator Capacitance		

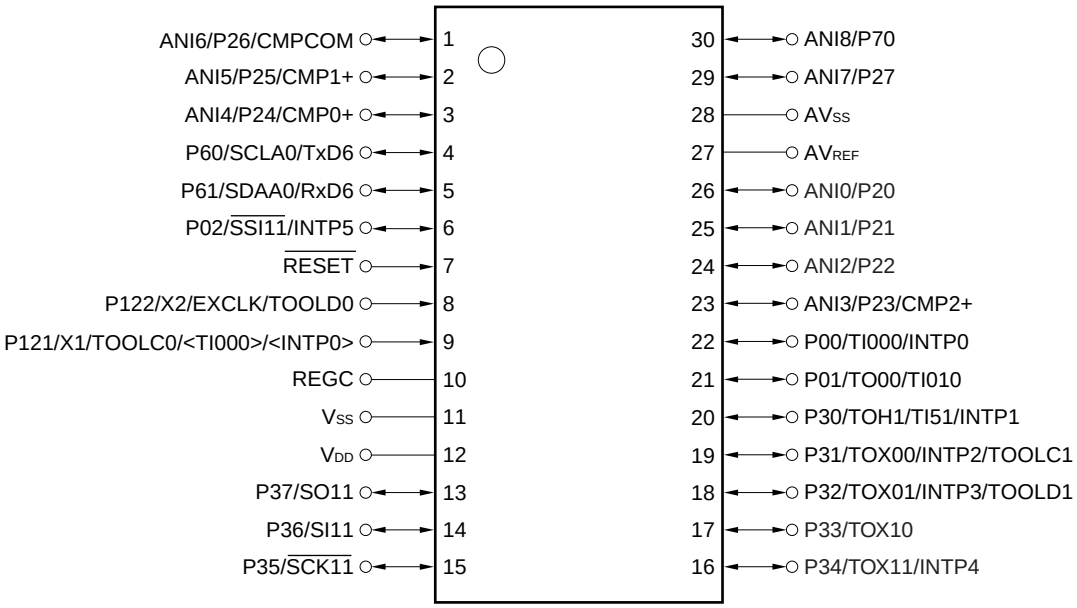
Cautions 1. VSS functions alternately as the ground potential of the A/D converter. Be sure to connect VSS to a stabilized GND (= 0 V).

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

3. ANI0/P20, ANI1/P21, ANI2/P22, ANI3/P23/CMP2+, ANI4/P24/CMP0+, and ANI5/P25/CMP1+ are set in the analog input mode after release of reset.

1.3.3 78K0/FB2-L (30 pins)

- 30-pin plastic SSOP (7.62 mm (300))



ANI0 to ANI8:	Analog Input	RxD6:	Receive Data
AVREF:	Analog Reference Voltage	SCLA0, SCK11:	Serial Clock Input/Output
AVSS:	Analog Ground	SDAA0:	Serial Data Input/Output
CMP0+ to CMP2+:	Comparator Input	SI11:	Serial Data Input
EXCLK:	External Clock Input	SO11:	Serial Data Output
	(Main System Clock)	SSI11:	Serial Interface Chip
CMPCOM:	Comparator Common Input	TI000, TI010, TI51:	Timer Input
INTP0 to INTP5:	External Interrupt Input	TO00, TOH1:	Timer Output
P00 to P02:	Port 0	TOOLC0, TOOLC1:	Clock Input for Tool
P20 to P27:	Port 2	TOOLD0, TOOLD1:	Data Input/Output for Tool
P30 to P37:	Port 3	TOX00, TOX01,	
P60, P61:	Port 6	TOX10, TOX11:	Timer Output
P70:	Port 7	TxD6:	Transmit Data
P121, P122:	Port 12	VDD:	Power Supply
REGC:	Regulator Capacitance	VSS:	Ground
RESET:	Reset	X1, X2:	Crystal Oscillator
			(Main System Clock)

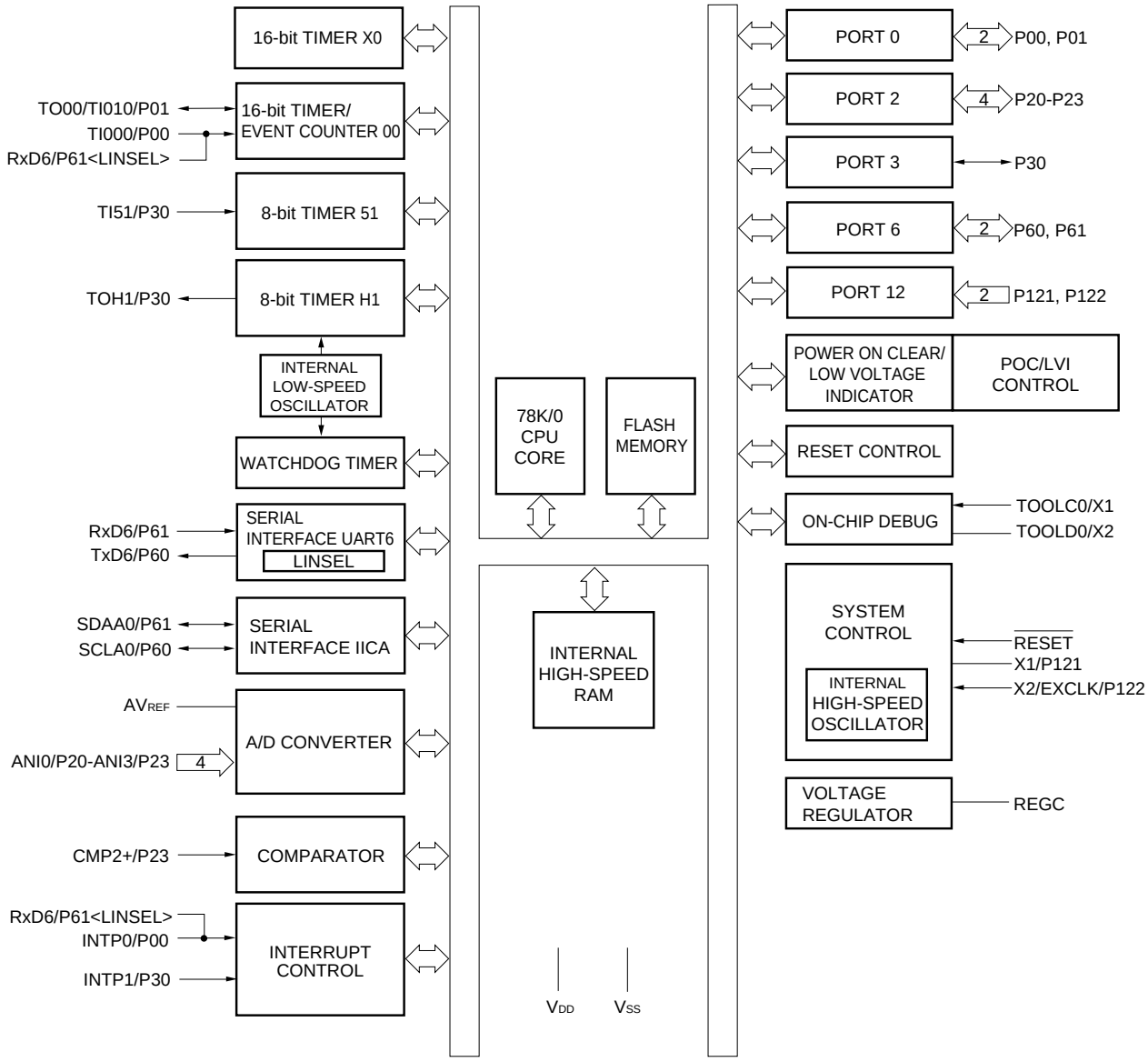
Cautions 1. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).

2. ANI0/P20, ANI1/P21, ANI2/P22, ANI3/P23/CMP2+, ANI4/P24/CMP0+, ANI5/P25/CMP1+, ANI6/P26/CMPCOM, ANI7/P27, and ANI8/P70 are set in the analog input mode after release of reset.

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

1.4 Block Diagram

1.4.1 78K0/FY2-L (16 pins)

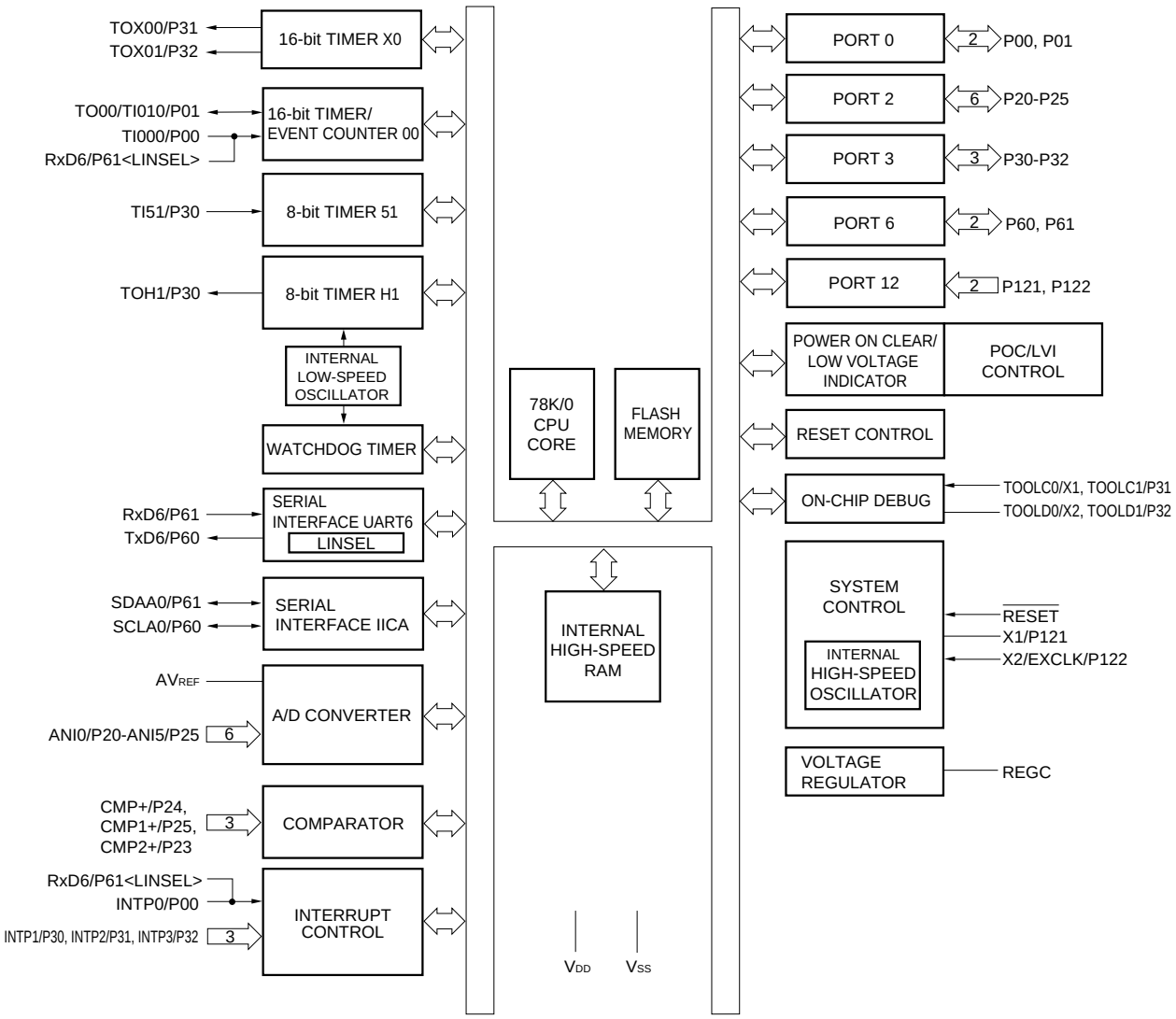


Cautions 1. VSS functions alternately as the ground potential of the A/D converter. Be sure to connect VSS to a stabilized GND (= 0 V).

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

3. ANI0/P20, ANI1/P21, ANI2/P22, and ANI3/P23/CMP2+ are set in the analog input mode after release of reset.

1.4.2 78K0/FA2-L (20 pins)

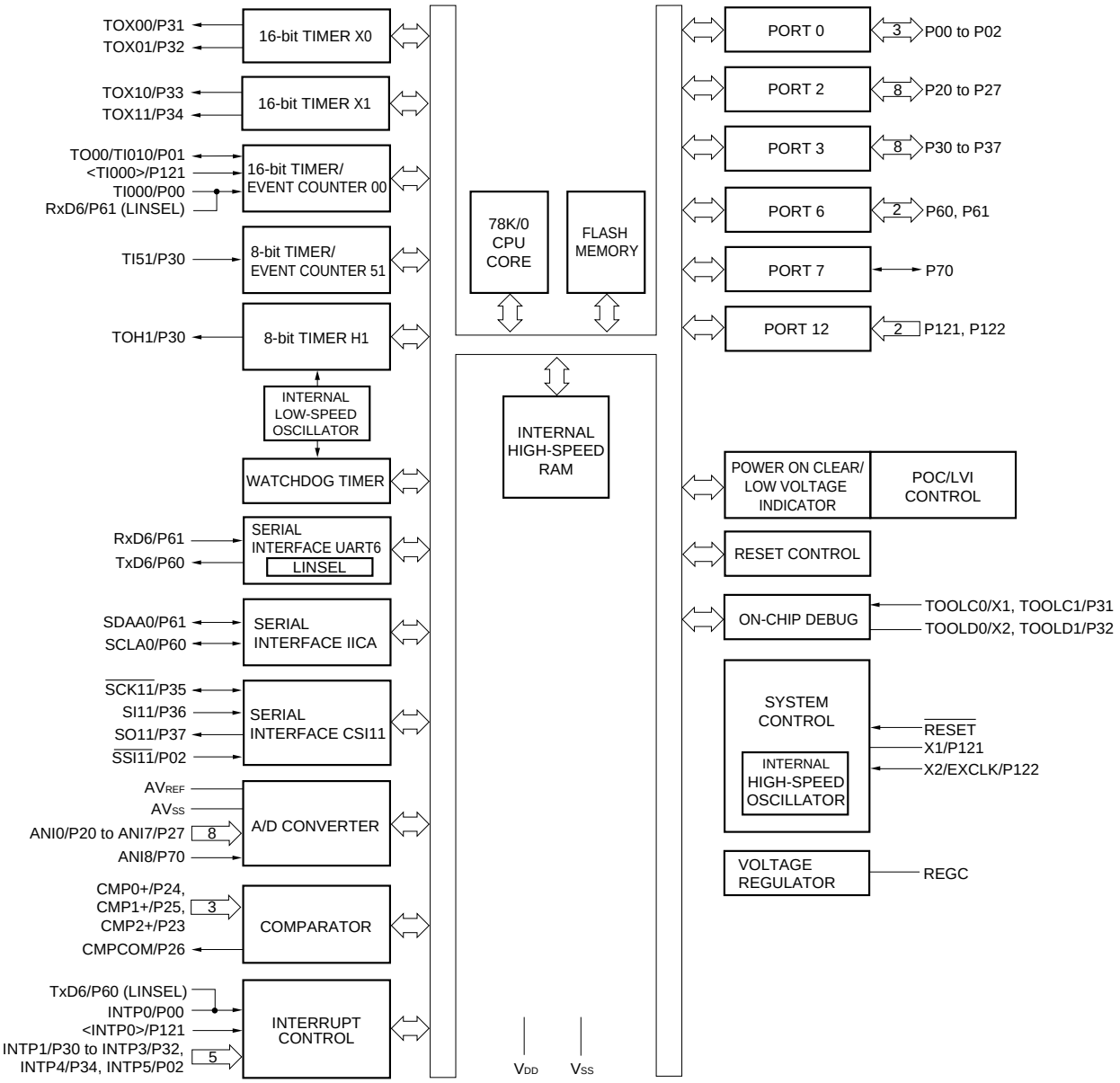


Cautions 1. VSS functions alternately as the ground potential of the A/D converter. Be sure to connect VSS to a stabilized GND (= 0 V).

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

3. ANI0/P20, ANI1/P21, ANI2/P22, ANI3/P23/CMP2+, ANI4/P24/CMP0+, and ANI5/P25/CMP1+ are set in the analog input mode after release of reset.

1.4.3 78K0/FB2-L (30 pins)



Cautions 1. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).

2. ANI0/P20, ANI1/P21, ANI2/P22, ANI3/P23/CMP2+, ANI4/P24/CMP0+, ANI5/P25/CMP1+, ANI6/P26/CMPCOM, ANI7/P27, and ANI8/P70 are set in the analog input mode after release of reset.

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

1.5 Outline of Functions

(1/2)

Item			78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
			16 pins	20 pins	30 pins
Internal memory		Flash memory (self-programming supported)	4 KB to 16 KB		8 KB and 16 KB
		High-Speed RAM	384 bytes to 768 bytes		512 bytes and 768 bytes
Memory space			64 KB		
Clock	Main	High-speed system (crystal/ceramic oscillation, external clock input)	2 to 20 MHz[Note 1]: V_{DD} = 4.0 to 5.5 V, 2 to 10 MHz: V_{DD} = 2.7 to 5.5 V, or 2 to 5 MHz: V_{DD} = 1.8 to 5.5 V[Note 2]		
		Internal high-speed oscillation	4 MHz ±2% (T_A = –20 to +70°C)[Note 1], 4 MHz ±3% (T_A = –40 to +85°C)[Note 1], or 8 MHz ±3% (T_A = –40 to +85°C): V_{DD} = 1.8 to 5.5 V[Note 2]		
	Internal low-speed oscillation		30 kHz (TYP.): V_{DD} = 1.8 to 5.5 V[Note 2]		
General-purpose registers			8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Instruction set			• 8-bit operation, 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.		
I/O ports			11 (CMOS I/O: 9, CMOS input: 2)	15 (CMOS I/O: 13, CMOS input: 2)	24 (CMOS I/O: 22, CMOS input: 2)
Timer	16 bits (TMX)		1 ch	1 ch (PWM output: 2)	2 ch (PWM output: 4)
	16 bits (TM0)		1 ch (capture input: 1)		1 ch (PPG output: 1, capture input: 2)
	8 bits (TM51)		1 ch		
	8 bits (TMH1)		1 ch (PWM output: 1)		
	Watchdog (WDT)		1 ch		

Notes 1. When using a 4 MHz clock, operation at 20 MHz is possible by using PLL.

2. This is applicable to a (A) grade product. See **CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)** for products with (A2) grade product.

(2/2)

Item			78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
			16 pins	20 pins	30 pins
Serial interface	UART6		1 ch		
	IICA		1 ch		
	CSI11		–		1 ch
10-bit A/D converter			4 ch	6 ch	9 ch
Comparator			1 ch	3 ch	
Multiplier			8 bits × 8 bits = 16 bits, 16 bits × 16 bits = 32 bits		
Vectored interrupt sources		External	3	7	9
		Internal	11	11	13
Reset			• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear (POC) • Internal reset by low-voltage detector (LVI)		
On-chip debug function			Provided		
Power supply voltage			V_{DD} = 1.8 to 5.5 V		
Operating ambient temperature			(A) grade products: T_A = –40 to +85°C, (A2) grade products: T_A = –40 to +125°C		
Package			16-pin plastic SSOP (5.72 mm (225))	20-pin plastic SSOP (7.62 mm (300))	30-pin plastic SSOP (7.62 mm (300))

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

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

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

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P70[Note]
V_{DD}	Pins other than P20 to P27, P70 [Note]

Note 78K0/FY2-L: P20 to P23
78K0/FA2-L: P20 to P25
78K0/FB2-L: P20 to P27, P70

2.1.1 78K0/FY2-L

(1) Port functions: 78K0/FY2-L

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/INTP0
P01				TO00/TI010
P20	I/O	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0
P21				ANI1
P22				ANI2
P23				ANI3/CMP2+
P30	I/O	Port 3. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOH1/TI51/INTP1
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Input can be set to SMBus input buffer in 1-bit units. Output can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA0/TxD6
P61				SDAA0/RxD6
P121	Input	Port 12. 2-bit input-only port.	Input port	X1/TOOLC0
P122				X2/EXCLK/TOOLD0

(2) Non-port functions: 78K0/FY2-L

Function Name	I/O	Function	After Reset	Alternate Function
ANI0	Input	A/D converter analog input	Analog input	P20
ANI1				P21
ANI2				P22
ANI3				P23
CMP2+	Input	Comparator input	Analog input	P23/ANI31
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	P00/TI000
INTP1				P30/TOH1/TI51
REGC	–	Connecting regulator output (2.0 V/2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	Reset Input	–
RxD6	Input	Serial data input to UART6	Input port	P61/SDAA0
TxD6	Output	Serial data output from UART6		P60/SCLA0
SCLA0	I/O	Clock input/output for I^2C	Input port	P60/TxD6
SDAA0		Serial data I/O for I^2C		P61/RxD6
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/INTP0
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TI51	Input	External count clock input to 8-bit timer/event counter 51	Input port	P30/TOH1/INTP1
TO00	Output	16-bit timer/event counter 00 output	Input port	P01/TI010
TOH1	Output	8-bit timer H1 output	Input port	P30/TI51/INTP1
X1	–	Connecting resonator for main system clock	Input port	P121/TOOLC0
X2				P122/EXCLK/TOOLD0
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/TOOLD0
V_{DD}	–	Positive power supply for pins other than port 2	–	–
AV_{REF}		A/D converter reference voltage input and positive power supply for port 2 and A/D converter		
V_{SS}	–	Ground potential	–	–
TOOLC0	Input	Clock input for flash memory programmer/on-chip debugger	Input port	P121/X1
TOOLD0	I/O	Data I/O for flash memory programmer/on-chip debugger		P122/X2/EXCLK

2.1.2 78K0/FA2-L

(1) Port functions: 78K0/FA2-L

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/INTP0
P01				TO00/TI010
P20	I/O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0
P21				ANI1
P22				ANI2
P23				ANI3/CMP2+
P24				ANI4/CMP0+
P25				ANI5/CMP1+
P30	I/O	Port 3. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOH1/TI51/INTP1
P31				TOX00/INTP2/ TOOLC1
P32				TOX01/INTP3/ TOOLD1
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Input can be set to SMBus input buffer in 1-bit units. Output can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA0/TxD6
P61				SDAA0/RxD6
P121	Input	Port 12. 2-bit input-only port.	Input port	X1/TOOLC0
P122				X2/EXCLK/TOOLD0

(2) Non-port functions : 78K0/FA2-L (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
ANI0	Input	A/D converter analog input	Analog input	P20
ANI1				P21
ANI2				P22
ANI3				P23
ANI4				P24
ANI5				P25

(2) Non-port functions : 78K0/FA2-L (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
CMP0+	Input	Comparator input	Analog input	P24/ANI4
CMP1+				P25/ANI5
CMP2+				P21/ANI1
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	P00/TI000
INTP1				P30/TOH1/TI51
INTP2				P31/TOOLC1
INTP3				P32/TOOLD1
REGC	–	Connecting regulator output (2.0 V/2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	Reset Input	–
RxD6	Input	Serial data input to UART6	Input port	P61/SDAA0
TxD6	Output	Serial data output from UART6		P60/SCLA0
SCLA0	I/O	Clock input/output for I^2C	Input port	P60/TxD6
SDAA0		Serial data I/O for I^2C		P61/RxD6
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/INTP0
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TI51	Input	External count clock input to 8-bit timer/event counter 51	Input port	P30/TOH1/INTP1
TO00	Output	16-bit timer/event counter 00 output	Input port	P01/TI010
TOH1	Output	8-bit timer H1 output	Input port	P30/TI51/INTP1
TOX00	Output	16-bit timer X0 output	Input port	P31/INTP2/TOOLC1
TOX01				P32/INTP3/TOOLD1
X1	–	Connecting resonator for main system clock	Input port	P121/TOOLC0
X2				P122/EXCLK/TOOLD0
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/TOOLD0
V_{DD}	–	Positive power supply for pins other than port 2	–	–
AV_{REF}		A/D converter reference voltage input and positive power supply for port 2 and A/D converter		
V_{SS}	–	Ground potential	–	–
TOOLC0	Input	Clock input for flash memory programmer/on-chip debugger	Input port	P121/X1
TOOLC1				P31/INTP2
TOOLD0	I/O	Data I/O for flash memory programmer/on-chip debugger		P122/X2/EXCLK
TOOLD1				P32/INTP3

2.1.3 78K0/FB2-L

(1) Port functions: 78K0/FB2-L

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI000/INTP0
P01				TO00/TI010
P02				$\overline{\text{SSI11}}$/INTP5
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0
P21				ANI1
P22				ANI2
P23				ANI3/CMP2+
P24				ANI4/CMP0+
P25				ANI5/CMP1+
P26				ANI6/CMPCOM
P27				ANI7
P30	I/O	Port 3. 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	TOH1/TI51/INTP1
P31				TOX00/INTP2/TOOLC1
P32				TOX01/INTP3/TOOLD1
P33				TOX10
P34				TOX11/INTP4
P35				$\overline{\text{SCK11}}$
P36				SI11
P37				SO11
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Input can be set to SMBus input buffer in 1-bit units. Output can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA0/TxD6
P61				SDAA0/RxD6
P70	I/O	Port 7. 1-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI8
P121	Input	Port 12. 2-bit input port.	Input port	X1/TOOLC0/ <TI000>/<INTP0>
P122				X2/EXCLK/TOOLD0

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

(2) Non-port functions : 78K0/FB2-L (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
ANI0	Input	A/D converter analog input	Analog input	P20
ANI1				P21
ANI2				P22
ANI3				P23/CMP2+
ANI4				P24/CMP0+
ANI5				P25/CMP1+
ANI6				P26/CMPCOM
ANI7				P27
ANI8				P70
CMP0+	Input	Comparator input	Analog input	P24/ANI4
CMP1+				P25/ANI5
CMP2+				P23/ANI3
CMPCOM	Input	Comparator common input	Analog input	P26/ANI6
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	P00/TI000
				P121/X1/TOOLC0/ <TI000>
INTP1				P30/TOH1/TI51
INTP2				P31/TOX00/TOOLC1
INTP3				P32/TOX01/TOOLD1
INTP4				P34/TOX11
INTP5				P02/$\overline{\text{SSI11}}$
REGC	–	Connecting regulator output (2.0 V/2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	Reset input	–
RxD6	Input	Serial data input to UART6	Input port	P61/SDAA0
TxD6	Output	Serial data output from UART6		P60/SCLA0
SCLA0	I/O	Clock input/output for I^2C	Input port	P60/TxD6
SDAA0		Serial data I/O for I^2C		P61/RxD6

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

(2) Non-port functions : 78K0/FB2-L (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
$\overline{SCK11}$	I/O	Clock input/output for CSI11	Input port	P35
SI11	Input	Serial data input to CSI11		P36
SO11	Output	Serial data output from CSI11		P37
$\overline{SSI11}$	Input	Chip select input to CSI11	Input port	P02/INTP5
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/INTP0
				P121/X1/TOOLC0/<INTP0>
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TI51	Input	External count clock input to 8-bit timer/event counter 51	Input port	P30/TOH1/INTP1
TO00	Output	16-bit timer/event counter 00 output	Input port	P01/TI010
TOH1	Output	8-bit timer H1 output	Input port	P30/TI51/INTP1
TOX00	Output	16-bit timer X0 output	Input port	P31/INTP2/TOOLC1
TOX01				P32/INTP3/TOOLD1
TOX10		16-bit timer X1 output		P33
TOX11				P34/INTP4
X1	–	Connecting resonator for main system clock	Input port	P121/TOOLC0/<TI000>/<INTP0>
X2				P122/EXCLK/TOOLD0
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/TOOLD0
V_{DD}	–	Positive power supply for pins other than ports 2, 7	–	–
AV_{REF}		A/D converter reference voltage input and positive power supply for ports 2, 7 and A/D converter		
V_{SS}	–	Ground potential for pins other than ports 2, 7	–	–
AV_{SS}		Ground potential for ports 2, 7 and A/D converter		
TOOLC0	Input	Clock input for flash memory programmer/on-chip debugger	Input port	P121/X1/<TI000>/<INTP0>
TOOLC1				P31/TOX00/INTP2
TOOLD0	I/O	Data I/O for flash memory programmer/on-chip debugger		P122/X2/EXCLK
TOOLD1				P32/TOX01/INTP3

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

2.2 Description of Pin Functions

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

2.2.1 P00 to P02 (port 0)

P00 to P02 function as an I/O port. These pins also function as timer I/O, external interrupt request input, and chip select input.

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
P00/TI000/INTP0	P00/TI000/INTP0	P00/TI000
P01/TO00/TI010	P01/TO00/TI010	P01/TO00/TI010
–	–	P02/$\overline{\text{SSI11}}$/INTP5

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

(1) Port mode

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

(2) Control mode

P00 to P02 function as timer I/O, external interrupt request input, and chip select input.

(a) TI000

This is the pin for inputting an external count clock to 16-bit timer/event counter 00 and is also for inputting a capture trigger signal to the capture registers (CR000, CR010) of 16-bit timer/event counter 00.

(b) TI010

This is the pin for inputting a capture trigger signal to the capture register (CR000) of 16-bit timer/event counter 00.

(c) TO00

This is a timer output pin of 16-bit timer/event counter 00.

(d) INTP0, INTP5

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

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

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

2.2.2 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 and comparator input.

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
P20/ANI0	P20/ANI0	P20/ANI0
P21/ANI1	P21/ANI1	P21/ANI1
P22/ANI2	P22/ANI2	P22/ANI2
P23/ANI3/CMP2+	P23/ANI3/CMP2+	P23/ANI3/CMP2+
–	P24/ANI4/CMP0+	P24/ANI4/CMP0+
–	P25/ANI5/CMP1+	P25/ANI5/CMP1+
–	–	P26/ANI6/CMPCOM
–	–	P27/ANI7

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

(1) Port mode

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

(2) Control mode

P20 to P27 function as A/D converter analog input and comparator input.

(a) ANI0 to ANI7

These are A/D converter analog input pins. When using these pins as analog input pins, refer to **(5) ANI0/P20 to ANI7/P27 and ANI8/P10 to ANI10/P12** in **11.6 Cautions for A/D Converter**.

(b) CMP0+ to CMP2+

These are comparator input pins.

(c) CMPCOM

This is a comparator common input pin.

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

2.2.3 P30 to P37 (port 3)

P30 to P37 function as an I/O port. These pins also function as pins for external interrupt request input, timer I/O, clock input and data I/O for flash memory programmer/on-chip debugger, and clock input and data I/O for serial interface.

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
P30/TOH1/TI51/INTP1	P30/TOH1/TI51/INTP1	P30/TOH1/TI51/INTP1
–	P31/TOX00/INTP2/TOOLC1	P31/TOX00/INTP2/TOOLC1
–	P32/TOX01/INTP3/TOOLD1	P32/TOX01/INTP3/TOOLD1
–	–	P33/TOX10
–	–	P34/TOX11/INTP4
–	–	P35/$\overline{\text{SCK11}}$
–	–	P36/SI11
–	–	P37/SO11

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

(1) Port mode

P30 to P37 function as an I/O port. P30 to P37 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 P37 function as external interrupt request input, timer I/O, clock input and data I/O for flash memory programmer/on-chip debugger, and clock input for serial interface.

(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) TI51

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

(c) TO51

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

(d) TOH1

This is the timer output pin of 8-bit timer H1.

(e) TOX00, TOX01

These are the timer output pins of 16-bit timer X0.

(f) TOX10, TOX11

These are the timer output pins of 16-bit timer X1.

(g) TOOLC1

This is the clock input pin for flash memory programmer/on-chip debugger.

(h) TOOLD1

This is the data I/O pin for flash memory programmer/on-chip debugger.

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

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

(j) SI11

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

(k) SO11

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

Remark For how to connect a flash memory programmer using TOOLC1/P31, TOOLD1/P32, refer to **CHAPTER 24 FLASH MEMORY**. For how to connect TOOLC1/P31, TOOLD1/P32 and an on-chip debug emulator, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

2.2.4 P60 and P61 (port 6)

P60 and P61 function as an I/O port. These pins also function as pins for serial interface data I/O and clock I/O.

Input to the P60 and P61 pins can be specified through a normal input buffer or an SMBus input buffer in 1-bit units, using port input mode register 6 (PIM6).

Output from the P60 and P61 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 6 (POM6).

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
P60/SCLA0/TxD6	P60/SCLA0/TxD6	P60/SCLA0/TxD6
P61/SDAA0/RxD6	P61/SDAA0/RxD6	P61/SDAA0/RxD6

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

(1) Port mode

P60 and P61 function as an I/O port. P60 and P61 can be set to input port or output port in 1-bit units using port mode register 6 (PM6). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

(2) Control mode

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

(a) SDAA0

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

(b) SCLA0

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

(c) RxD6

This is a serial data input pin for serial interface UART6.

(d) TxD6

This is a serial data output pin for serial interface UART6.

2.2.5 P70 (port 7)

P70 functions as an I/O port. This pin also functions as the pin for A/D converter analog input.

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
–	–	P70/ANI8

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

(1) Port mode

P70 functions as an I/O port. P70 can be set to input or output port in 1-bit units using port mode register 7 (PM7).

(2) Control mode

P70 functions as A/D converter analog input.

(a) ANI8

This is an A/D converter analog input pin. When using this pin as analog input pin, refer to **(5) ANI0/P20 to ANI7/P27 and ANI8/P70** in **11.6 Cautions for A/D Converter**.

Cautions 1. ANI8/P70 is set in the analog input mode after release of reset.

2. Make the AV_{REF} pin the same potential as the V_{DD} pin when ANI8 is used.

2.2.6 P121 and P122 (port 12)

P121 and P122 function as an input port. These pins also function as pins for external interrupt request input, connecting resonator for main system clock, external clock input for main system clock, timer input, and clock input and data I/O for flash memory programmer/on-chip debugger.

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
P121/X1/TOOLC0	P121/X1/TOOLC0	P121/X1/TOOLC0/<TI000>/<INTP0>
P122/X2/EXCLK/TOOLD0	P122/X2/EXCLK/TOOLD0	P122/X2/EXCLK/TOOLD0

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

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

(1) Port mode

P121 and P122 function as an input port.

(2) Control mode

P121 and P122 function as pins for external interrupt request input, connecting resonator for main system clock, external clock input for main system clock, timer input, and clock input and data I/O for flash memory programmer/on-chip debugger.

(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) X1, X2

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

(c) EXCLK

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

(d) TI000

This is the pin for inputting an external count clock to 16-bit timer/event counter 00 and is also for inputting a capture trigger signal to the capture registers (CR000, CR010) of 16-bit timer/event counter 00.

(e) TOOLC0

This is the clock input pin for flash memory programmer/on-chip debugger.

(f) TOOLD0

This is the data I/O pin for flash memory programmer/on-chip debugger.

Remark For how to connect a flash memory programmer using TOOLC0/X1, TOOLD0/X2, refer to **CHAPTER 24 FLASH MEMORY**. For how to connect TOOLC0/X1, TOOLD0/X2 and an on-chip debug emulator, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

2.2.7 AV_{REF}, AV_{SS}, V_{DD}, V_{SS}

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
AV_{REF}	AV_{REF}	AV_{REF}
–	–	AV_{SS}
V_{DD}	V_{DD}	V_{DD}
V_{SS}	V_{SS}	V_{SS}

(a) AV_{REF}

This is the A/D converter reference voltage input pin and the positive power supply pin of port 2 and A/D converter.

When the A/D converter is not used, connect this pin directly to 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 ground potential pin of A/D converter and port 2. 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}

V_{DD} is the positive power supply pin.

(d) V_{SS}

V_{SS} is the ground potential pin[Note].

Note In the 78K0/FY2-L and 78K0/FA2-L, V_{SS} functions alternately as the ground potential of the A/D converter. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

2.2.8 $\overline{\text{RESET}}$

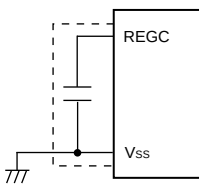
This is the active-low system reset input pin.

2.2.9 REGC

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
REGC	REGC	REGC

(a) REGC

This is the pin for connecting regulator output (2.0 V/2.4 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



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

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

Tables 2-2 to 2-4 show the types of pin I/O circuits and the recommended connections of unused pins.
Refer to **Figure 2-1** for the configuration of the I/O circuit of each type.

Table 2-2. Pin I/O Circuit Types (78K0/FY2-L)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000/INTP0	5-AQ	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/TO00/TI010			
ANI0/P20	11-G		<Digital input setting> Independently connect to AV_{REF} or V_{SS} via a resistor. <Analog output setting and digital input setting> Leave open.[Note 3]
ANI1/P21			
ANI2/P22			
ANI3/P23/CMP2+			
P30/TOH1/TI51/INTP1	5-AQ		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P60/SCLA0/TxD6	5-AS		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P61/SDAA0/RxD6			
P121/X1/TOOLC0[Note 1]	37-A	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK/TOOLD0[Notes 1, 2]			
$\overline{\text{RESET}}$	2		Connect directly to V_{DD} or via a resistor.
AV_{REF}	–	–	Connect directly to V_{DD}.

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

2. If operating in the standalone mode after writing to a load module file (extension: *.lnk or *.lmf) that has debugging information, pull up TOOLD0. Note that operation is not guaranteed in this environment.

3. If this pin is left open when specified as an analog input pin, the input voltage level might become undefined. It is therefore recommended to leave this pin open after specifying it as a digital output pin.

Caution ANI0/P20, ANI1/P21, ANI2/P22, and ANI3/P23/CMP2+ are set in the analog input mode after release of reset.

Table 2-3. Pin I/O Circuit Types (78K0/FA2-L)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000/INTP0	5-AQ	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/TO00/TI010			
ANI0/P20	11-G		<Digital input setting> Independently connect to AV_{REF} or V_{SS} via a resistor. <Analog output setting and digital input setting> Leave open.[Note 3]
ANI1/P21			
ANI2/P22			
ANI3/P23/CMP2+			
ANI4/P24/CMP0+			
ANI5/P25/CMP1+			
P30/TOH1/TI51/INTP1	5-AQ		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P31/INTP2/TOX00/TOOLC1			
P32/TOH1/INTP3/TOX01/ TOOLD1			
P60/SCLA0/TxD6	5-AS		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P61/SDAA0/RxD6			
P121/X1/TOOLC0[Note 1]	37-A	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK/TOOLD0[Notes 1, 2]			
$\overline{\text{RESET}}$	42	Input	Connect directly to V_{DD} or via a resistor.
AV_{REF}	–	–	Connect directly to V_{DD}.

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

2. If operating in the standalone mode after writing to a load module file (extension: *.lnk or *.lmf) that has debugging information, pull up TOOLD0. Note that operation is not guaranteed in this environment.

3. If this pin is left open when specified as an analog input pin, the input voltage level might become undefined. It is therefore recommended to leave this pin open after specifying it as a digital output pin.

Caution ANI0/P20, ANI1/P21, ANI2/P22, ANI3/P23/CMP2+, ANI4/P24/CMP0+, and ANI5/P25/CMP1+ are set in the analog input mode after release of reset.

Table 2-4. Pin I/O Circuit Types (78K0/FB2-L)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000/INTP0	5-AQ	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/TO00/TI010			
P02/$\overline{SSI11}$/INTP5			
ANI0/P20	11-G		<Digital input setting> Independently connect to AV_{REF} or AV_{SS} via a resistor. <Analog output setting and digital input setting> Leave open.[Note 3]
ANI1/P21			
ANI2/P22			
ANI3/P23/CMP2+			
ANI4/P24/CMP0+			
ANI5/P25/CMP1+			
ANI6/P26/CMPCOM			
ANI7/P27			
P30/TOH1/TI51/INTP1	5-AQ		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P31/INTP2/TOX00/TOOLC1			
P32/INTP3/TOX01/TOOLD1			
P33/TOX10			
P34/TOX11/INTP4			
P35/$\overline{SCK11}$			
P36/SI11			
P37/SO11	5-AG		
P60/SCLA0/TxD6	5-AS		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P61/SDAA0/RxD6			
ANI8/P70	11-G		<Digital input setting> Independently connect to AV_{REF} or AV_{SS} via a resistor. <Analog input setting and digital output setting> Leave open.
P121/X1/TOOLC0[Note 1]/<TI000>/<INTP0>	37-A	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK/TOOLD0[Notes 1, 2]			
$\overline{RESET}$	2		Connect directly to V_{DD} or via a resistor.
AV_{REF}	–	–	Connect directly to V_{DD}.[Note 2]
AV_{SS}	–	–	Connect directly to V_{SS}.

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

2. If operating in the standalone mode after writing to a load module file (extension: *.lnk or *.lmf) that has debugging information, pull up TOOLD0. Note that operation is not guaranteed in this environment.

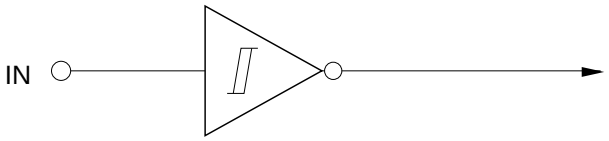
3. If this pin is left open when specified as an analog input pin, the input voltage level might become undefined. It is therefore recommended to leave this pin open after specifying it as a digital output pin.

Caution ANI0/P20, ANI1/P21, ANI2/P22, ANI3/P23/CMP2+, ANI4/P24/CMP0+, ANI5/P25/CMP1+, ANI6/P26/CMPCOM, ANI7/P27, and ANI8/P70 are set in the analog input mode after release of reset.

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

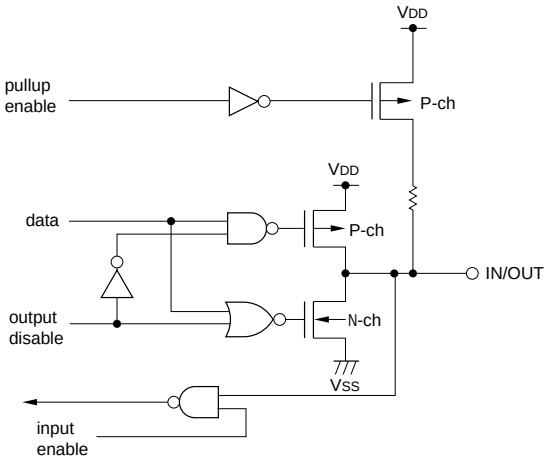
Figure 2-1. Pin I/O Circuit List

Type 2

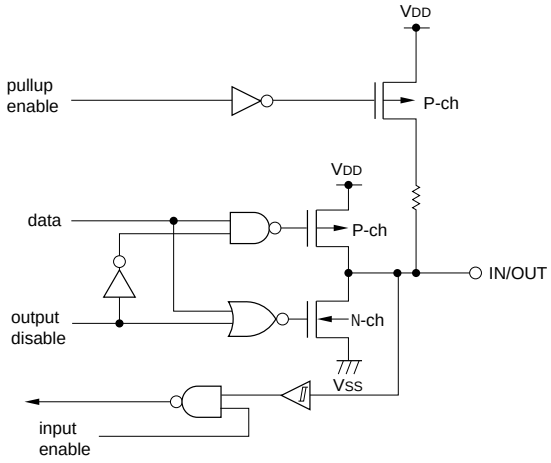


Schmitt-triggered input with hysteresis characteristics

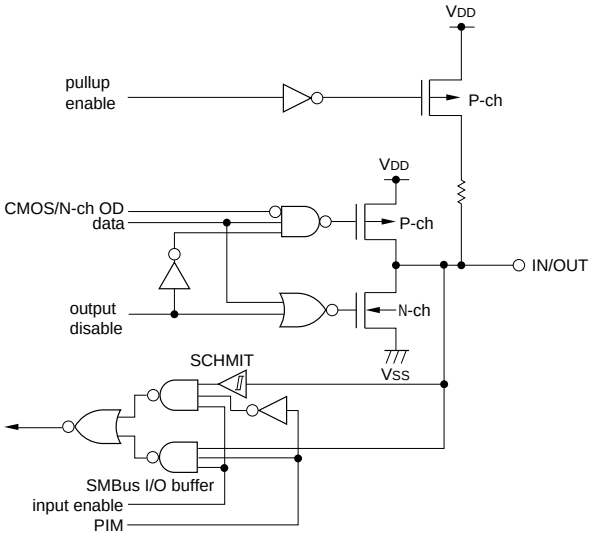
Type 5-AG



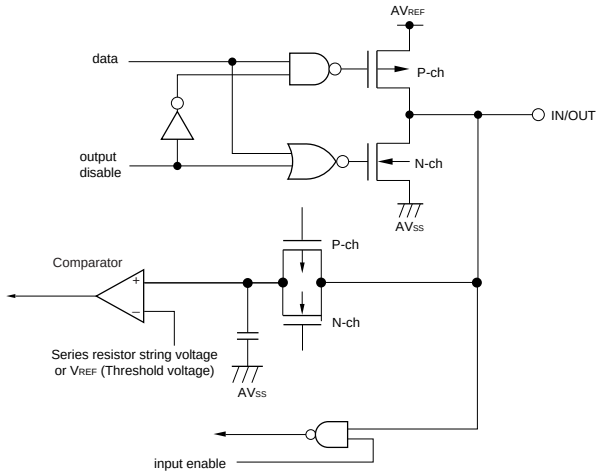
Type 5-AQ



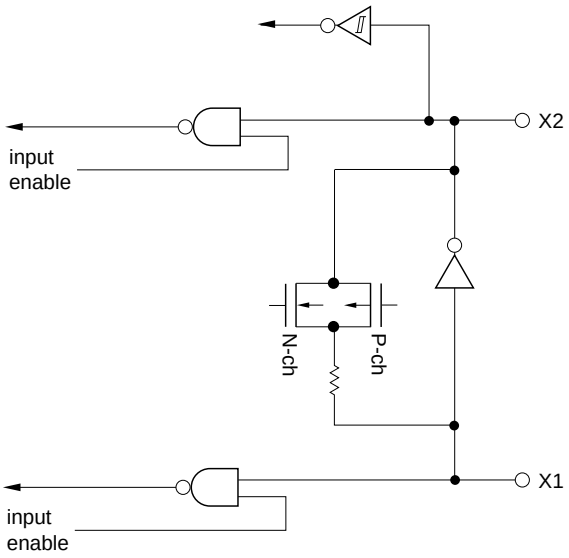
Type 5-AS



Type 11-G



Type 37-A



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

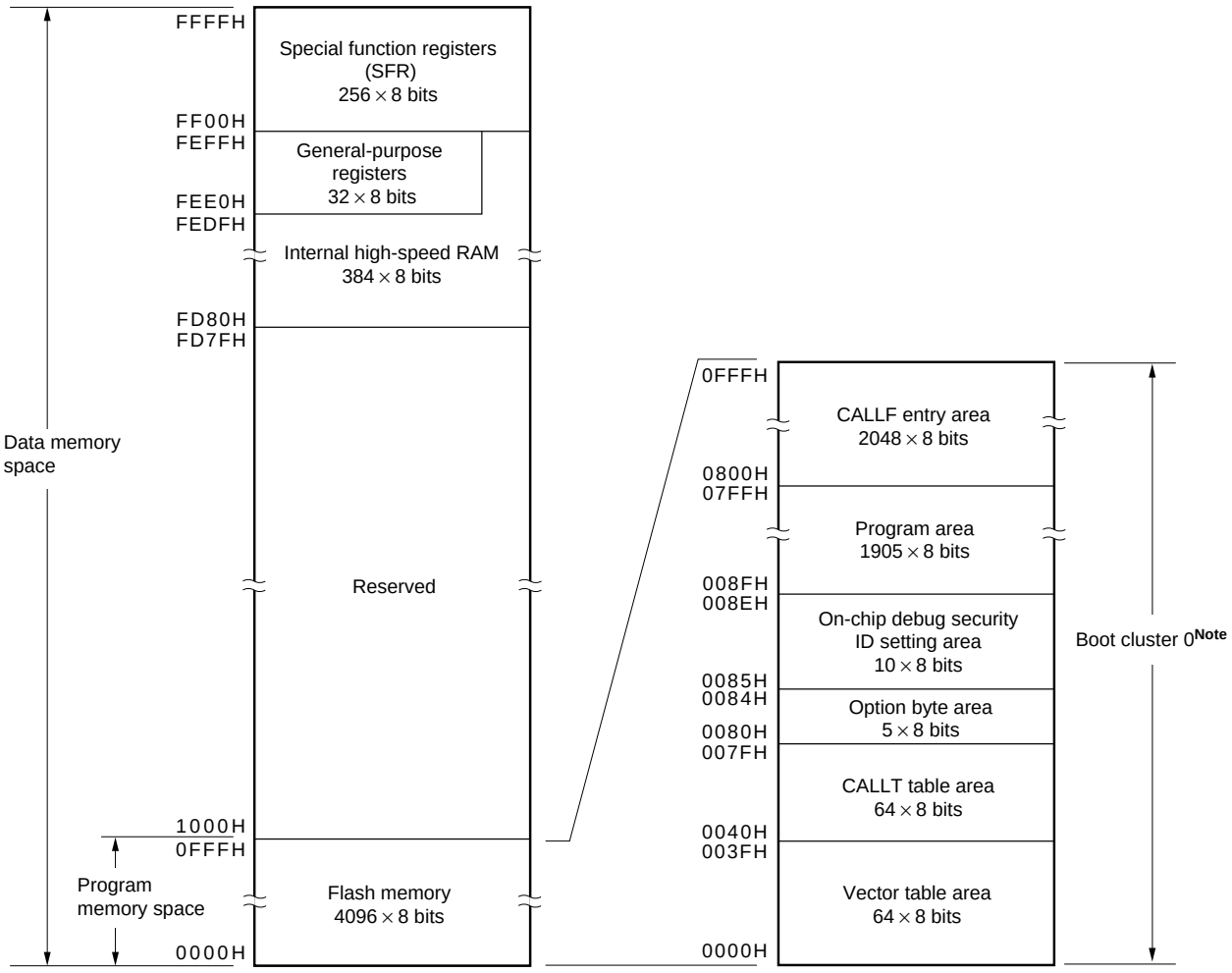
Products in the 78K0/Fx2-L microcontrollers can access a 64 KB memory space. Figures 3-1 to 3-3 show the memory maps.

Caution Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated below after release of reset.

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

Products			IMS	ROM Capacity	Internal High-Speed RAM Capacity
78K0/FY2-L	78K0/FA2-L	78K0/FB2-L			
μPD78F0854	μPD78F0857	–	61H	4 KB	384 bytes
μPD78F0855	μPD78F0858	μPD78F0864	42H	8 KB	512 bytes
μPD78F0856	μPD78F0859	μPD78F0865	04H	16 KB	768 bytes

Figure 3-1. Memory Map (μPD78F0854, 78F0857)



Note Writing boot cluster 0 can be prohibited depending on the setting of security (refer to **24.6 Security Settings**).

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

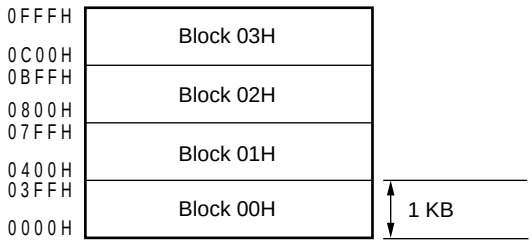
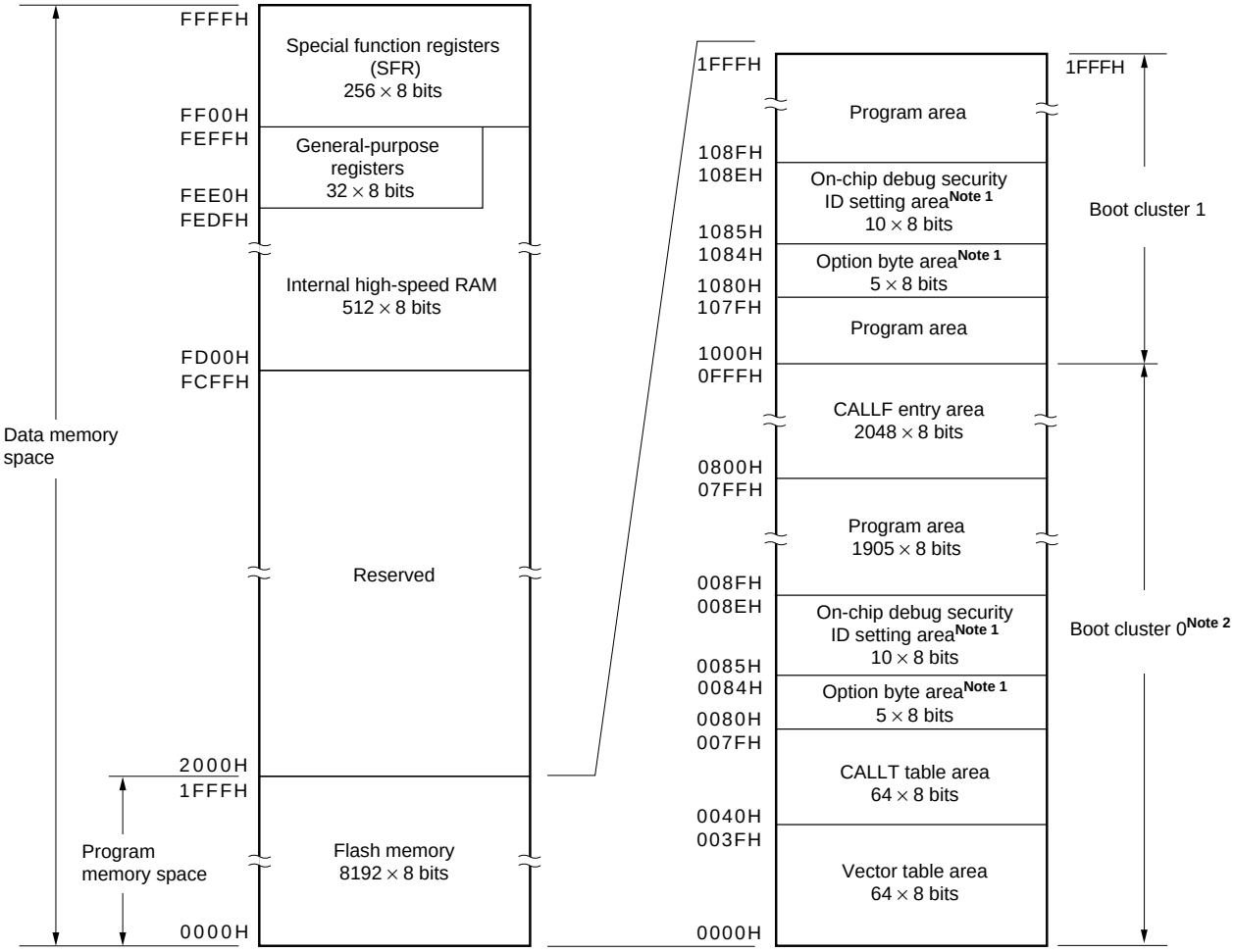


Figure 3-2. Memory Map (μPD78F0855, 78F0858, 78F0864)



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 (refer to **24.6 Security Settings**).

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

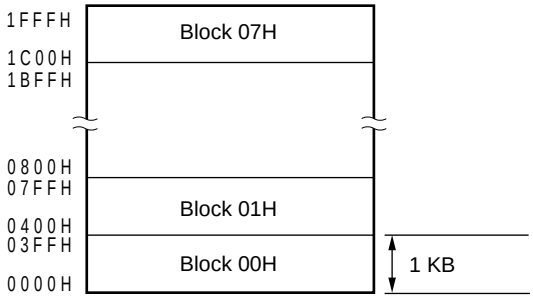
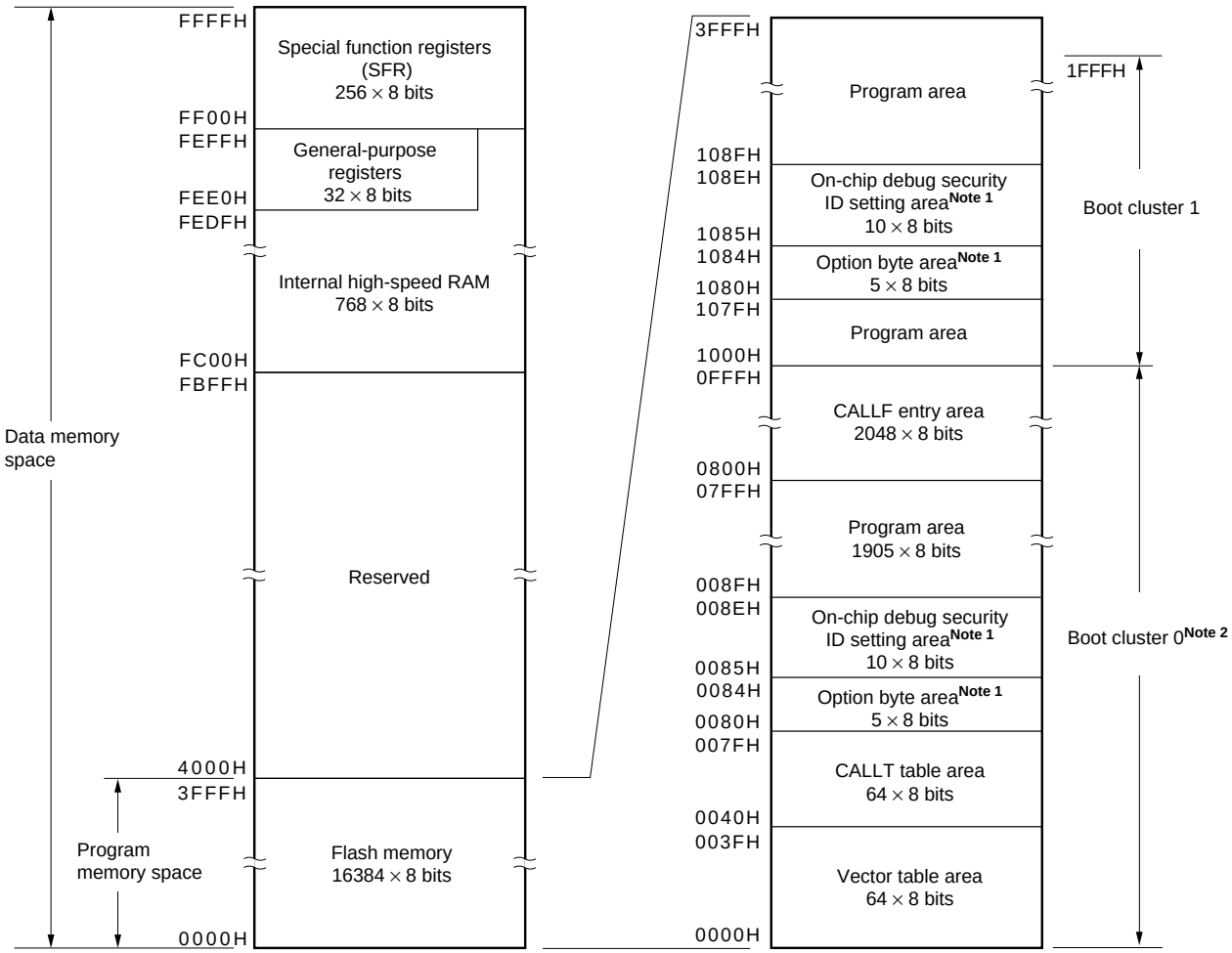


Figure 3-3. Memory Map (μPD78F0856, 78F0859, 78F0865)

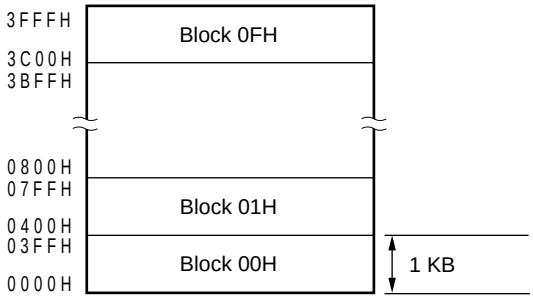


Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

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 (refer to **24.6 Security Settings**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, refer to **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
0000H to 03FFH	00H
0400H to 07FFH	01H
0800H to 0BFFH	02H
0C00H to 0FFFH	03H
1000H to 13FFH	04H
1400H to 17FFH	05H
1800H to 1BFFH	06H
1C00H to 1FFFH	07H
2000H to 23FFH	08H
2400H to 27FFH	09H
2800H to 2BFFH	0AH
2C00H to 2FFFH	0BH
3000H to 33FFH	0CH
3400H to 37FFH	0DH
3800H to 3BFFH	0EH
3C00H to 3FFFH	0FH

Remark μPD78F0854, 78F0857: Block numbers 00H to 03H
μPD78F0855, 78F0858, 78F0864: Block numbers 00H to 07H
μPD78F0856, 78F0859, 78F0865: Block numbers 00H to 0FH

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/Fx2-L microcontrollers incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Product			Internal ROM	
78K0/FY2-L	78K0/FA2-L	78K0/FB2-L	Structure	Capacity
μPD78F0854	μPD78F0857	–	Flash memory	4096 × 8 bits (0000H to 0FFFH)
μPD78F0855	μPD78F0858	μPD78F0864		8192 × 8 bits (0000H to 1FFFH)
μPD78F0856	μPD78F0859	μPD78F0865		16384 × 8 bits (0000H to 3FFFH)

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	78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
		16 Pins	20 Pins	30 Pins
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	√	√	√
0004H	INTLVI	√	√	√
0006H	INTP0	√	√	√
0008H	INTP1	√	√	√
000AH	INTP2	–	√	√
000CH	INTP3	–	√	√
000EH	INTP4	–	–	√
0010H	INTP5	–	–	√
0012H	INTSRE6	√	√	√
0014H	INTSR6	√	√	√
0016H	INTST6	√	√	√
0018H	INTCSI11	–	–	√
001AH	INTTMH1	√	√	√
001CH	INTTMX0	√	√	√
001EH	INTTMX1	–	–	√
0020H	INTTM000	√	√	√
0022H	INTTM010	√	√	√
0024H	INTAD	√	√	√
002AH	INTTM51	√	√	√
002CH	INTCMP0	–	√	√
002EH	INTCMP1	–	√	√
0030H	INTCMP2	√	√	√
0034H	INTIICA0	√	√	√
003EH	BRK	√	√	√

Remark √: Mounted, –: Not mounted

(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, refer to **CHAPTER 23 OPTION BYTE**.

(4) 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, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

(5) CALLF instruction entry area

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

3.1.2 Internal data memory space

78K0/Fx2-L microcontrollers incorporate the following RAMs.

(1) Internal high-speed RAM

Table 3-5. Internal High-Speed RAM Capacity

Product			Internal High-Speed RAM
78K0/FY2-L	78K0/FA2-L	78K0/FB2-L	
μPD78F0854	μPD78F0857	–	384 × 8 bits (FD80H to FEFFH)
μPD78F0855	μPD78F0858	μPD78F0864	512 × 8 bits (FD00H to FEFFH)
μPD78F0856	μPD78F0859	μPD78F0865	768 × 8 bits (FC00H to FEFFH)

The 32-byte area FEE0H to FEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per bank.

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.

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Table 3-6 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/Fx2-L microcontrollers, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-4 to 3-6 show correspondence between data memory and addressing. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

Figure 3-4. Correspondence Between Data Memory and Addressing
(μPD78F0854, 78F0857)

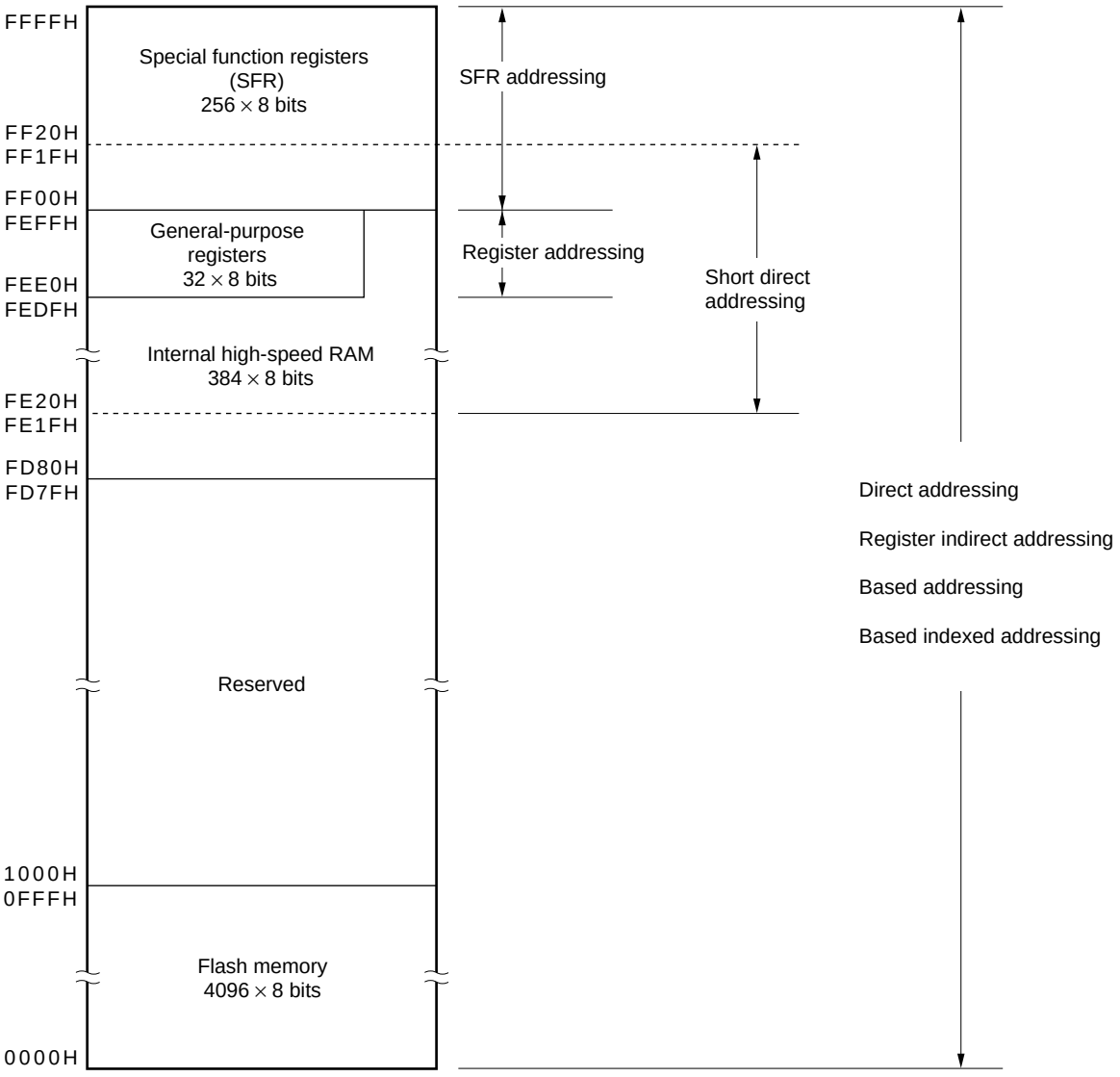


Figure 3-5. Correspondence Between Data Memory and Addressing
(μPD78F0855, 78F0858, 78F0864)

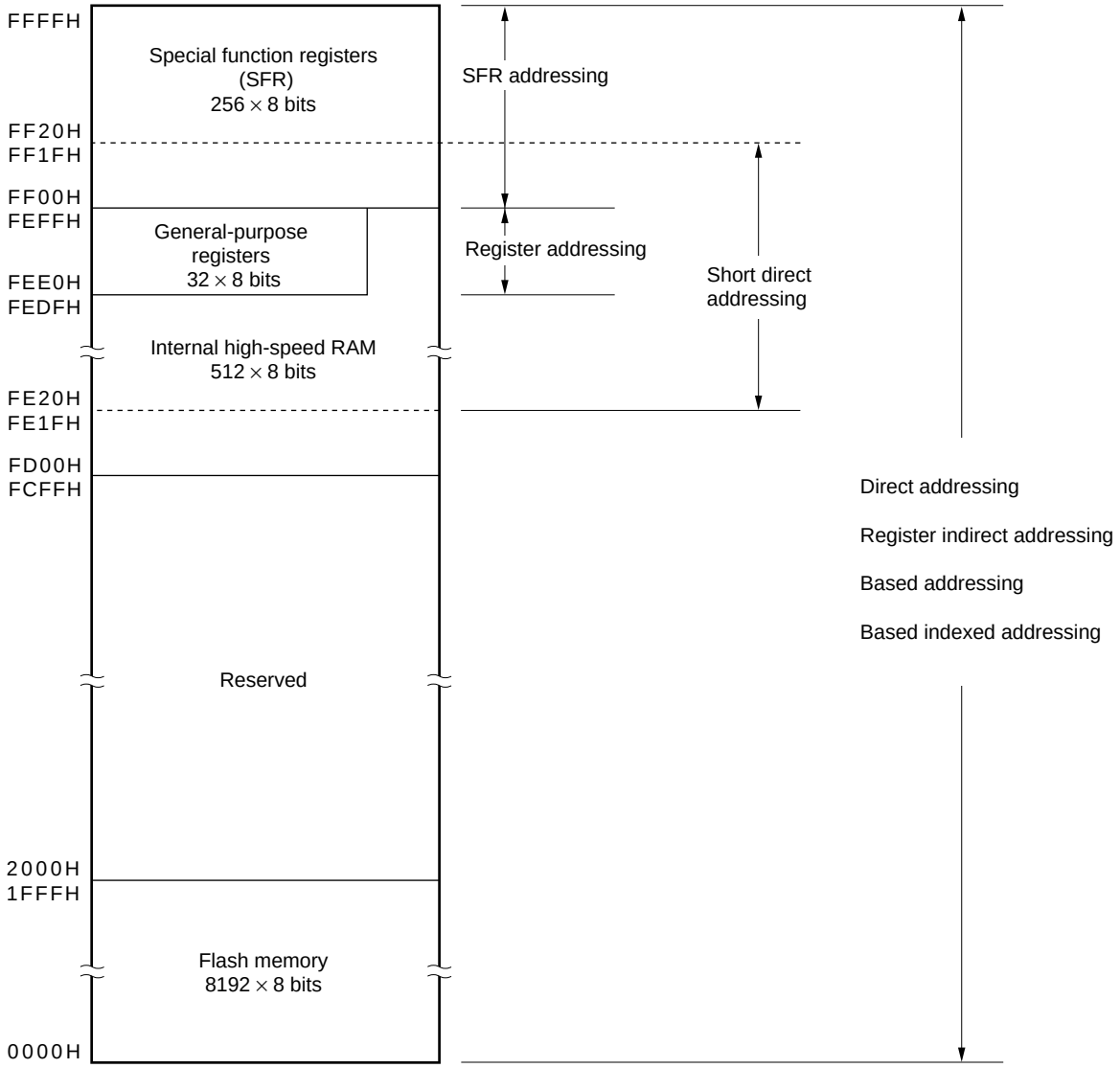
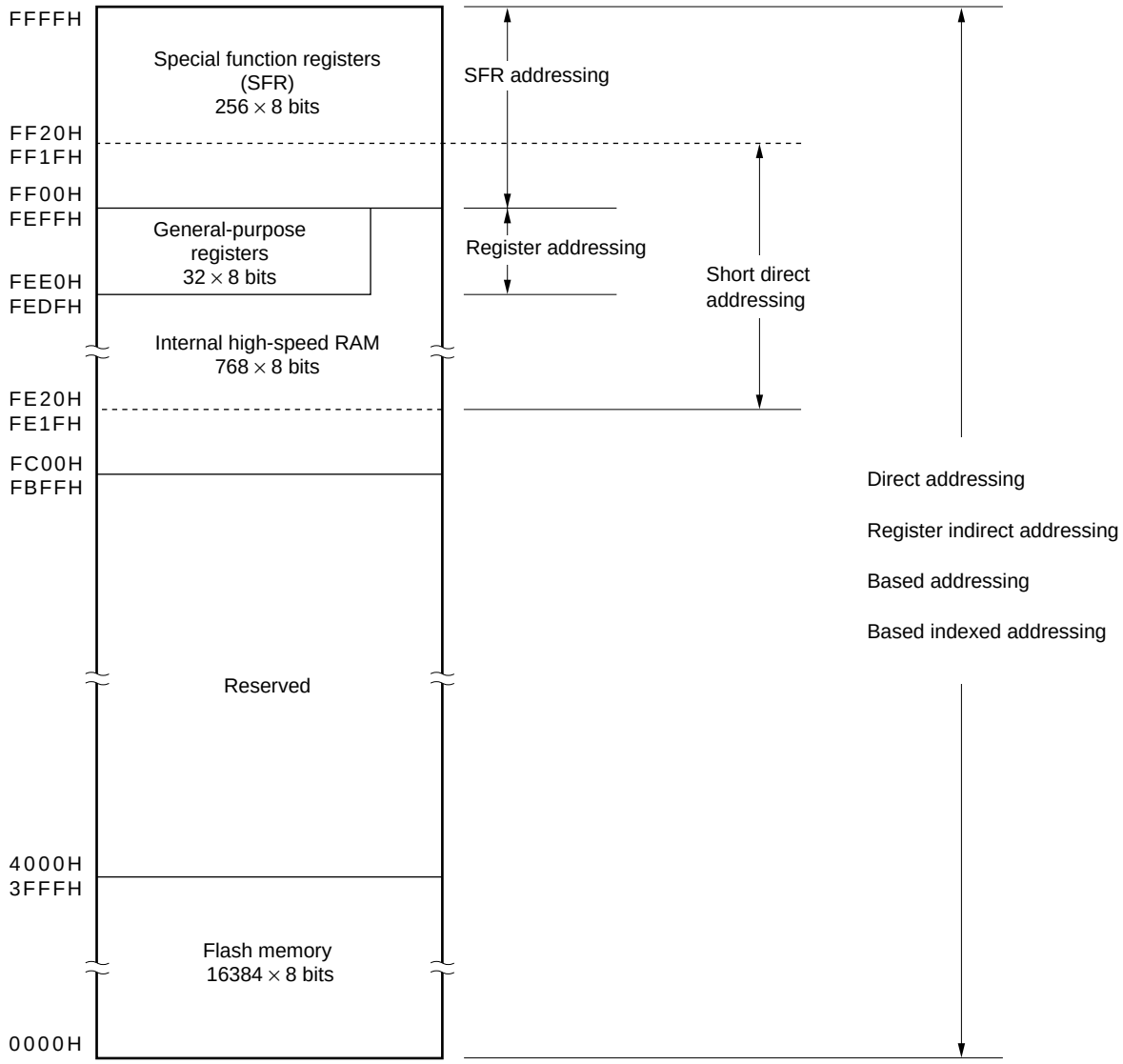


Figure 3-6. Correspondence Between Data Memory and Addressing
(μPD78F0856, 78F0859, 78F0865)



3.2 Processor Registers

The 78K0/Fx2-L microcontrollers incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 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 vectored interrupt request acknowledge 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) (refer to **17.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

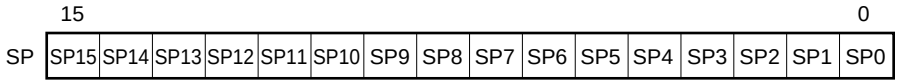
(f) Carry flag (CY)

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



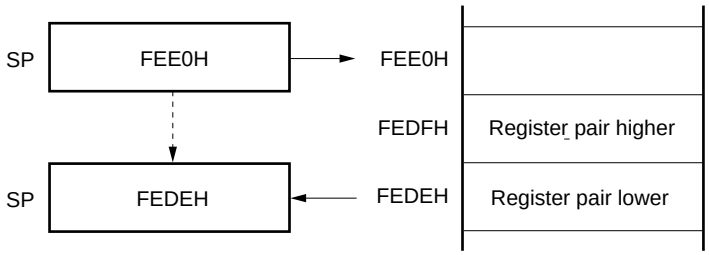
The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

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

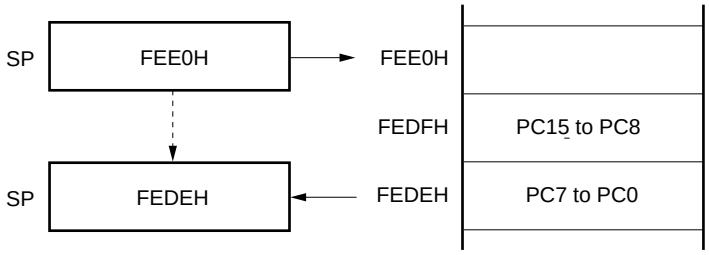
Caution Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

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)

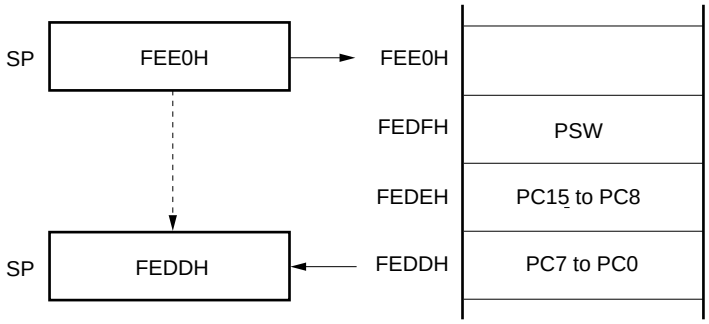
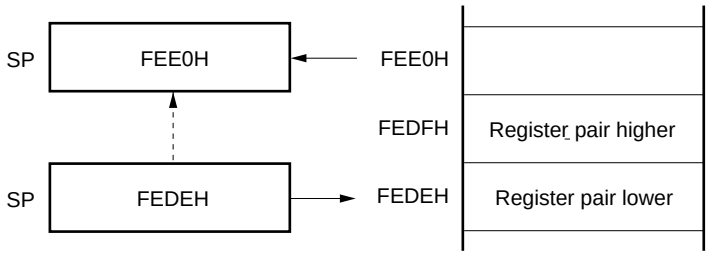
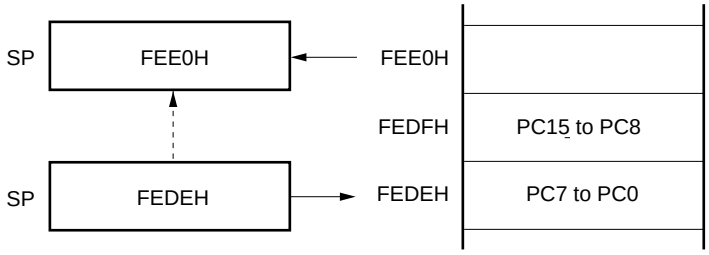


Figure 3-11. Data to Be Restored from Stack Memory

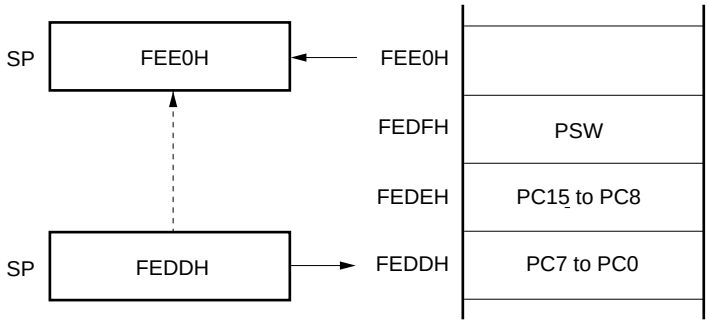
(a) POP rp instruction (when SP = FEDEH)



(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).

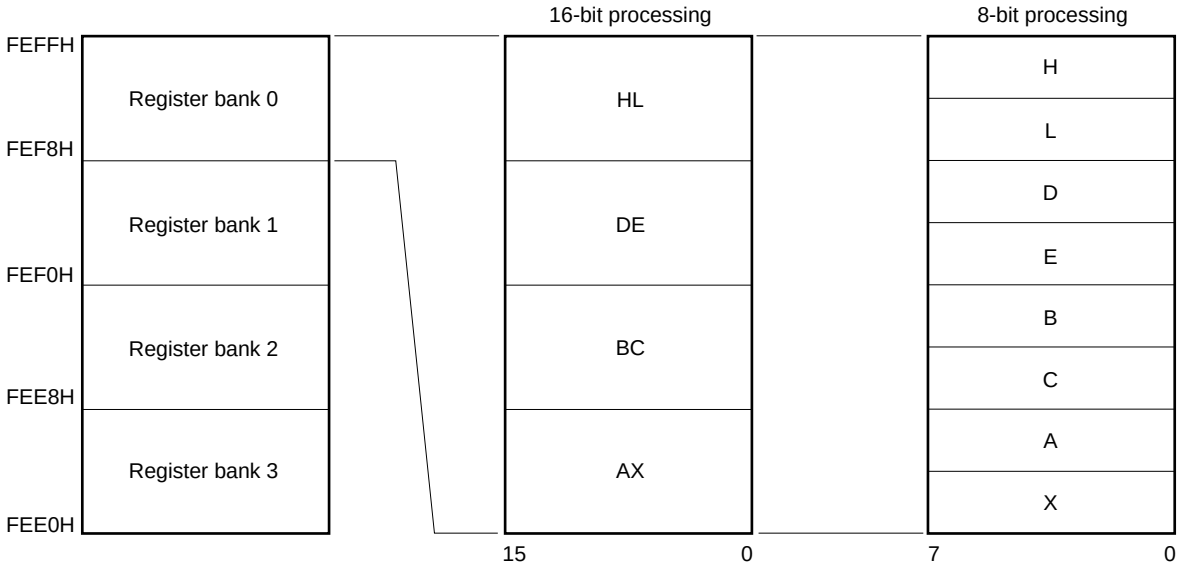
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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).

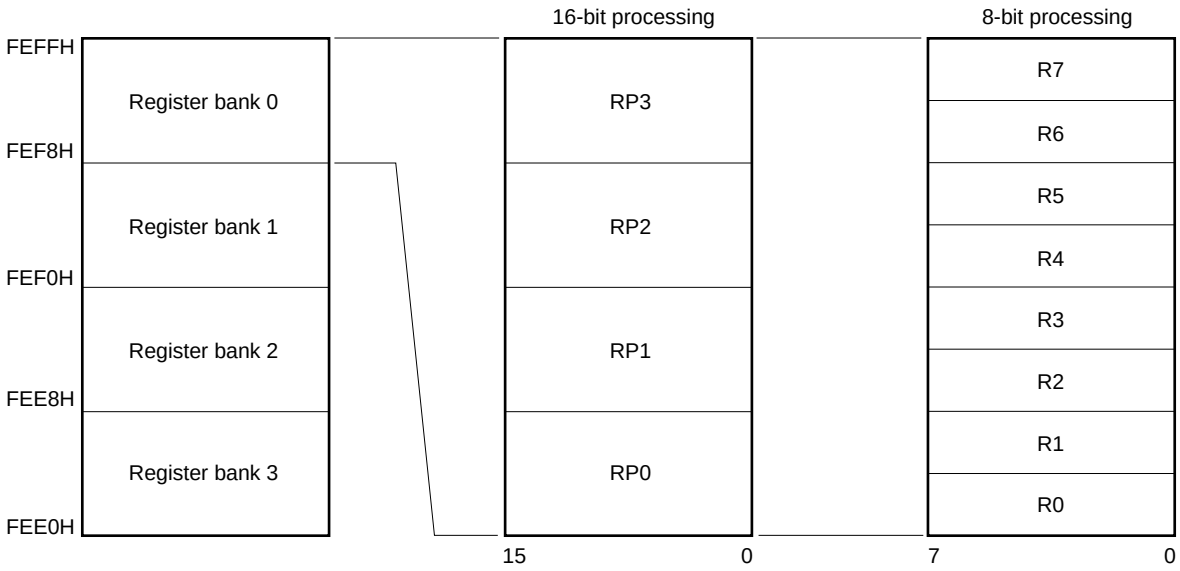
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

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 area.

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 3-6 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 system simulator, 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 3-6. Special Function Register List (1/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	FY 2-L	FA 2-L	FB 2-L
				1 Bit	8 Bits	16 Bits				
FF00H	Port register 0	P0	R/W	√	√	–	00H	√	√	√
FF02H	Port register 2	P2	R/W	√	√	–	00H	√	√	√
FF03H	Port register 3	P3	R/W	√	√	–	00H	√	√	√
FF06H	Port register 6	P6	R/W	√	√	–	00H	√	√	√
FF07H	Port register 7	P7	R/W	√	√	–	00H	–	–	√
FF08H	10-bit 8-bit A/D conversion result register L	ADCRL	R	–	√	–	00H	√	√	√
FF09H	A/D conversion result register	ADCR	R	–	–	√	0000H	√	√	√
FF0AH	Receive buffer register 6	RXB6	R	–	√	–	FFH	√	√	√
FF0BH	Transmit buffer register 6	TXB6	R/W	–	√	–	FFH	√	√	√
FF0CH	Port register 12	P12	R/W	√	√	–	00H	√	√	√
FF0DH	8-bit A/D conversion result register	ADCRH	R	–	√	–	00H	√	√	√
FF0EH	Analog input channel specification register	ADS	R/W	√	√	–	00H	√	√	√
FF0FH	Serial I/O shift register 11	SIO11	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	10-bit A/D conversion result register for TMX0 synchronization: 8-bit A/D conversion result register L for TMX0 synchronization	ADCRX0L	R	–	√	–	00H	√	√	√
FF17H		ADCRX0	R	–	–	√	0000H	√	√	√
FF18H	10-bit A/D conversion result register for TMX1 synchronization: 8-bit A/D conversion result register L for TMX1 synchronization	ADCRX1L	R	–	√	–	00H	–	–	√
FF19H		ADCRX1	R	–	–	√	0000H	–	–	√
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	√	√	√
FF22H	Port mode register 2	PM2	R/W	√	√	–	FFH	√	√	√
FF23H	Port mode register 3	PM3	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 0	ADM0	R/W	√	√	–	00H	√	√	√
FF2AH	Port output mode register 6	POM6	R/W	√	√	–	00H	√	√	√
FF2BH	Self programming mode select register	FPCTL	R/W	√	√	–	00H	√	√	√

Table 3-6. Special Function Register List (2/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	FY 2-L	FA 2-L	FB 2-L
				1 Bit	8 Bits	16 Bits				
FF2EH	A/D port configuration register 0	ADPC0	R/W	√	√	–	00H	√	√	√
FF2FH	A/D port configuration register 1	ADPC1	R/W	√	√	–	00H	–	–	√
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H	√	√	√
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H	√	√	√
FF36H	Pull-up resistor option register 6	PU6	R/W	√	√	–	00H	√	√	√
FF39H	Port alternate switch control register	MUXSEL	R/W	√	√	–	00H	–	–	√
FF3DH	Regulator mode control register	RMC	R/W	–	√	–	00H	√	√	√
FF3EH	Port input mode register 6	PIM6	R/W	√	√	–	00H	√	√	√
FF41H	8-bit timer compare register 51	CR51	R/W	–	√	–	00H	√	√	√
FF43H	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H	√	√	√
FF48H	External interrupt rising edge enable register 0	EGPCTL0	R/W	√	√	–	00H	√	√	√
FF49H	External interrupt falling edge enable register 0	EGNCTL0	R/W	√	√	–	00H	√	√	√
FF4AH	External interrupt rising edge enable register 1	EGPCTL1	R/W	√	√	–	00H	√	√	√
FF4BH	External interrupt falling edge enable register 1	EGNCTL1	R/W	√	√	–	00H	√	√	√
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H	√	√	√
FF50H	Asynchronous serial interface operation mode register 6	ASIM6	R/W	√	√	–	01H	√	√	√
FF53H	Asynchronous serial interface reception error status register 6	ASIS6	R	–	√	–	00H	√	√	√
FF55H	Asynchronous serial interface transmission status register 6	ASIF6	R	–	√	–	00H	√	√	√
FF56H	Clock selection register 6	CKSR6	R/W	–	√	–	00H	√	√	√
FF57H	Baud rate generator control register 6	BRGC6	R/W	–	√	–	FFH	√	√	√
FF58H	Asynchronous serial interface control register 6	ASICL6	R/W	√	√	–	16H	√	√	√
FF62H	Comparator 0 control register	C0CTL	R/W	√	√	–	00H	–	√	√
FF63H	Comparator 0 internal reference voltage setting register	C0RVM	R/W	√	√	–	00H	√	√	√
FF64H	Comparator 1 control register	C1CTL	R/W	√	√	–	00H	–	√	√
FF65H	Comparator 1 internal reference voltage setting register	C1RVM	R/W	√	√	–	00H	√	√	√
FF66H	Comparator 2 control register	C2CTL	R/W	√	√	–	00H	√	√	√
FF67H	Comparator 2 internal reference voltage setting register	C2RVM	R/W	√	√	–	00H	√	√	√
FF69H	Comparator output flag register	CMPFLG	R	√	√	–	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	High-impedance output function enable register	HIZTREN	R/W	√	√	–	00H	–	√	√
FF6FH	High-impedance output mode select register	HIZTRS	R/W	√	√	–	00H	–	√	√

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	FY 2-L	FA 2-L	FB 2-L
					1 Bit	8 Bits	16 Bits				
FF70H	Multiplication input data register A	MULA	MULAL	R/W	–	√	√	00H	√	√	√
FF71H			MULAH	R/W	–	√	√	00H	√	√	√
FF72H	Multiplication input data register B	MULB	MULBH	R/W	–	√	√	00H	√	√	√
FF73H			MULBH	R/W	–	√	√	00H	√	√	√
FF74H	16-bit higher multiplication result storage register	MUL0H		R	–	–	√	0000H	√	√	√
FF75H											
FF76H	16-bit lower multiplication result storage register	MUL0L		R	–	–	√	0000H	√	√	√
FF77H											
FF78H	High-impedance output function control register 0	HZA0CTL0		R/W	√	√	–	00H	–	√	√
FF7CH	Transmit buffer register 11	SOTB11		R/W	–	√	–	00H	–	–	√
FF7EH	16-bit timer X0 operation control register 0	TX0CTL0		R/W	√	√	–	00H	√	√	√
FF7FH	16-bit timer X0 operation control register 1	TX0CTL1		R/W	√	√	–	00H	√	√	√
FF80H	16-bit timer X0 operation control register 2	TX0CTL2		R/W	√	√	–	00H	√	√	√
FF81H	16-bit timer X0 operation control register 3	TX0CTL3		R/W	√	√	–	00H	√	√	√
FF82H	16-bit timer X0 operation control register 4	TX0CTL4		R/W	√	√	–	00H	–	√	√
FF83H	16 bit timer X0 output control register 0	TX0IOC0		R/W	√	√	–	00H	–	√	√
FF84H	16 bit timer X0 compare register 0	TX0CR0		R/W	–	–	√	0000H	√	√	√
FF85H											
FF86H	16 bit timer X0 compare register 1	TX0CR1		R/W	–	–	√	0000H	√	√	√
FF87H											
FF88H	Serial operation mode register 11	CSIM11		R/W	√	√	–	00H	–	–	√
FF89H	Serial clock selection register 11	CSIC11		R/W	√	√	–	00H	–	–	√
FF8AH	16 bit timer X0 compare register 2	TX0CR2		R/W	–	–	√	0000H	√	√	√
FF8BC											
FF8CH	Timer clock selection register 51	TCL51		R/W	√	√	–	00H	√	√	√
FF90H	16 bit timer X0 compare register 3	TX0CR3		R/W	–	–	√	0000H	√	√	√
FF91H											
FF92H	16 bit timer X0 capture/compare register 0	TX0CCR0		R/W	–	–	√	0000H	√	√	√
FF94H	16-bit timer X1 operation control register 0	TX1CTL0		R/W	√	√	–	00H	–	–	√
FF95H	16-bit timer X1 operation control register 1	TX1CTL1		R/W	√	√	–	00H	–	–	√
FF96H	16-bit timer X1 operation control register 2	TX1CTL2		R/W	√	√	–	00H	–	–	√

Table 3-6. Special Function Register List (4/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	FY 2-L	FA 2-L	FB 2-L
				1 Bit	8 Bits	16 Bits				
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/ 9AH[Note 1]	√	√	√
FF9AH	16-bit timer X1 operation control register 4	TX1CTL4	R/W	√	√	–	00H	–	–	√
FF9BH	16-bit timer X1 output control register 0	TX1IOC0	R/W	√	√	–	00H	–	–	√
FF9CH	16-bit timer X1 compare register 0	TX1CR0	R/W	–	–	√	0000H	–	–	√
FF9DH										
FF9FH	Clock operation mode select register	OSCCTL	R/W	√	√	–	00H	√	√	√
FFA0H	Internal oscillation mode register	RCM	R/W	√	√	–	80H[Note 2]	√	√	√
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	IICA shift register	IICA	R/W	–	√	–	00H	√	√	√
FFA6H	Slave address register 0	SVA0	R/W	–	√	–	00H	√	√	√
FFA7H	IICA control register 0	IICACTL0	R/W	√	√	–	00H	√	√	√
FFA8H	IICA control register 1	IICACTL1	R/W	√	√	–	00H	√	√	√
FFA9H	IICA flag register 0	IICAF0	R/W	√	√	–	00H	√	√	√
FFAAH	IICA status register 0	IICAS0	R	√	√	–	00H	√	√	√
FFACH	Reset control flag register	RESF	R	–	√	–	00H[Note 3]	√	√	√
FFADH	IICA low-level width setting register	IICWL	R/W	–	√	–	FFH	√	√	√
FFAEH	IICA high-level width setting register	IICWH	R/W	–	√	–	FFH	√	√	√
FFB0H	16-bit timer X1 compare register 1	TX1CR1	R/W	–	–	√	0000H	–	–	√
FFB1H										
FFB2H	16-bit timer X1 compare register 2	TX1CR2	R/W	–	–	√	0000H	–	–	√
FFB3H										
FFB4H	16-bit timer X1 compare register 3	TX1CR3	R/W	–	–	√	0000H	–	–	√
FFB5H										
FFB6H	16-bit timer X1 capture/compare register 0	TX1CCR0	R/W	–	–	√	0000H	–	–	√
FFB7H										

Notes 1. The reset value of WDTE is determined by setting of option byte.

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

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	FY 2-L	FA 2-L	FB 2-L
					1 Bit	8 Bits	16 Bits				
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	√	√	√
FFBEH	Low-voltage detection register	LVIM		R/W	√	√	–	00H Note 1	√	√	√
FFBFH	Low-voltage detection level selection register	LVIS		R/W	√	√	–	00H Note 1	√	√	√
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	√	√	√
FFF0H	Internal memory size switching register Note 2	IMS		R/W	–	√	–	CFH	√	√	√
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. Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated below after release of reset.

Products			IMS	ROM Capacity	Internal High-Speed RAM Capacity
78K0/FY2-L	78K0/FA2-L	78K0/FB2-L			
μPD78F0854	μPD78F0857	–	61H	4 KB	384 bytes
μPD78F0855	μPD78F0858	μPD78F0864	42H	8 KB	512 bytes
μPD78F0856	μPD78F0859	μPD78F0865	04H	16 KB	768 bytes

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

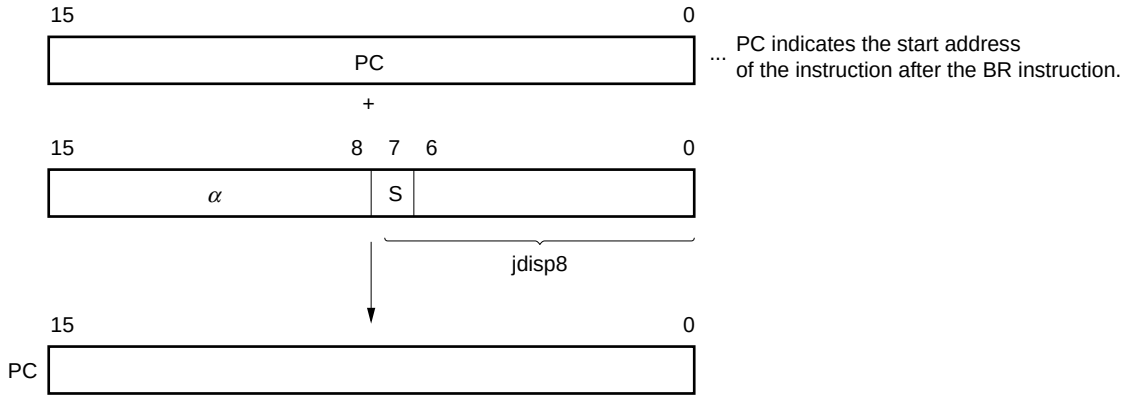
[Function]

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.

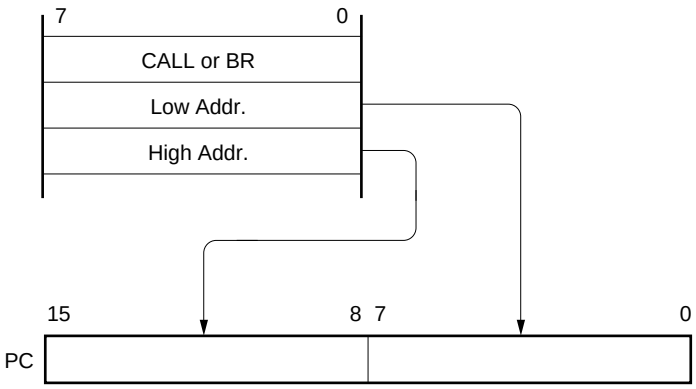
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space.

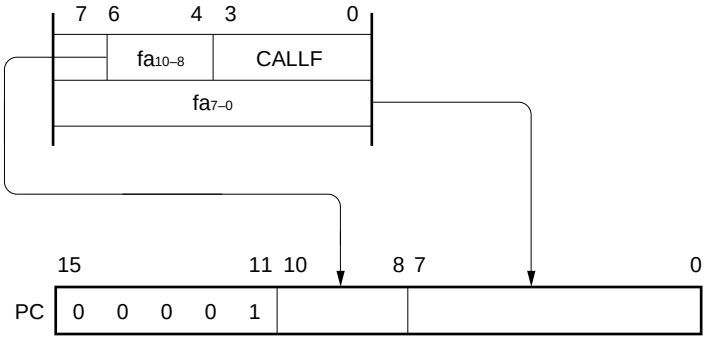
The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

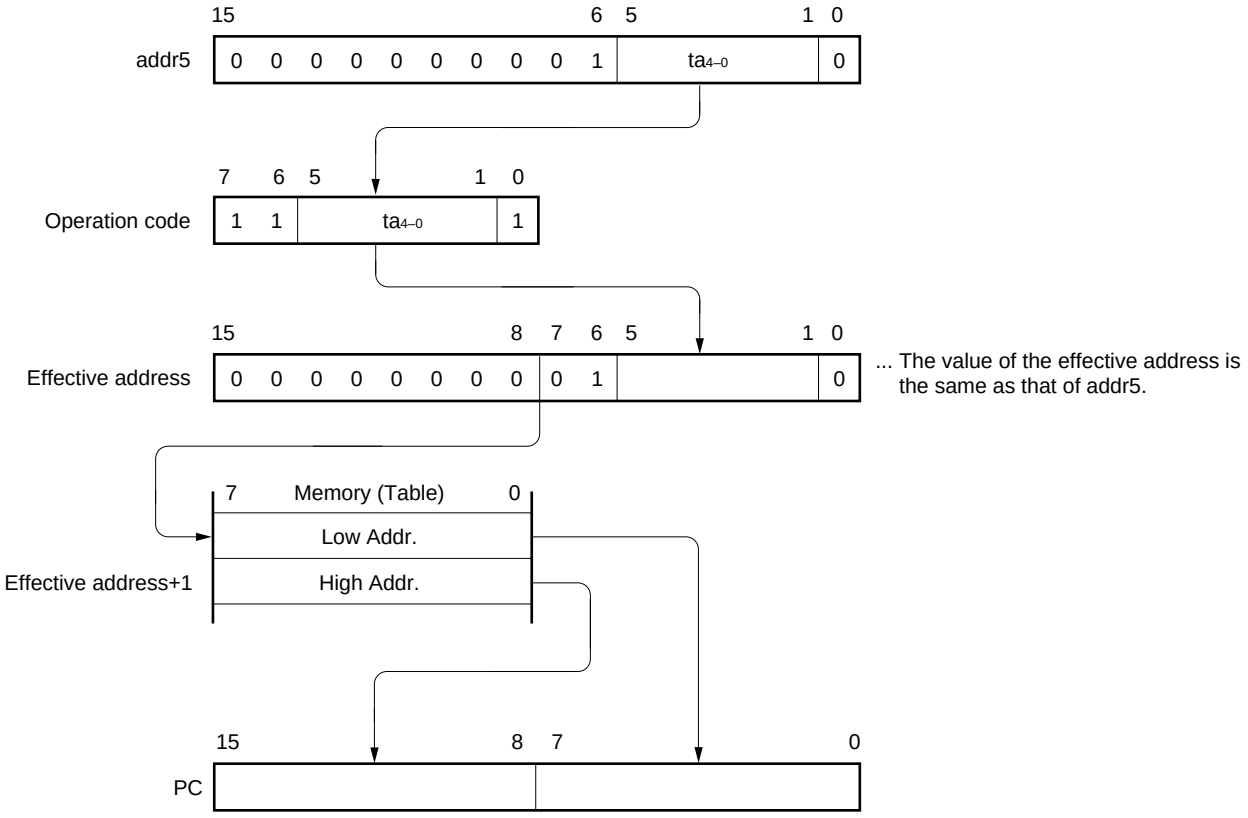
[Function]

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 that is indicated by addr5 and is stored in the memory table from 0040H to 007FH, and allows branching to the entire memory space.

[Illustration]



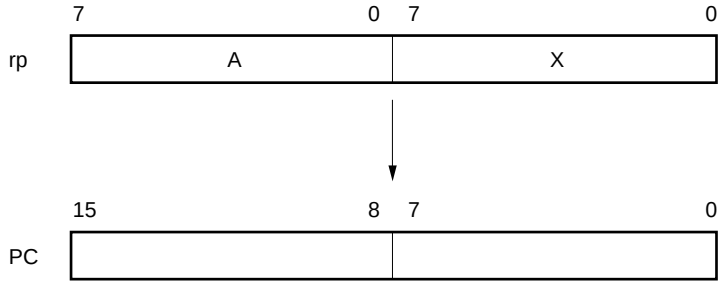
3.3.4 Register addressing

[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/Fx2-L microcontroller 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.

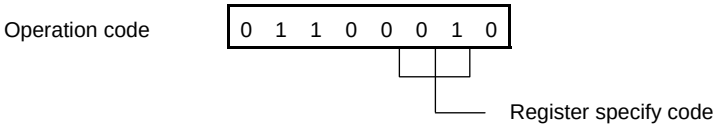
[Operand format]

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

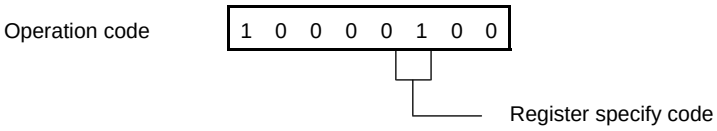
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[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.

This addressing can be carried out for all of the memory spaces.

[Operand format]

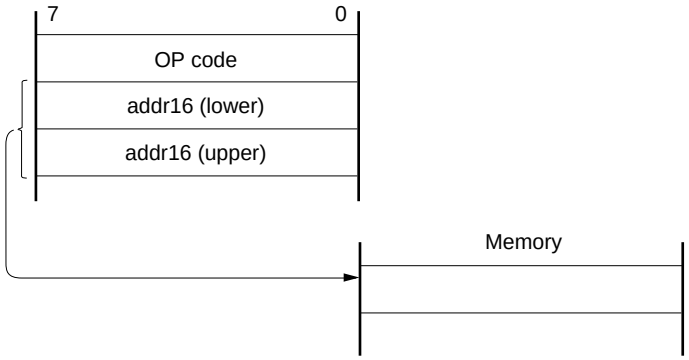
Identifier	Description
addr16	Label or 16-bit immediate data

[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. Refer to the **[Illustration]** shown below.

[Operand format]

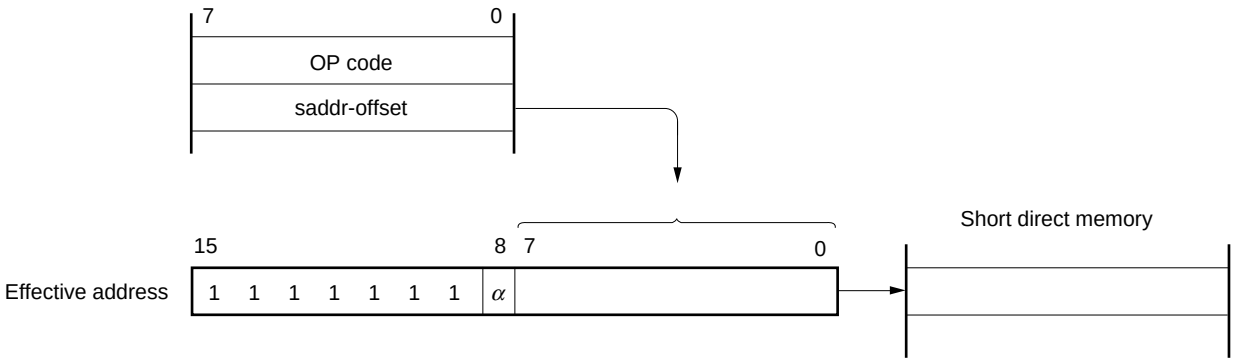
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

```
LB1 EQU 0FE30H   ; Defines FE30H by LB1.
      :
MOV LB1, A       ; When LB1 indicates FE30H of the saddr area and the value of register A is transferred to that
                   address
```

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]

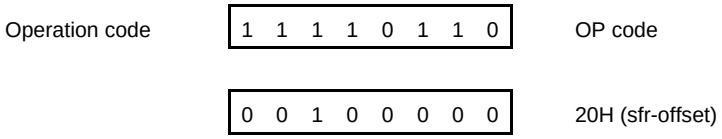
A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word. This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

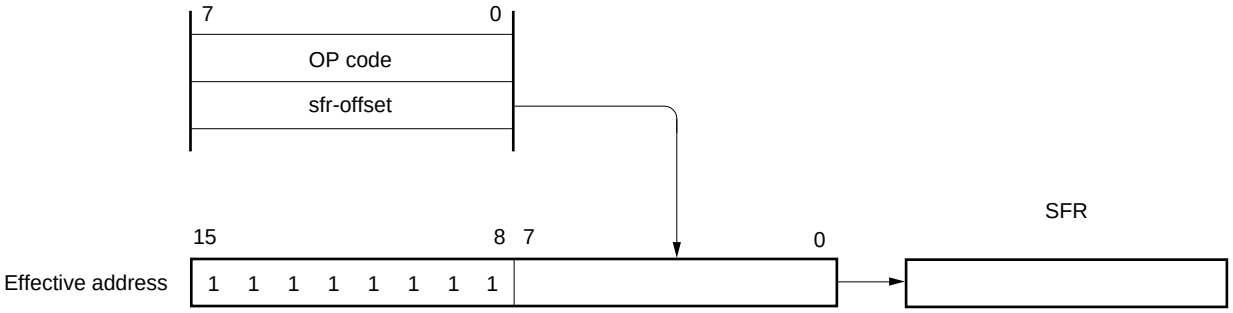
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[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]

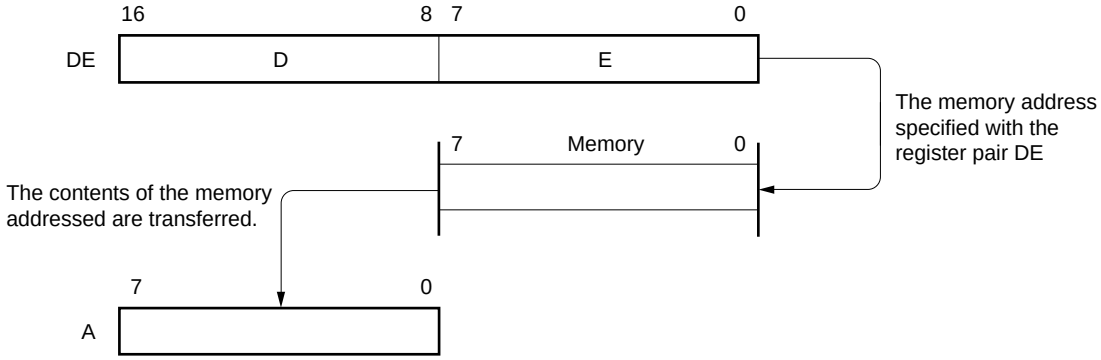
Identifier	Description
–	[DE], [HL]

[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]

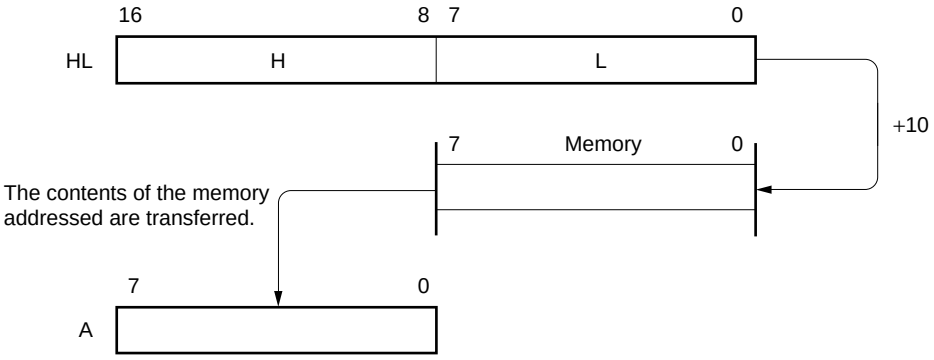
MOV A, [HL + 10H]; when setting byte to 10H

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]

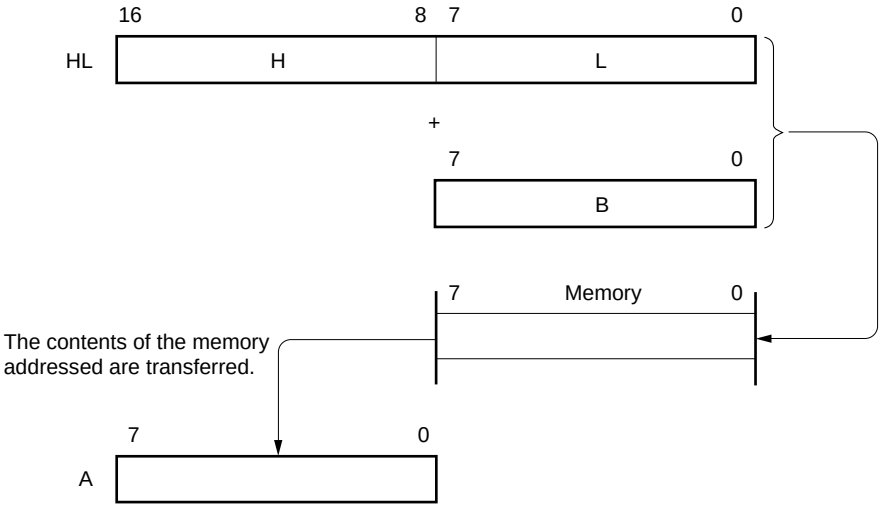
Identifier	Description
–	[HL + B], [HL + C]

[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.

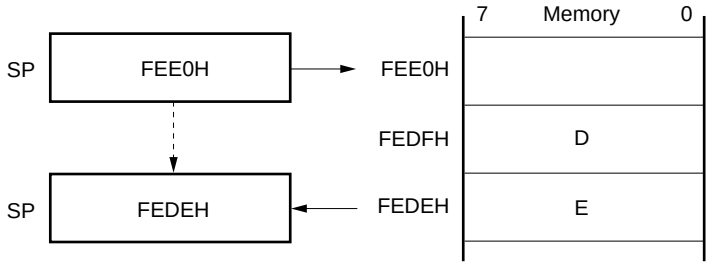
With stack addressing, only the internal high-speed RAM area can be accessed.

[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

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

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

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P70[Note]
V_{DD}	Pins other than P20 to P27, P70[Note]

Note 78K0/FY2-L: P20 to P23
78K0/FA2-L: P20 to P25
78K0/FB2-L: P20 to P27, P70

78K0/Fx2-L microcontrollers are provided with digital I/O ports, which enable variety of control operations. The functions of each port are shown in Tables 4-2 to 4-4.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, refer to **CHAPTER 2 PIN FUNCTIONS**.

Table 4-2. Port Functions (78K0/FY2-L)

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/INTP0
P01				TO00/TI010
P20	I/O	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0
P21				ANI1
P22				ANI2
P23				ANI3/CMP2+
P30	I/O	Port 3. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOH1/TI51/INTP1
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Input can be set to SMBus input buffer in 1-bit units. Output can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA0/TxD6
P61				SDAA0/RxD6
P121	Input	Port 12. 2-bit input-only port.	Input port	X1/TOOLC0
P122				X2/EXCLK/TOOLD0

Table 4-3. Port Functions (78K0/FA2-L)

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/INTP0
P01				TO00/TI010
P20	I/O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0
P21				ANI1
P22				ANI2
P23				ANI3/CMP2+
P24				ANI4/CMP0+
P25				ANI5/CMP1+
P30	I/O	Port 3. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOH1/TI51/INTP1
P31				TOX00/INTP2/TOOLC1
P32				TOX01/INTP3/TOOLD1
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Input can be set to SMBus input buffer in 1-bit units. Output can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA0/TxD6
P61				SDAA0/RxD6
P121	Input	Port 12. 2-bit input-only port.	Input port	X1/TOOLC0
P122				X2/EXCLK/TOOLD0

Table 4-4. Port Functions (78K0/FB2-L)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI000/INTP0
P01				TO00/TI010
P02				$\overline{\text{SSI11}}$/INTP5
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI0
P21				ANI1
P22				ANI2
P23				ANI3/CMP2+
P24				ANI4/CMP0+
P25				ANI5/CMP1+
P26				ANI6/CMPCOM
P27				ANI7
P30	I/O	Port 3. 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	TOH1/TI51/INTP1
P31				TOX00/INTP2/TOOLC1
P32				TOX01/INTP3/TOOLD1
P33				TOX10
P34				TOX11/INTP4
P35				$\overline{\text{SCK11}}$
P36				SI11
P37				SO11
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Input can be set to SMBus input buffer in 1-bit units. Output can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCLA0/TxD6
P61				SDAA0/RxD6
P70	I/O	Port 7. 1-bit I/O port. Input/output can be specified in 1-bit units.	Analog input	ANI8
P121	Input	Port 12. 2-bit input port.	Input port	X1/TOOLC0/ <TI000>/<INTP0>
P122				X2/EXCLK/TOOLD0

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

4.2 Port Configuration

Ports include the following hardware.

Table 4-5. Port Configuration

Item	Configuration
Control registers	• 78K0/FY2-L, 78K0/FA2-L Port mode register (PMxx): PM0, PM2, PM3, PM6 Port register (Pxx): P0, P2, P3, P6, P12 Pull-up resistor option register (PUxx): PU0, PU3, PU6 Port input mode register 6 (PIM6) Port output mode register 6 (POM6) A/D port configuration register 0 (ADPC0) • 78K0/FB2-L Port mode register (PMxx): PM0, PM2, PM3, PM6, PM7 Port register (Pxx): P0, P2, P3, P6, P7, P12 Pull-up resistor option register (PUxx): PU0, PU3, PU6 Port input mode register 6 (PIM6) Port output mode register 6 (POM6) A/D port configuration register 0 (ADPC0) A/D port configuration register 1 (ADPC1)**Note** Port alternate switch control register (MUXSEL)
Port	• 78K0/FY2-L: Total: 11 (CMOS I/O: 9, CMOS input: 2) • 78K0/FA2-L: Total: 15 (CMOS I/O: 13, CMOS input: 2) • 78K0/FB2-L: Total: 24 (CMOS I/O: 22, CMOS input: 2)
Pull-up resistor	• 78K0/FY2-L: Total: 5 • 78K0/FA2-L: Total: 7 • 78K0/FB2-L: Total: 13

Note 78K0/FB2-L only.

4.2.1 Port 0

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
P00/TI000/INTP0	P00/TI000/INTP0	P00/TI000/INTP0
P01/TO00/TI010	P01/TO00/TI010	P01/TO00/TI010
–	–	P02/$\overline{\text{SSI11}}$/INTP5

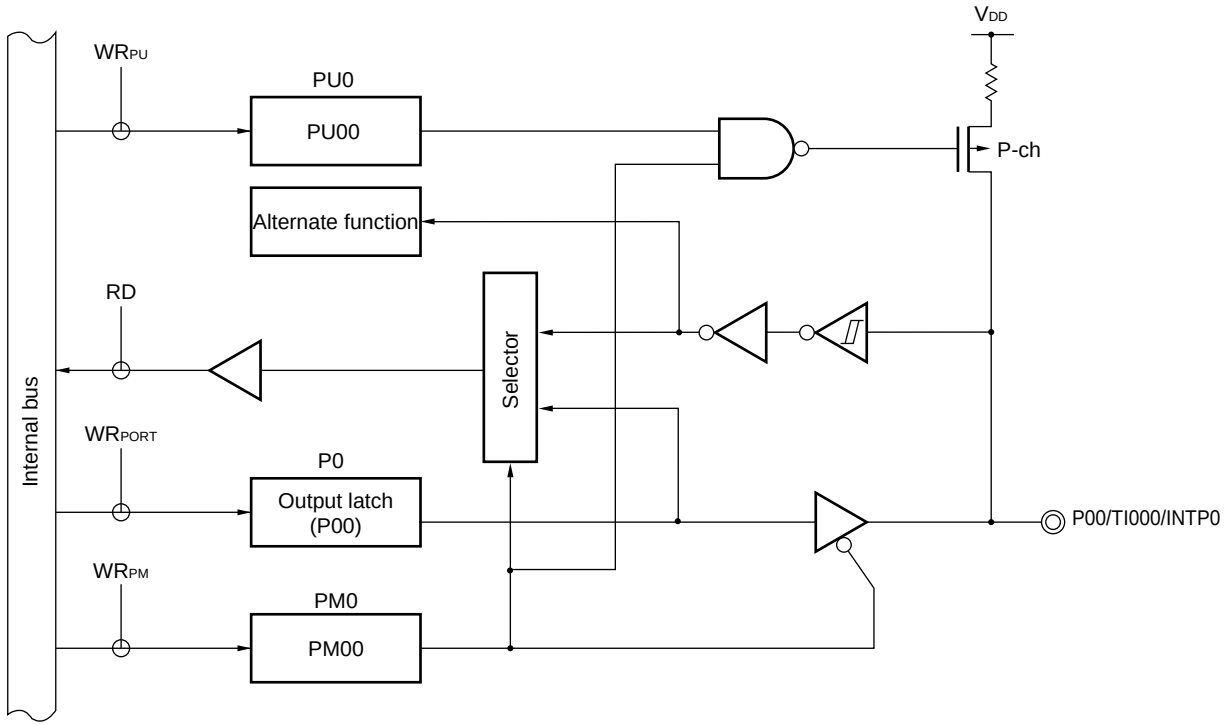
Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P02 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0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

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

Reset signal generation sets port 0 to input mode.

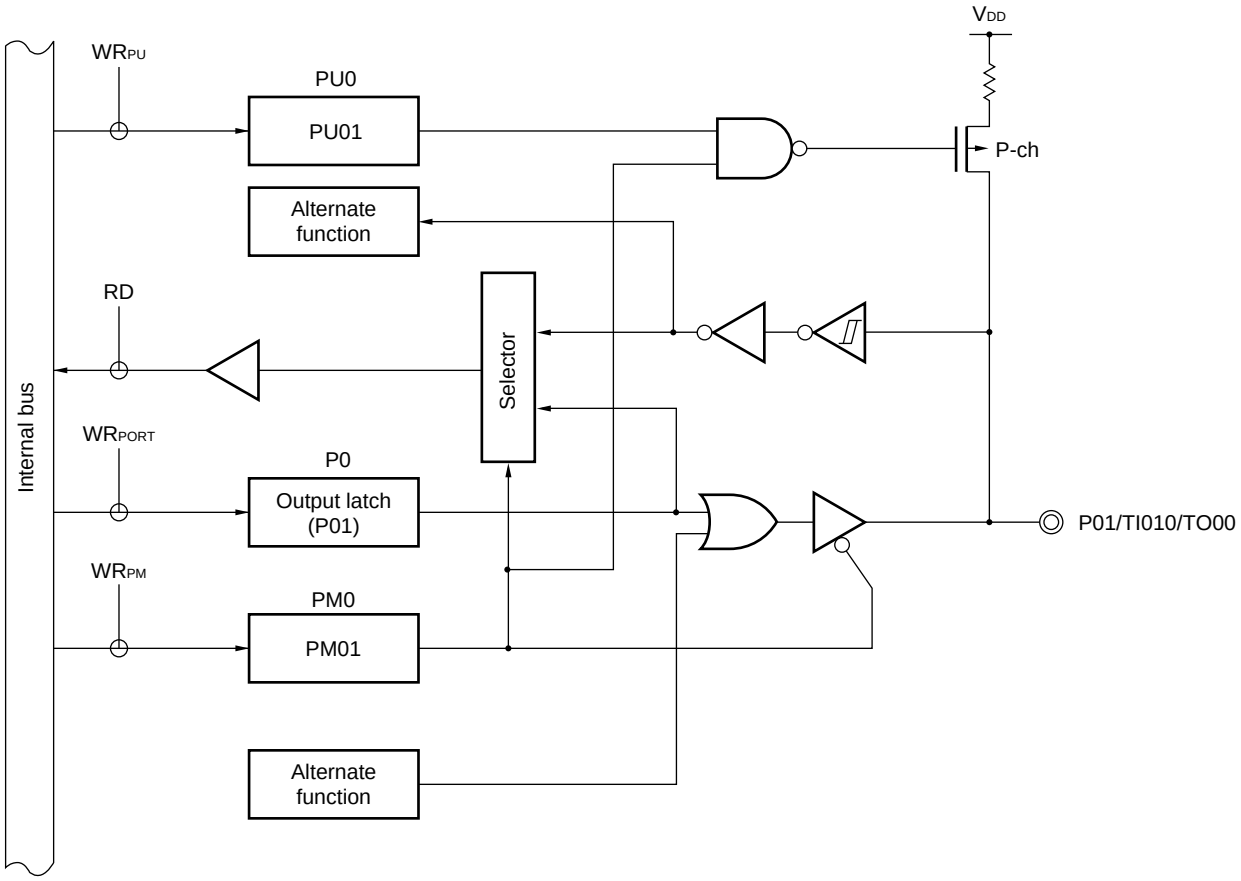
Figures 4-1 to 4-3 show block diagrams of port 0.

Figure 4-1. Block Diagram of P00



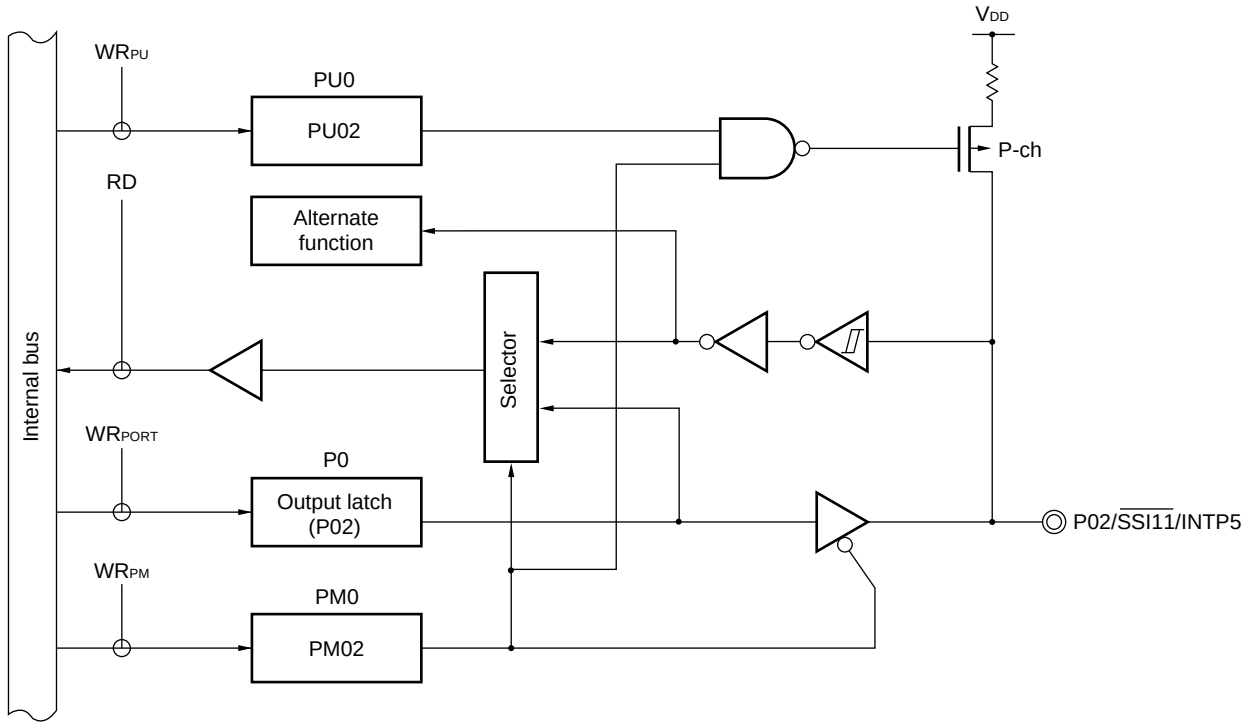
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

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

Figure 4-3. Block Diagram of P02



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

4.2.2 Port 2

78K0/FY2-L (16 Pins)	78K0/FA2-L (20 Pins)	78K0/FB2-L (30 Pins)
P20/ANI0	P20/ANI0	P20/ANI0
P21/ANI1	P21/ANI1	P21/ANI1
P22/ANI2	P22/ANI2	P22/ANI2
P23/ANI3/CMP2+	P23/ANI3/CMP2+	P23/ANI3/CMP2+
–	P24/ANI4/CMP0+	P24/ANI4/CMP0+
–	P25/ANI5/CMP1+	P25/ANI5/CMP1+
–	–	P26/ANI6/CMPCOM
–	–	P27/ANI7

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

This port can also be used for A/D converter analog input, comparator input, and comparator common input.

When using P20/ANI0 to P27/ANI7, set the registers according to the pin function to be used (refer to **Tables 4-6** to **4-10**).

To use P20/ANI0 to P27/ANI7 as a digital input or a digital output, it is recommended to select a pin to use starting with the furthest pin from AV_{REF} (for example, the P24/CMP0+/ANI4 pin in the 78K0/FB2-L). To use P20/ANI0 to P27/ANI7 as an analog input, it is recommended to select a pin to use starting with the closest pin to AV_{SS} (for example, the P27/ANI7 pin in the 78K0/FB2-L).

Table 4-6. Setting Functions of P20/ANI0 and P22/ANI2 Pins

ADPC0 Register	PM2 Register	ADS Register (n = 0, 2)	P20/ANI0 and P22/ANI2 Pins
Digital I/O selection	Input mode	Selects ANIn.	Setting prohibited
		Does not select ANIn.	Digital input
	Output mode	Selects ANIn.	Setting prohibited
		Does not select ANIn.	Digital output
Analog input selection	Input mode	Selects ANIn.	Analog input (to be converted into digital signals)
		Does not select ANIn.	Analog input (not to be converted into digital signals)
	Output mode	–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
ADS: Analog input channel specification register

Table 4-7. Setting Functions of P21/ANI1 Pin

ADPC0 Register	PM2 Register	ADS Register	P21/ANI1 Pin
Digital I/O selection	Input mode	Selects ANI1.	Setting prohibited
		Does not select ANI1.	Digital input
	Output mode	Selects ANI1.	Setting prohibited
		Does not select ANI1.	Digital output
Analog input selection	Input mode	Selects ANI1.	Analog input (to be converted into digital signals)
		Does not select ANI1.	Analog input (not to be converted into digital signals)
	Output mode	–	Setting prohibited

Table 4-8. Setting Functions of P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins

ADPC0 Register	PM2 Register	CMPmEN bit (m = 0 to 2)	ADS Register (n = 3 to 5)	P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins
Digital I/O selection	Input mode	–	Selects ANIn.	Setting prohibited
			Does not select ANIn.	Digital input
	Output mode	–	Selects ANIn.	Setting prohibited
			Does not select ANIn.	Digital output
Analog input selection	Input mode	0	Selects ANIn.	Analog input (to be converted into digital signal)
			Does not select ANIn.	Analog input (not to be converted into digital signal)
		1	Selects ANIn.	Analog input (to be converted into digital signal), Comparator input
			Does not select ANIn.	Comparator input
	Output mode	–	–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
CMPmEN: Bit 7 of comparator m control register (CmCTL)
ADS: Analog input channel specification register

Table 4-9. Setting Functions of P26/ANI6/CMPCOM Pin

ADPC0 Register	PM2 Register	CmMODSEL1 bit (m = 0 to 2)	CmMODSEL0 bit (m = 0 to 2)	ADS Register	P26/ANI6/CMPCOM Pin
Digital I/O selection	Input mode	–		Selects ANI6.	Setting prohibited
				Does not select ANI6.	Digital input
	Output mode	–		Selects ANI6.	Setting prohibited
				Does not select ANI6.	Digital output
Analog input selection	Input mode	CmMODSEL1 = 0, or CmMODSEL0 = 0		Selects ANI6.	Analog input (to be converted into digital signal)
				Does not select ANI6.	Analog input (not to be converted into digital signal)
		CmMODSEL1 = 1, CmMODSEL0 = 1		Selects ANI6.	Analog input (to be converted into digital signal), Comparator common input
				Does not select ANI6.	Comparator common input
	Output mode	–		–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
CmMODSEL1, CmMODSEL0: Bits 4, 3 of comparator m control register (CmCTL)
ADS: Analog input channel specification register

Table 4-10. Setting Functions of P27/ANI7 Pin

ADPC0 Register	PM2 Register	ADS Register	P27/ANI7 Pin
Digital I/O selection	Input mode	Selects ANI7.	Setting prohibited
		Does not select ANI7.	Digital input
	Output mode	Selects ANI7.	Setting prohibited
		Does not select ANI7.	Digital output
Analog input selection	Input mode	Selects ANI7.	Analog input (to be converted into digital signal)
		Does not select ANI7.	Analog input (not to be converted into digital signal)
	Output mode	–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
ADS: Analog input channel specification register

Reset signal generation sets port 2 to analog input.
Figures 4-4 to 4-8 show block diagrams 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 4-4. Block Diagram of P20 to P22 and P27

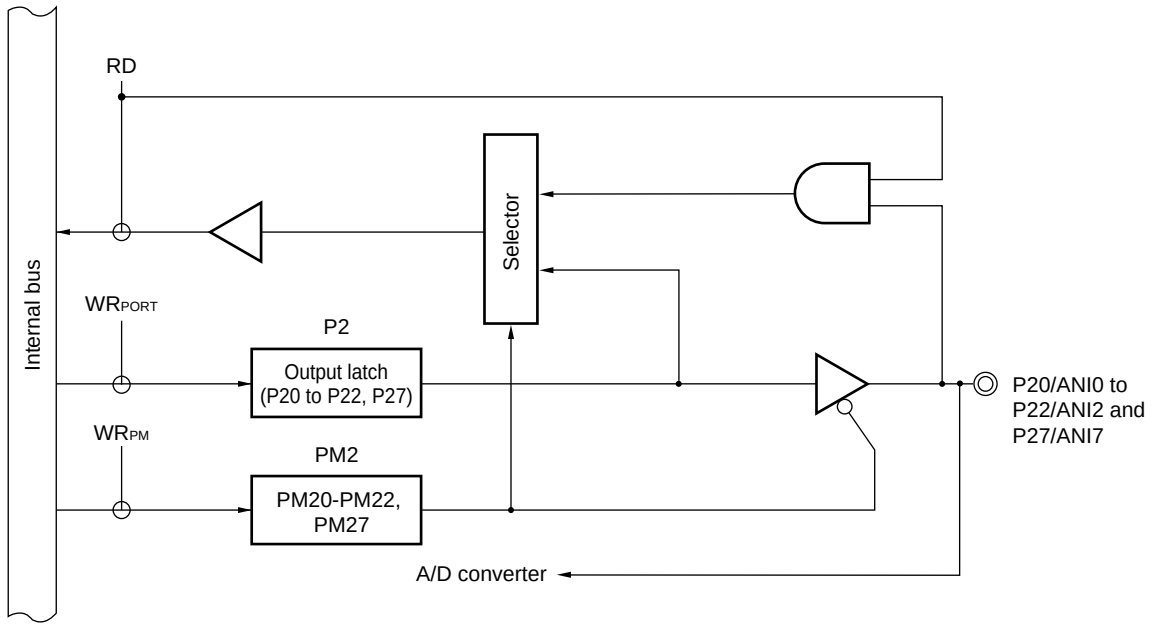
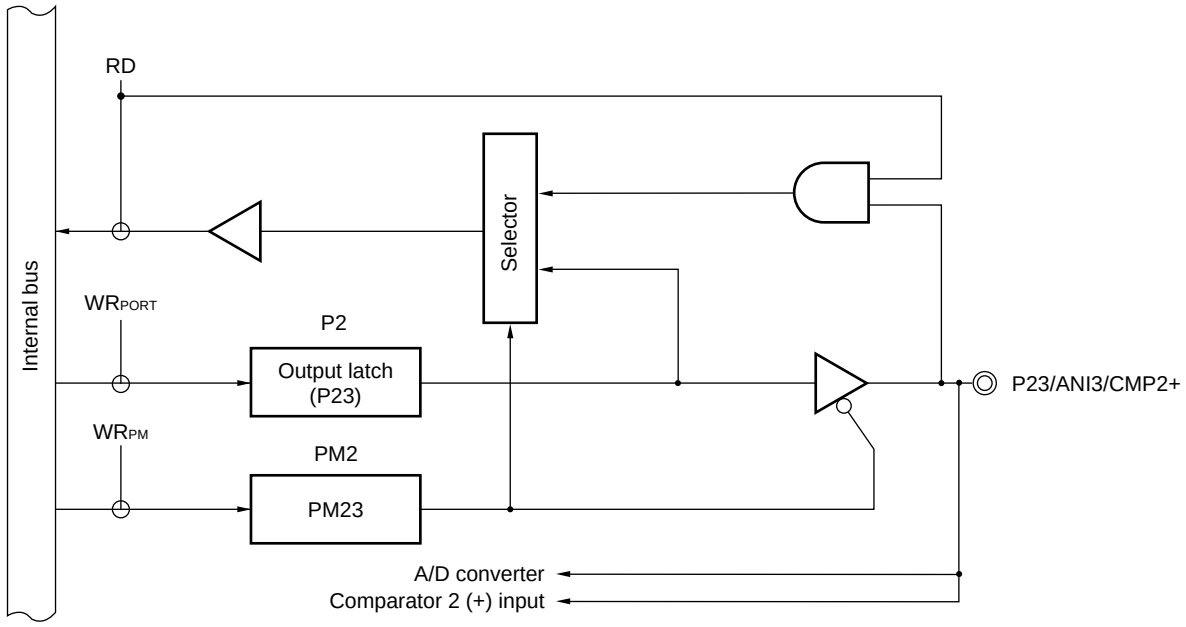


Figure 4-5. Block Diagram of P23



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

Figure 4-6. Block Diagram of P24

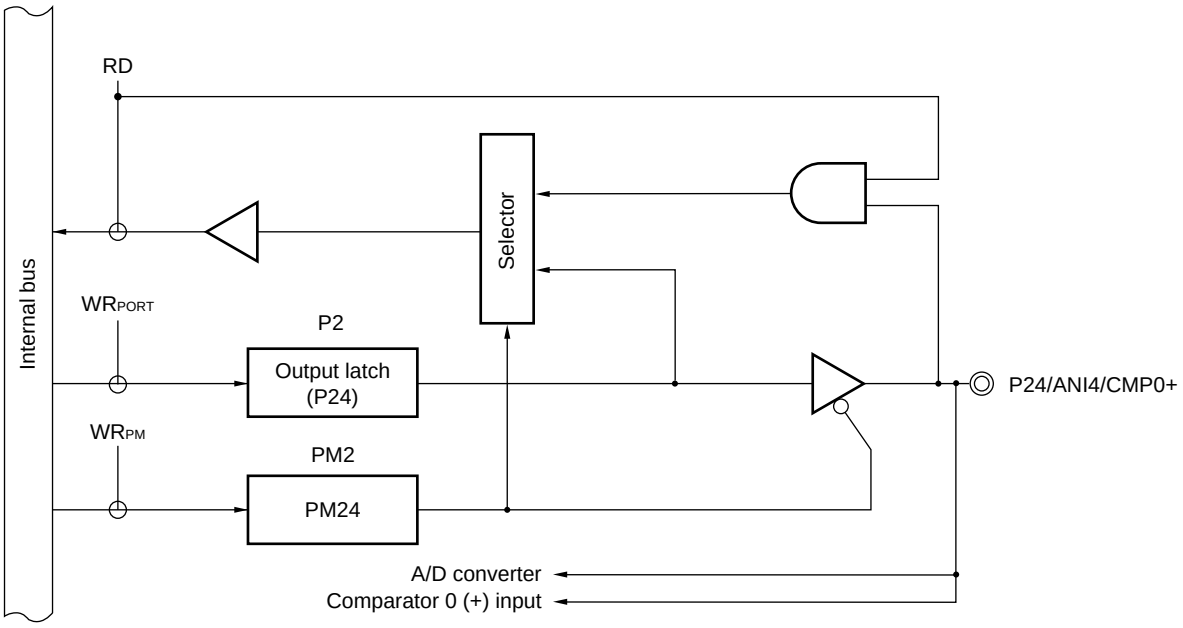
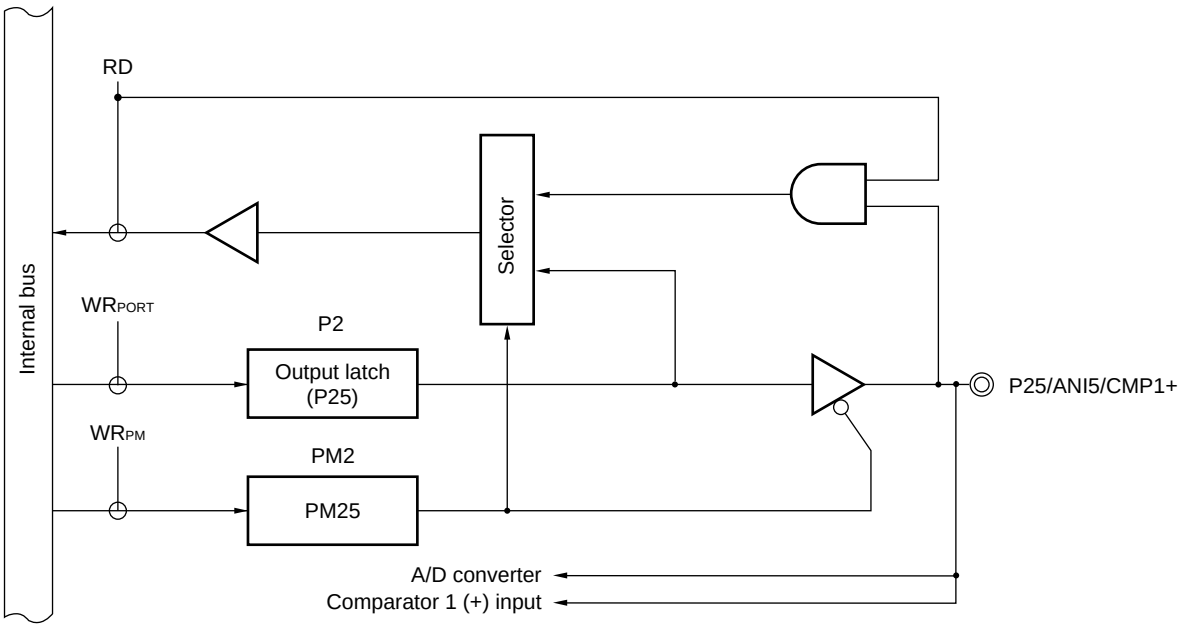
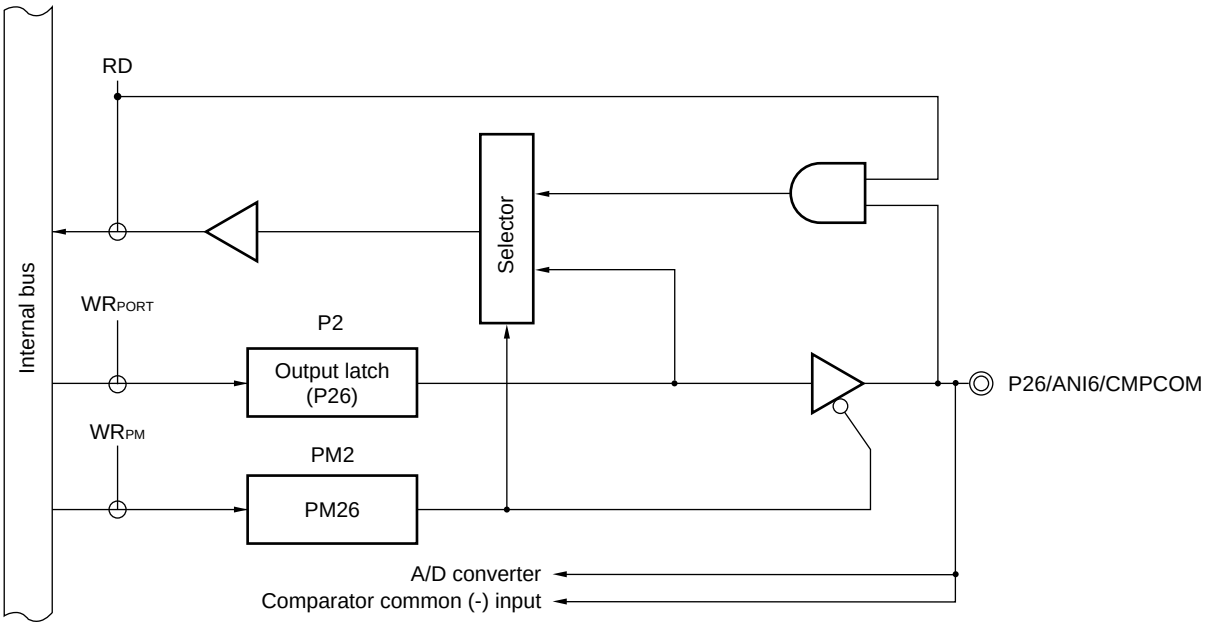


Figure 4-7. Block Diagram of P25



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

Figure 4-8. Block Diagram of P26



4.2.3 Port 3

78K0/FY2-L (16 Pins)	78K0/FA2-L (20 Pins)	78K0/FB2-L (30 Pins)
P30/TOH1/TI51/INTP1	P30/TOH1/TI51/INTP1	P30/TOH1/TI51/INTP1
–	P31/TOX10/INTP2/TOOLC1	P31/TOX00/INTP2/TOOLC1
–	P32/TOX11/INTP3/TOOLD1	P32/TOX01/INTP3/TOOLD1
–	–	P33/TOX10
–	–	P34/TOX11/INTP4
–	–	P35/$\overline{\text{SCK11}}$
–	–	P36/SI11
–	–	P37/SO11

Port 3 is an 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 P37 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, timer I/O, clock I/O and data I/O for serial interface, and clock input and data I/O for flash memory programmer/on-chip debugger.

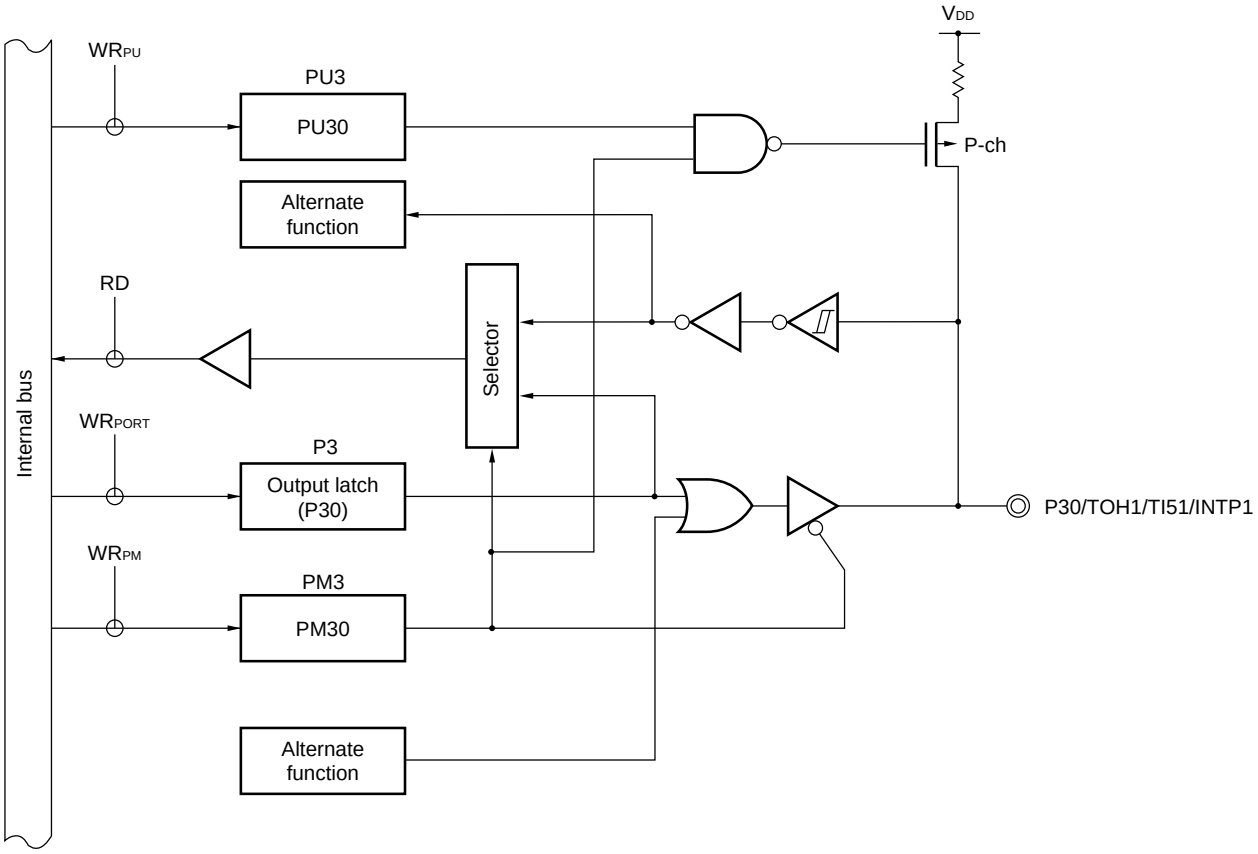
Reset signal generation sets port 3 to input mode.

Figures 4-9 to 4-16 show block diagrams of port 3.

Caution To use P35/$\overline{\text{SCK11}}$ and P37/SO11 of 78K0/FB2-L as general-purpose ports, set serial operation mode register 11 (CSIM11) and serial clock selection register 11 (CSIC11) to the default status (00H).

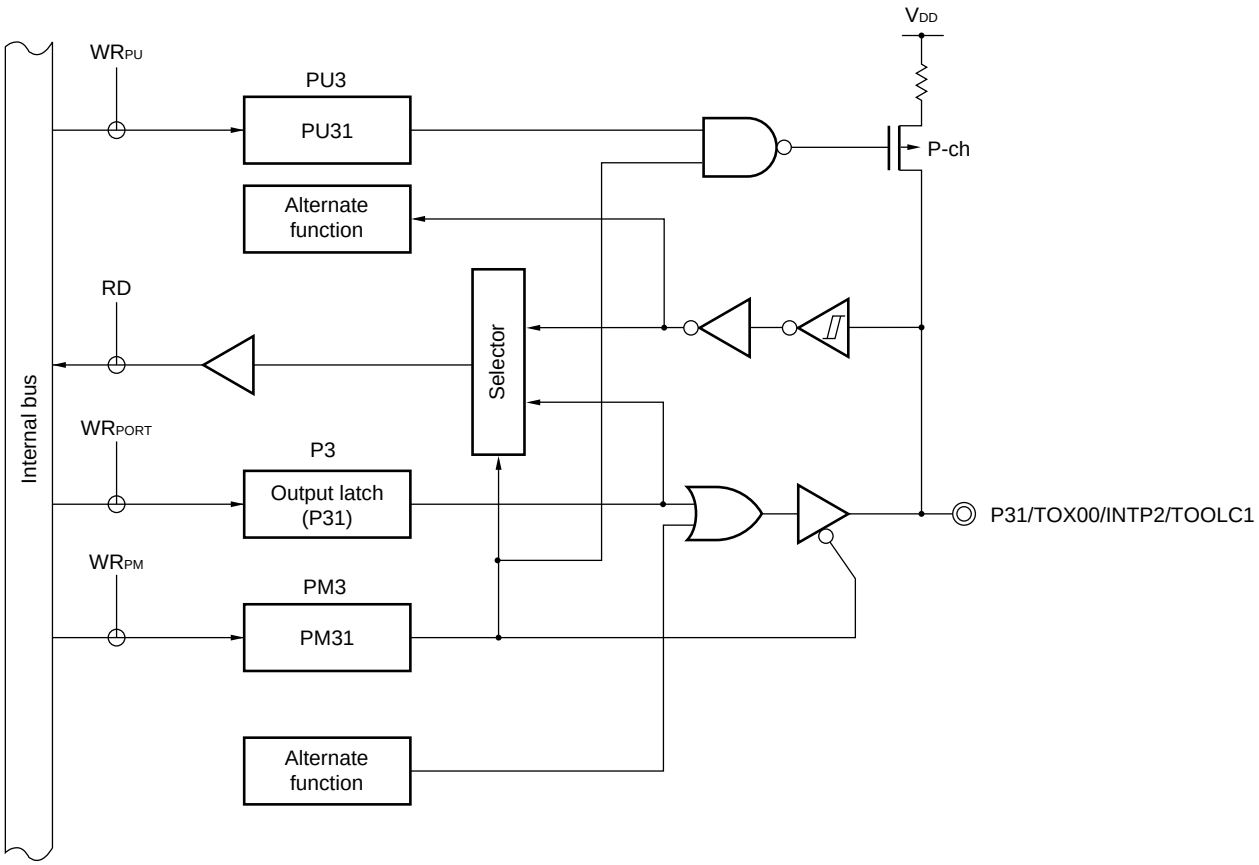
Remark For how to connect a flash memory programmer using TOOLC1/P31, TOOLD1/P32, refer to **CHAPTER 24 FLASH MEMORY**. For how to connect TOOLC1/P31, TOOLD1/P32 and an on-chip debug emulator, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

Figure 4-9. Block Diagram of P30



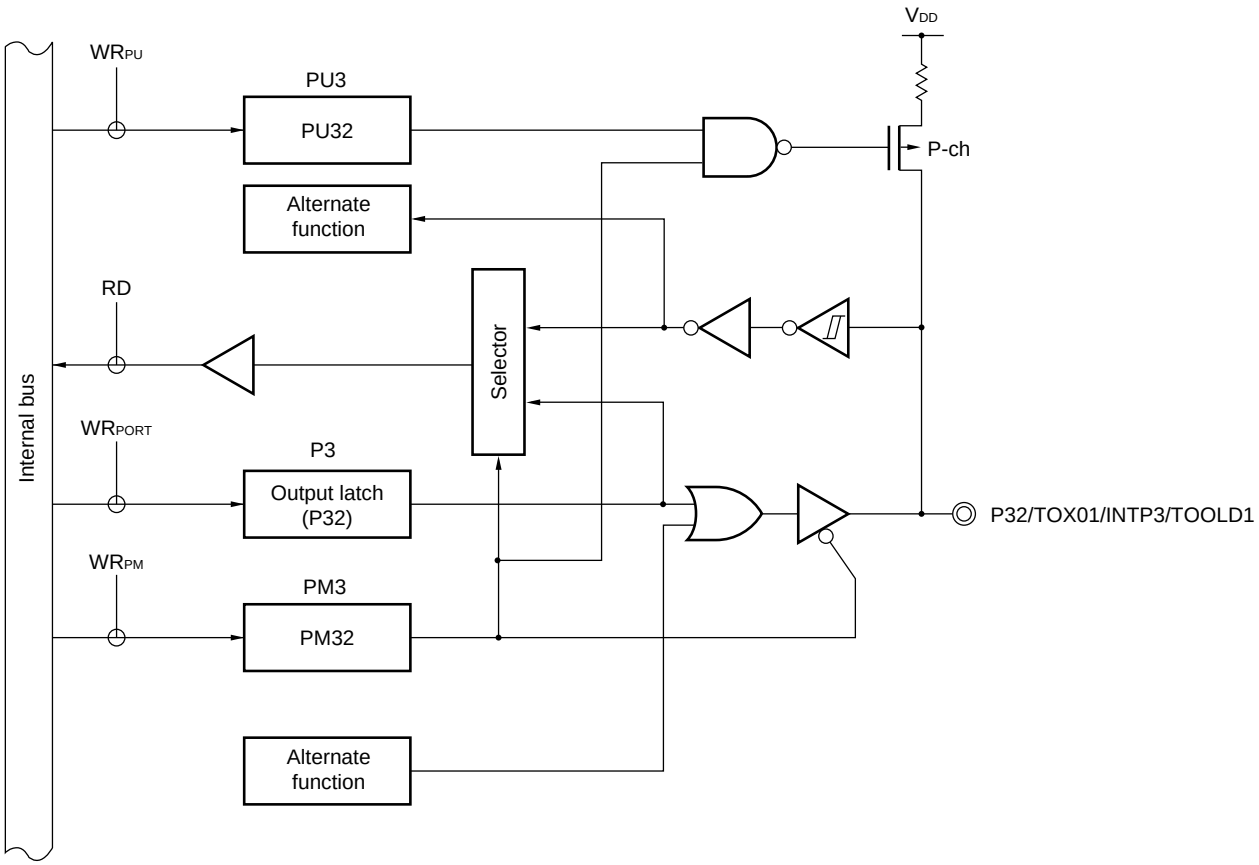
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-10. Block Diagram of P31



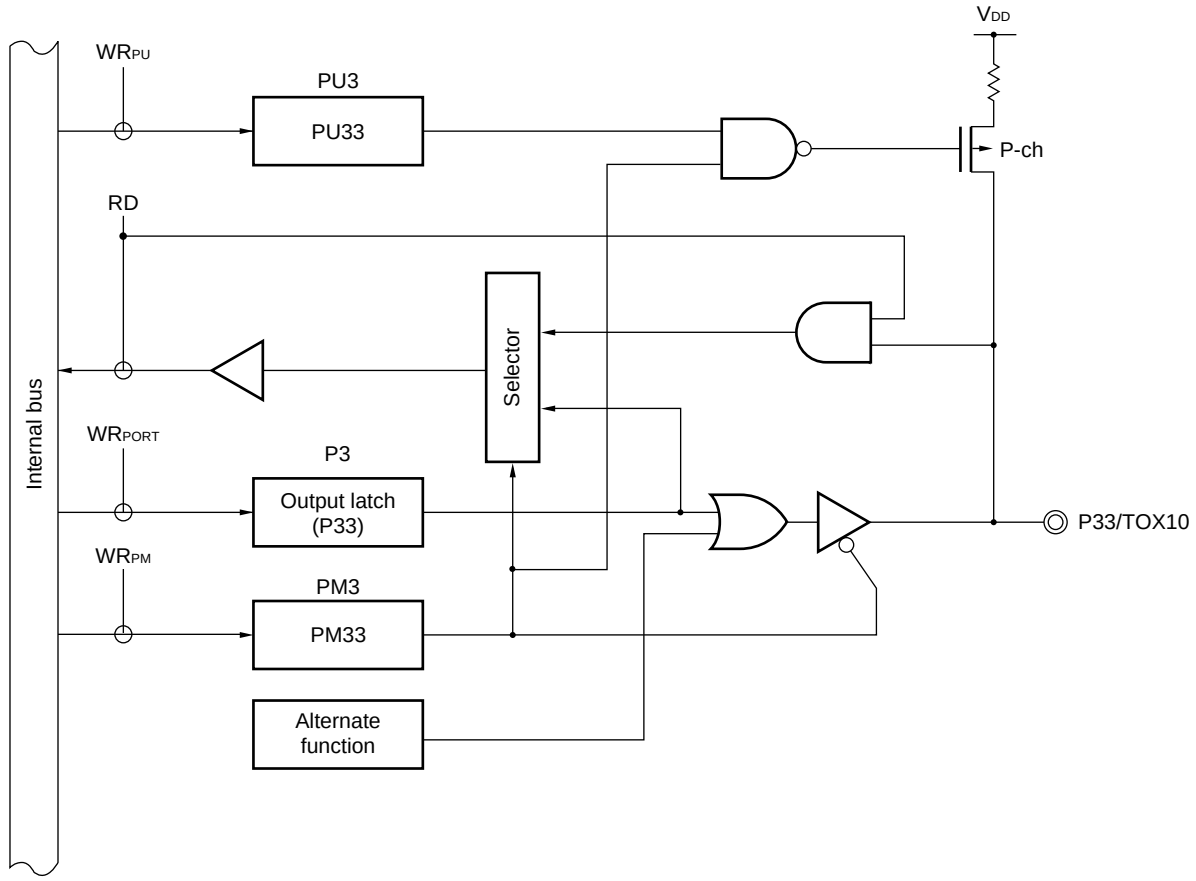
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-11. Block Diagram of P32



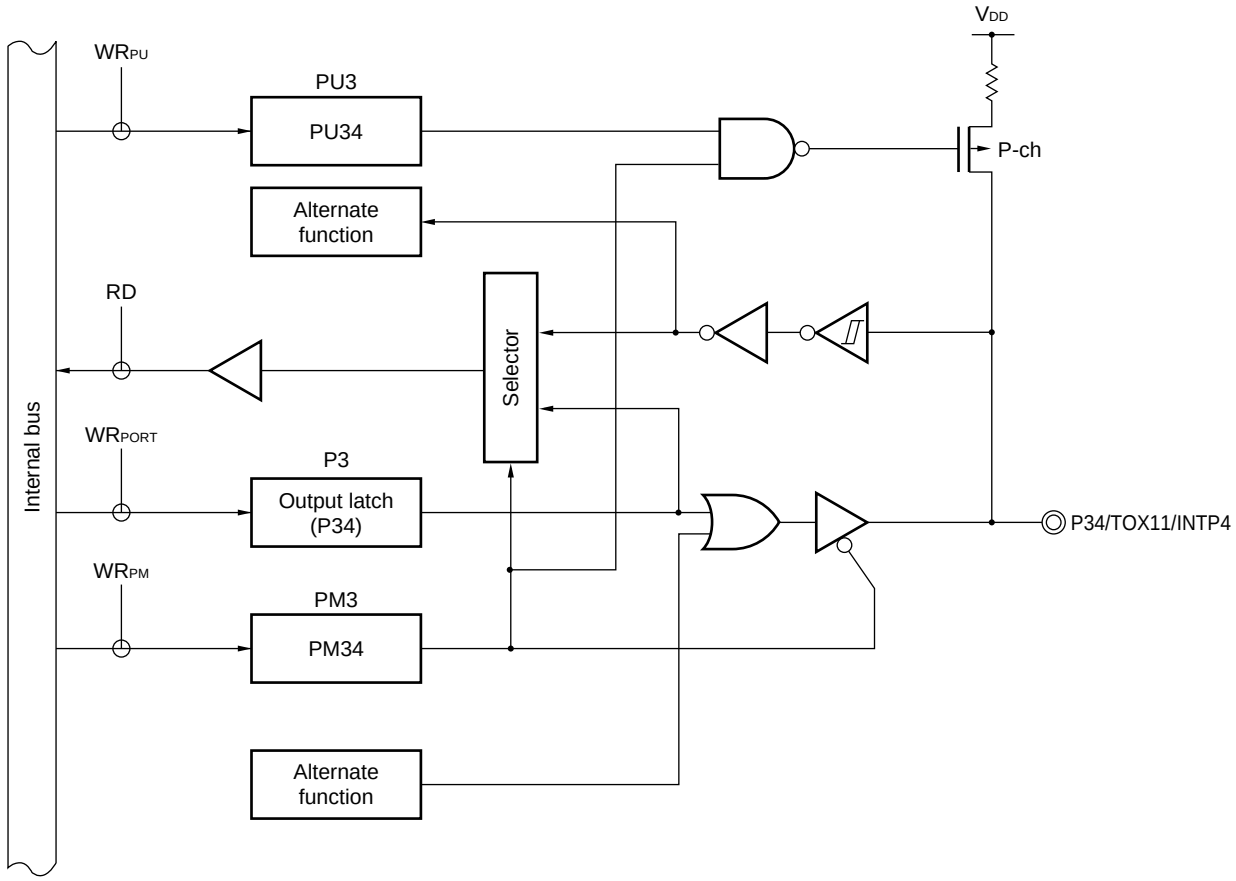
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-12. Block Diagram of P33



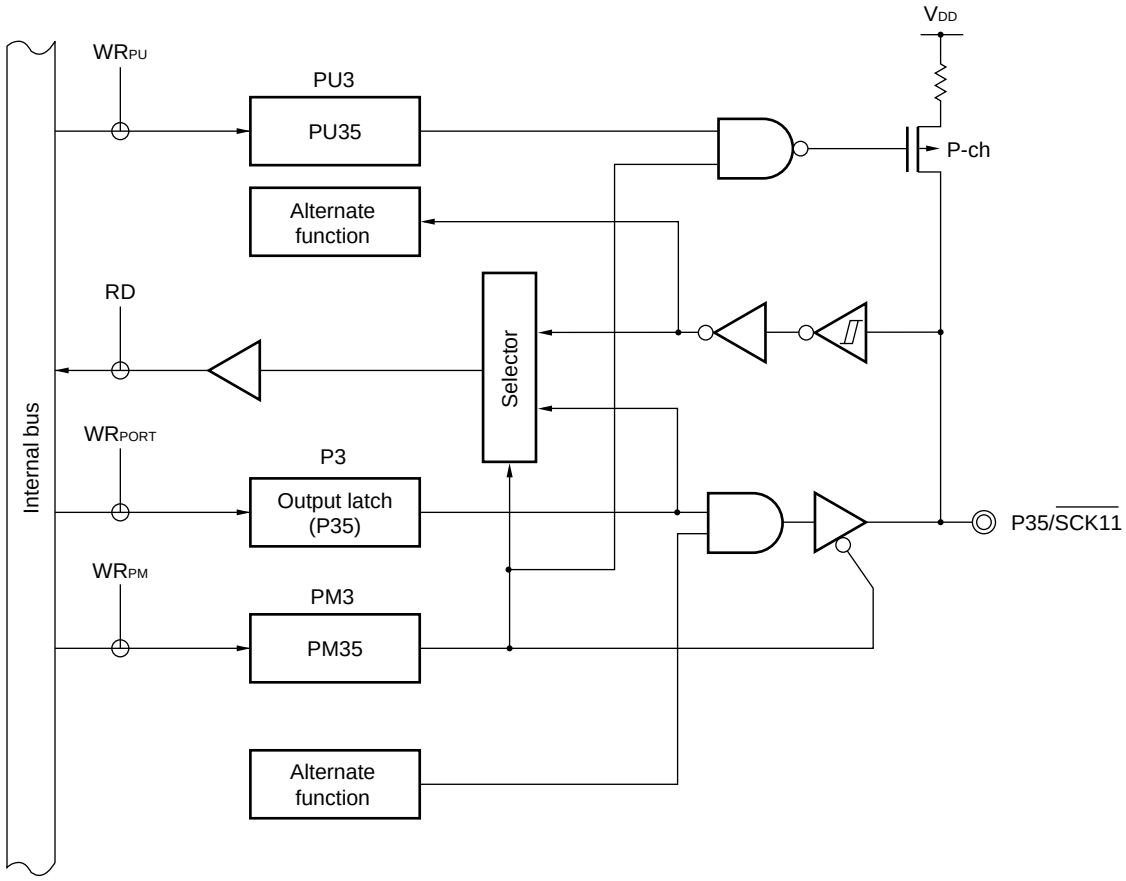
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-13. Block Diagram of P34



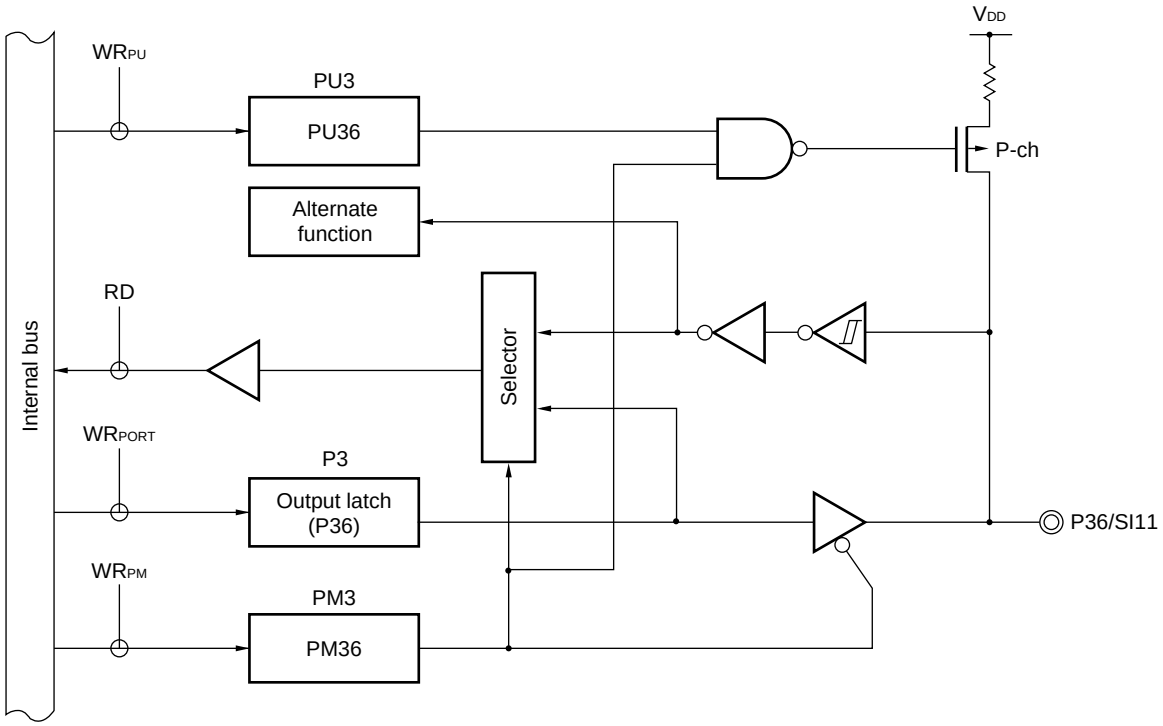
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-14. Block Diagram of P35



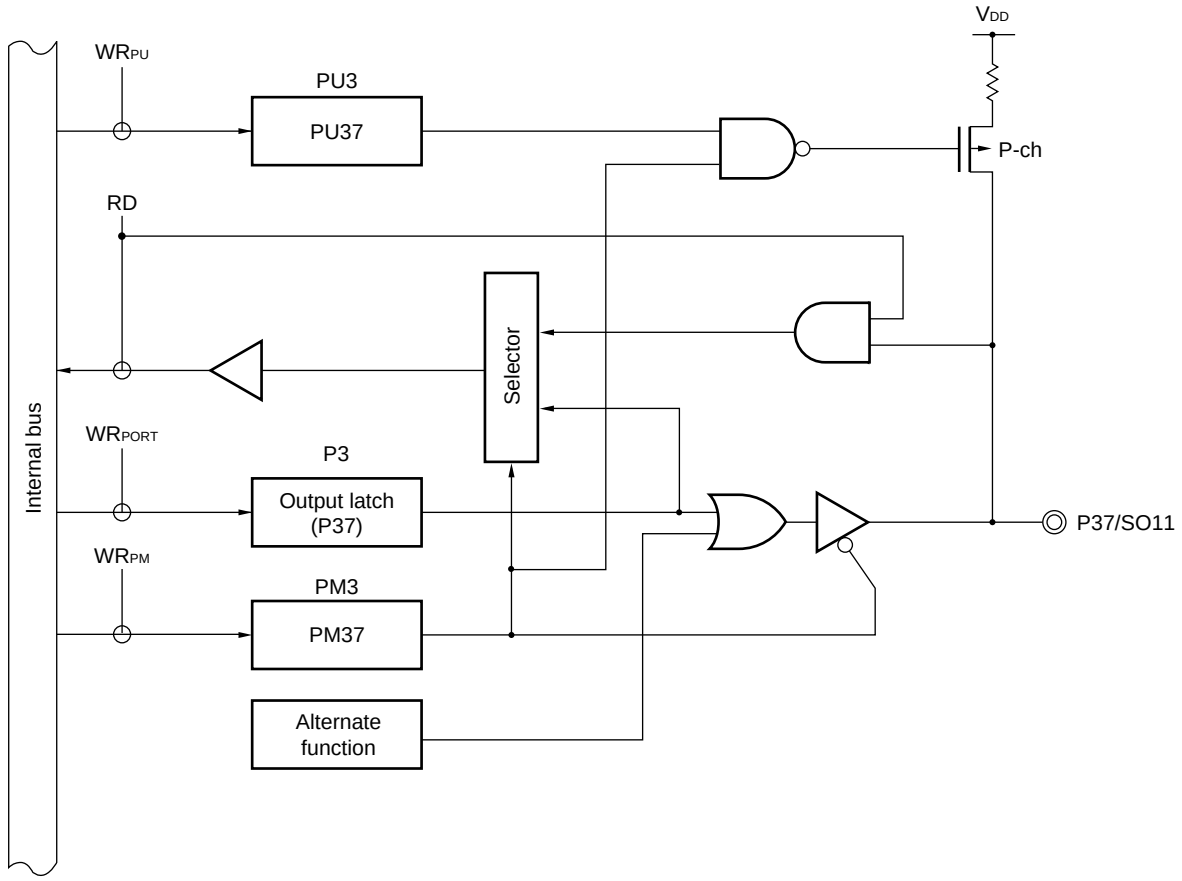
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-15. Block Diagram of P36



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

Figure 4-16. Block Diagram of P37



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

4.2.4 Port 6

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
P60/SCLA0/TxD6	P60/SCLA0/TxD6	P60/SCLA0/TxD6
P61/SDAA0/RxD6	P61/SDAA0/RxD6	P61/SDAA0/RxD6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6). When the P60 and P61 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 6 (PU6).

Input to the P60 and P61 pins can be specified through a normal input buffer or an SMBus input buffer in 1-bit units, using port input mode register 6 (PIM6).

Output from the P60 and P61 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 6 (POM6).

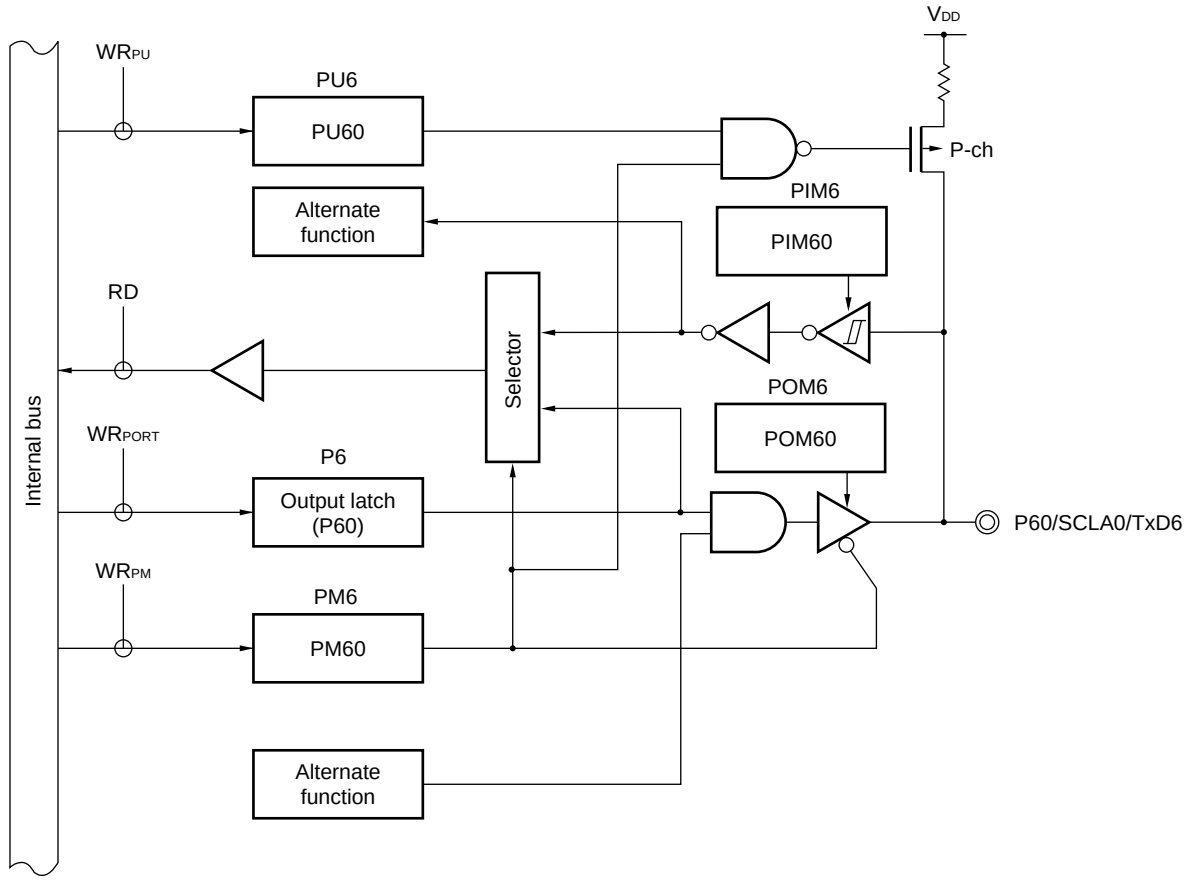
This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

Caution To use P60/SCLA0/TxD6 of 78K0/FY2-L, 78K0/FA2-L, and 78K0/FB2-L as general-purpose port, clear bit 0 (TXDLV6) of asynchronous serial interface control register 6 (ASICL6) to 0 (normal output of TxD6).

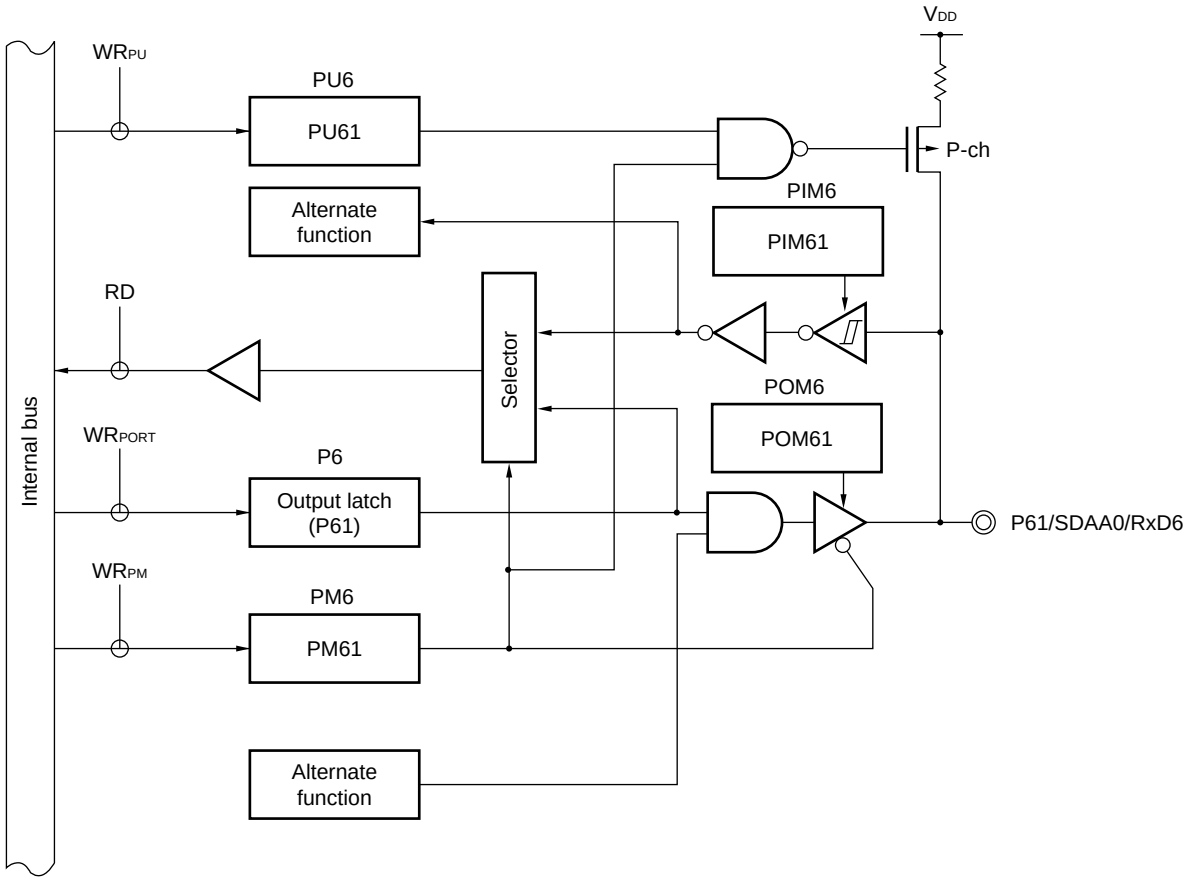
Figures 4-17 and 4-18 show block diagrams of port 6.

Figure 4-17. Block Diagram of P60



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PIM6: Port input mode register 6
POM6: Port output mode register 6
RD: Read signal
WR××: Write signal

Figure 4-18. Block Diagram of P61



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PIM6: Port input mode register 6
POM6: Port output mode register 6
RD: Read signal
WR××: Write signal

4.2.5 Port 7

78K0/FY2-L (16-pin)	78K0/FA2-L (20-pin)	78K0/FB2-L (30-pin)
–	–	P70/ANI8

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).

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

When using P70/ANI8, set the registers according to the pin function to be used (refer to **Table 4-11**).

Table 4-11. Setting Functions of P70/ANI8 Pin

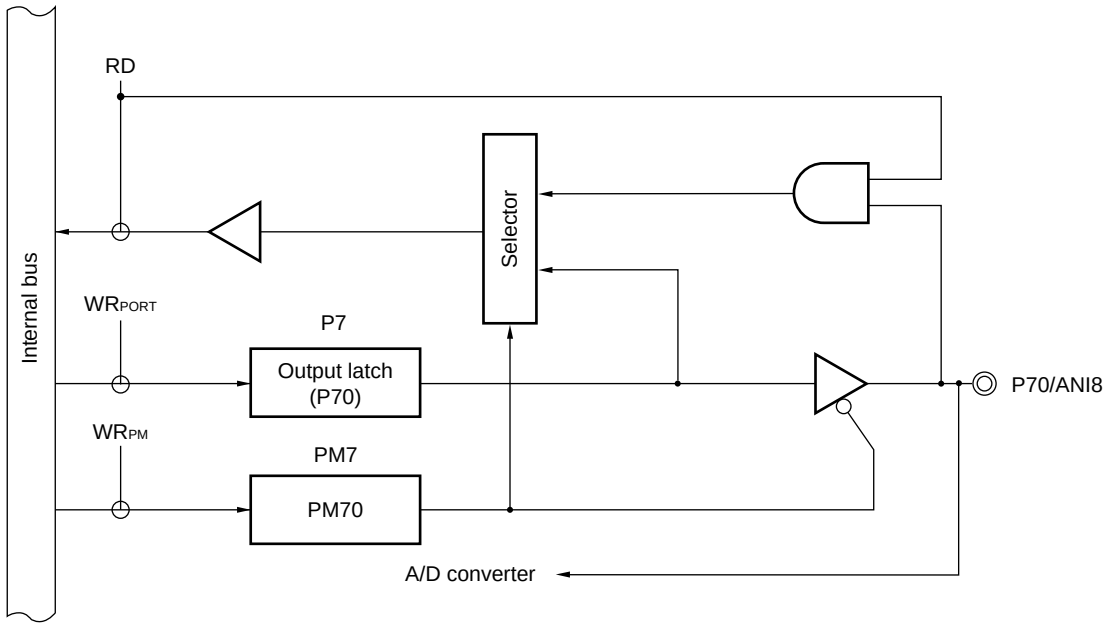
ADPC1 Register	PM7 Register	ADS Register	P70/ANI8 Pin
Digital I/O selection	Input mode	Selects ANI8.	Setting prohibited
		Does not select ANI8.	Digital input
	Output mode	Selects ANI8.	Setting prohibited
		Does not select ANI8.	Digital output
Analog input selection	Input mode	Selects ANI8.	Analog input (to be converted into digital signal)
		Does not select ANI8.	Analog input (not to be converted into digital signal)
	Output mode	–	Setting prohibited

Remark ADPC1: A/D port configuration register 1
PM7: Port mode register 7
ADS: Analog input channel specification register

Reset signal generation sets port 7 to analog input.

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

Figure 4-19. Block Diagram of P70



P7: Port register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

4.2.6 Port 12

78K0/FY2-L (16 Pins)	78K0/FA2-L (20 Pins)	78K0/FB2-L (30 Pins)
P121/X1/TOOLC0	P121/X1/TOOLC0	P121/X1/TOOLC0/<TI000>/<INTP0>
P122/X2/EXCLK/TOOLD0	P122/X2/EXCLK/TOOLD0	P122/X2/EXCLK/TOOLD0

Remark Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

P121 and P122 function as an input port.

This port can also be used as pins for potential input for connecting resonator for main system clock, external clock input for main system clock, and clock input and data I/O for flash memory programmer/on-chip debugger.

The timer input or external interrupt request input can be assigned to P121 of the 78K0/FY2-L by setting the port alternate switch control register (MUXSEL).

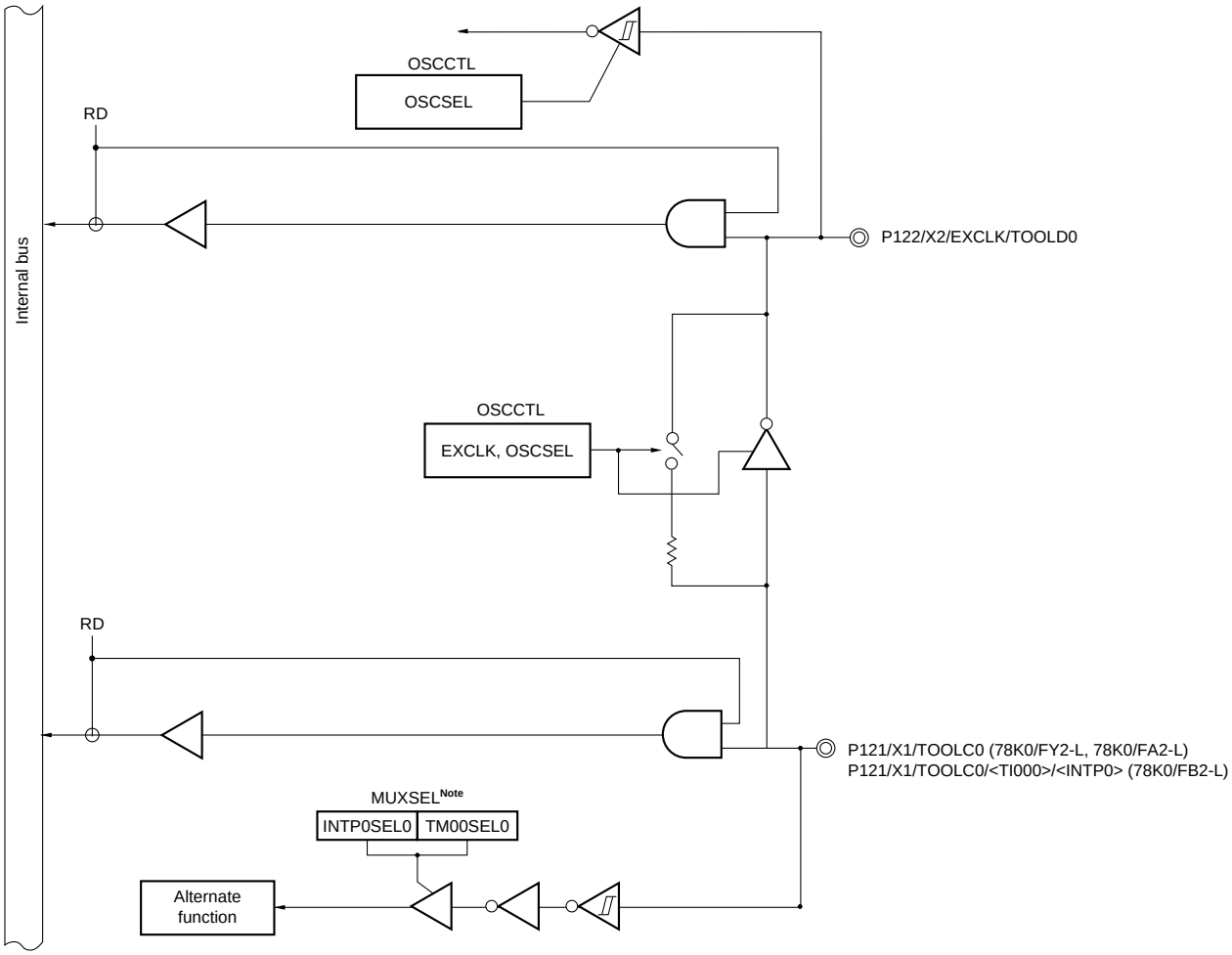
Reset signal generation sets port 12 to input mode.

Figure 4-20 shows block diagrams of port 12.

Caution When using the P121 and P122 pins to connect a resonator for the main system clock (X1, X2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, refer to 5.3 (1) Clock operation mode select register (OSCCTL)). The reset value of OSCCTL is 00H (all of the P121 and P122 pins are input port pins).

Remark For how to connect a flash memory programmer using TOOLC0/X1, TOOLD0/X2, refer to **CHAPTER 24 FLASH MEMORY**. For how to connect TOOLC0/X1, TOOLD0/X2 and an on-chip debug emulator, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

Figure 4-20. Block Diagram of P121 and P122



MUXSEL: Port alternate switch control register
OSCCTL: Clock operation mode select register
RD: Read signal
WR××: Write signal

Note 78K0/FB2-L only.

4.3 Registers Controlling Port Function

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

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode register 6 (PIM6)
- Port output mode register 6 (POM6)
- A/D port configuration registers n (ADPCn)
- Port alternate switch control register (MUXSEL)

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(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 4-21. Format of Port Mode Register (78K0/FY2-L)

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
PM2	1	1	1	1	PM23[Note]	PM22[Note]	PM21[Note]	PM20[Note]	FF22H	FFH	R/W
PM3	1	1	1	1	1	1	1	PM30	FF23H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 2, 3, 6; n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note If this pin is set as an analog input by using the ADPC0 register, be sure to set it to input mode.

Caution Be sure to set bits 2 to 7 of PM0, bits 4 to 7 of PM2, bits 1 to 7 of PM3, bits 2 to 7 of PM6 to 1.

Figure 4-22. Format of Port Mode Register (78K0/FA2-L)

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
PM2	1	1	PM25[Note]	PM24[Note]	PM23[Note]	PM22[Note]	PM21[Note]	PM20[Note]	FF22H	FFH	R/W
PM3	1	1	1	1	1	PM32	PM31	PM30	FF23H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 2, 3, 6; n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note If this pin is set as an analog input by using the ADPC0 register, be sure to set it to input mode.

Caution Be sure to set bits 2 to 7 of PM0, bits 6, 7 of PM2, bits 3 to 7 of PM3, bits 2 to 7 of PM6 to 1.

Figure 4-23. Format of Port Mode Register (78K0/FB2-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	PM02	PM01	PM00	FF20H	FFH	R/W
PM2	PM27Note	PM26Note	PM25Note	PM24Note	PM23Note	PM22Note	PM21Note	PM20Note	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	1	1	1	1	1	PM70Note	FF27H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 2, 3, 6, 7; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note If this pin is set as an analog input by using the ADPC1 or ADPC0 register, be sure to set it to input mode.

Caution Be sure to set bits 3 to 7 of PM0, bits 2 to 7 of PM6, bits 1 to 7 of PM7 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 4-24. Format of Port Register (78K0/FY2-L)

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
P2	0	0	0	0	P23Note 1	P22Note 1	P21Note 1	P20Note 1	FF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	0	P30	FF03H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (output latch)	R/W
P12	0	0	0	0	0	P122Note 2	P121Note 2	0	FF0CH	00H	R

Pmn	m = 0, 2, 3, 6, 12; n = 0 to 3	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Notes 1. If this pin is set as an analog input and to input mode, do not access the output latch.

2. “0” is always read from the output latch of the pin in the X1 oscillation mode or external clock input mode.

Figure 4-25. Format of Port Register (78K0/FA2-L)

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
P2	0	0	P25Note 1	P24Note 1	P23Note 1	P22Note 1	P21Note 1	P20Note 1	FF02H	00H (output latch)	R/W
P3	0	0	0	0	0	P32	P31	P30	FF03H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (output latch)	R/W
P12	0	0	0	0	0	P122Note 2	P121Note 2	0	FF0CH	00H	R

Pmn	m = 0, 2, 3, 6, 12; n = 0 to 5	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Notes 1. If this pin is set as an analog input and to input mode, do not access the output latch.

2. “0” is always read from the output latch of the pin in the X1 oscillation mode or external clock input mode.

Figure 4-26. Format of Port Register (78K0/FB2-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	P02	P01	P00	FF00H	00H (output latch)	R/W
P2	P27 Note 1	P26 Note 1	P25 Note 1	P24 Note 1	P23 Note 1	P22 Note 1	P21 Note 1	P20 Note 1	FF02H	00H (output latch)	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (output latch)	R/W
P7	0	0	0	0	0	0	0	P70	FF07H	00H (output latch)	R/W
P12	0	0	0	0	0	P122 Note 2	P121 Note 2	0	FF0CH	00H	R

Pmn	m = 0, 2, 3, 6, 7, 12; 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

Notes 1. If this pin is set as an analog input and to input mode, do not access the output latch.

2. “0” is always read from the output latch of the pin in the X1 oscillation mode or external clock input mode.

(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 4-27. Format of Pull-up Resistor Option Register (78K0/FY2-L)

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
PU3	0	0	0	0	0	0	0	PU30	FF33H	00H	R/W
PU6	0	0	0	0	0	0	PU61	PU60	FF36H	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 3, 6; n = 0, 1)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Figure 4-28. Format of Pull-up Resistor Option Register (78K0/FA2-L)

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
PU3	0	0	0	0	0	PU32	PU31	PU30	FF33H	00H	R/W
PU6	0	0	0	0	0	0	PU61	PU60	FF36H	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 3, 6; n = 0 to 2)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Figure 4-29. Format of Pull-up Resistor Option Register (78K0/FB2-L)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	PU02	PU01	PU00	FF30H	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU6	0	0	0	0	0	0	PU61	PU60	FF36H	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 3, 6; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

(4) Port input mode register 6 (PIM6)

This register sets the input buffer of P60 or P61 in 1-bit units.

When using an input compliant with the SMBus specifications in I^2C communication, set PIM60 and PIM61 to 1.

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

Reset signal generation clears this register to 00H.

Figure 4-30. Format of Port Input Mode Register 6 (PIM6)

Address: FF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM6	0	0	0	0	0	0	PIM61	PIM60

PIM6n	P6n pin input buffer selection (n = 0, 1)
0	Normal input (Schmitt) buffer
1	SMBus input buffer

(5) Port output mode register 6 (POM6)

This register sets the output mode of P60 and P61 in 1-bit units.

During I^2C communication, set POM60 and POM61 to 1.

When using the P60/TxD6/SCLA0 pin as the data output of serial interface UART6, clear POM60 to 0.

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

Reset signal generation clears this register to 00H.

Figure 4-31. Format of Port Output Mode Register 6 (POM6)

Address: FF2AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM6	0	0	0	0	0	0	POM61	POM60

POM6n	P6n pin output mode selection (n = 0, 1)
0	Normal output (CMOS output) mode
1	N-ch open drain output (V_{DD} tolerance) mode

(6) A/D port configuration registers n (ADPCn)

ADPC0 switches the P20/ANI0 to P27/ANI7 pins to digital I/O or analog input of port. Each bit of ADPC0 corresponds to a pin of port 2 and can be specified in 1-bit units.

ADPC1 switches the P70/ANI8 pin to digital I/O or analog input of port. Each bit of ADPC1 corresponds to the P70 pin in port 7 and can be specified in 1-bit units.

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

Reset signal generation clears ADPCn to 00H.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 4-32. Format of A/D Port Configuration Register 0 (ADPC0)

(a) 78K0/FY2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPCS3	ADPCS2	ADPCS1	ADPCS0

(b) 78K0/FA2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	ADPCS5	ADPCS4	ADPCS3	ADPCS2	ADPCS1	ADPCS0

(c) 78K0/FB2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	ADPCS7	ADPCS6	ADPCS5	ADPCS4	ADPCS3	ADPCS2	ADPCS1	ADPCS0

ADPCSn	Digital I/O or analog input selection (n = 0 to 7)
0	Analog input
1	Digital I/O

Cautions 1. Set the pin set to analog input to the input mode by using port mode register 2 (PM2).

2. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the peripheral hardware clock is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

Figure 4-33. Format of A/D Port Configuration Register 1 (ADPC1)
(78K0/FB2-L Only)

Address: FF2FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC1	0	0	0	0	0	0	0	ADPCS8

ADPCS8	Digital I/O or analog input selection
0	Analog input
1	Digital I/O

Cautions 1. Set the pin set to analog input to the input mode by using port mode register 7 (PM7).
2. If data is written to ADPC1, a wait cycle is generated. Do not write data to ADPC1 when the peripheral hardware clock is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(7) Port alternate switch control register (MUXSEL)[Note]

This register assigns the pin function.

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

Reset signal generation clears MUXSEL to 00H.

Note 78K0/FB2-L only

Figure 4-34. Format of Port Alternate Switch Control Register (MUXSEL)
(78K0/FB2-L Only)

Address: FF39H After reset: 00H R/W

Symbol	7	<6>	5	<4>	3	2	1	0
MUXSEL	0	INTP0SEL0	0	TM00SEL0	0	0	0	0

INTP0SEL0	External interrupt input (INTP0) pin assignment
0	(default)
1	P121/INTP0

TM00SEL0	16-bit timer/event counter 00 input (TI000) pin assignment
0	(default)
1	P121/TI000

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 Port Mode Register and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the port mode register and output latch as shown in **Tables 4-12** to **4-14**.

Table 4-12. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FY2-L) (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI000	Input	1	×
	INTP0	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P20	ANI0Note 1	Input	1	×
P21	ANI1Note 1	Input	1	×
P22	ANI2Note 1	Input	1	×
P23	ANI3Note 1	Input	1	×
	CMP2+Note 2	Input	1	×
P30	INTP1	Input	1	×
	TI51	Input	1	×
	TOH1	Output	0	0
P60	SCLA0Notes 3, 4	I/O	0	1
	TxD6Note 5	Output	0	1
P61	SDAA0Notes 3, 4	I/O	0	1
	RxD6	Input	1	×

Notes 1. The pin function can be selected by using ADPC0 register, PM2 register, and ADS register. Refer to **Tables 4-6** to **4-8** of **4.2.3 Port 2**.

2. The pin function can be selected by using ADPC0 register, PM2 register, ADS register, and CMPmEN (m = 0 to 2) bit. Refer to **Table 4-8** in **4.2.2 Port 2**.

3. During I^2C communication, set SCLA0 and SDAA0 to N-ch open drain output (V_{DD} tolerance) mode by using POM6 register (refer to **4.3 (5) Port output mode register 6 (POM6)**).

4. When using an input compliant with the SMBus Specifications in I^2C communication, select the SMBus input buffer by using PIM6 register (refer to **4.3 (4) Port input mode register 6 (PIM6)**).

5. During UART communication, set TxD6 to normal output (CMOS output) mode by using POM6 register (refer to **4.3 (5) Port output mode register 6 (POM6)**).

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

Table 4-12. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FY2-L) (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P121	X1[Note]	–	×	×
	TOOLC0	Input	×	×
P122	X2[Note]	–	×	×
	EXCLK[Note]	Input	×	×
	TOOLD0	I/O	×	×

Note When using the P121 and P122 pins to connect a resonator for the main system clock (X1, X2) or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode or external clock input mode must be set by using OSCCTL register (for details, refer to **5.3 (1) Clock operation mode select register (OSCCTL)**). The reset value of OSCCTL is 00H (both P121 and P122 are input port pins).

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

Table 4-13. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FA2-L) (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI000	Input	1	×
	INTP0	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P20	ANI0Note 1	Input	1	×
P21	ANI1Note 1	Input	1	×
P22	ANI2Note 1	Input	1	×
P23	ANI3Note 2	Input	1	×
	CMP2+Note 2	Input	1	×
P24	ANI4Note 2	Input	1	×
	CMP0+Note 2	Input	1	×
P25	ANI5Note 2	Input	1	×
	CMP1+Note 2	Input	1	×
P30	INTP1	Input	1	×
	TI51	Input	1	×
	TOH1	Output	0	0
P31	INTP2	Input	1	×
	TOOLC1	Input	×	×
P32	INTP3	Input	1	×
	TOOLD1	I/O	×	×
P60	SCLA0Notes 3, 4	I/O	0	1
	TxD6Note 5	Output	0	1
P61	SDAA0Notes 3, 4	I/O	0	1
	RxD6	Input	1	×

Notes 1. The pin function can be selected by using ADPC0 register, PM2 register, and ADS register. Refer to **Tables 4-6** and **4-7** of **4.2.2 Port 2**.

2. The pin function can be selected by using ADPC0 register, PM2 register, ADS register, and CMPmEN (m = 0 to 2) bit. Refer to **Table 4-8** in **4.2.2 Port 2**.

3. During I^2C communication, set SCLA0 and SDAA0 to N-ch open drain output (V_{DD} tolerance) mode by using POM6 register (refer to **4.3 (5) Port output mode register 6 (POM6)**).

4. When using an input compliant with the SMBus Specifications in I^2C communication, select the SMBus input buffer by using PIM6 register (refer to **4.3 (4) Port input mode register 6 (PIM6)**).

5. During UART communication, set TxD6 to normal output (CMOS output) mode by using POM6 register (refer to **4.3 (5) Port output mode register 6 (POM6)**).

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

Table 4-13. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FA2-L) (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P121	X1[Note]	–	×	×
	TOOLC0	Input	×	×
P122	X2[Note]	–	×	×
	EXCLK[Note]	Input	×	×
	TOOLD0	I/O	×	×

Note When using the P121 and P122 pins to connect a resonator for the main system clock (X1, X2) or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode or external clock input mode must be set by using OSCCTL register (for details, refer to **5.3 (1) Clock operation mode select register (OSCCTL)**). The reset value of OSCCTL is 00H (both P121 and P122 are input port pins).

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

Table 4-14. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FB2-L) (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI000	Input	1	×
	INTP0	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P02	$\overline{SSI11}$	Input	1	×
	INTP5	Input	1	×
P20	ANI0Note 1	Input	1	×
P21	ANI1Note 1	Input	1	×
P22	ANI2Note 1	Input	1	×
P23	ANI3Note 2	Input	1	×
	CMP2+Note 2	Input	1	×
P24	ANI4Note 2	Input	1	×
	CMP0+Note 2	Input	1	×
P25	ANI5Note 2	Input	1	×
	CMP1+Note 2	Input	1	×
P26	ANI6Note 3	Input	1	×
	CMPCOMNote 3	Input	1	×
P27	ANI7Note 4	Input	1	×
P30	INTP1	Input	1	×
	TI51	Input	1	×
	TOH1	Output	0	0

Notes 1. The pin function can be selected by using ADPC0 register, PM2 register, and ADS register. Refer to **Tables 4-6 and 4-7** in **4.2.2 Port 2**.

2. The pin function can be selected by using ADPC0 register, PM2 register, ADS register, and CMPmEN (m = 0 to 2) bit. Refer to **Table 4-8** in **4.2.2 Port 2**.

3. The pin function can be selected by using ADPC0 register, PM2 register, ADS register, and CmMODSEL1 and CmMODSEL0 (m = 0 to 2) bits. Refer to **Table 4-9** in **4.2.2 Port 2**.

4. The pin function can be selected by using ADPC0 register, PM2 register, and ADS register. Refer to **Table 4-10** in **4.2.2 Port 2**.

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

Table 4-14. Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FB2-L) (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P31	TOX00	Output	0	0
	INTP2	Input	1	×
	TOOLC1	Input	×	×
P32	TOX01	Output	0	0
	INTP3	Input	1	×
	TOOLD1	I/O	×	×
P33	TOX10	Output	0	0
P34	TOX11	Output	0	0
	INTP4	Input	1	×
P35	$\overline{SCK11}$	Input	1	×
		Output	0	1
P36	SI11	Input	1	×
P37	SO11	Output	0	0
P60	SCLA0Notes 1, 2	I/O	0	1
	TxD6Note 3	Output	0	1
P61	SDAA0Notes 1, 2	I/O	0	1
	RxD6	Input	1	×
P70	ANI8Note 4	Input	1	×
P121	X1Note 5	–	×	×
	TOOLC0	Input	×	×
	<TI000>	Input	×	×
	<INTP0>	Input	×	×
P122	X2Note 5	–	×	×
	EXCLKNote 5	Input	×	×
	TOOLD0	I/O	×	×

Notes 1. During I^2C communication, set SCLA0 and SDAA0 to N-ch open drain output (V_{DD} tolerance) mode by using POM6 register (refer to **4.3 (5) Port output mode register 6 (POM6)**).

2. When using an input compliant with the SMBus Specifications in I^2C communication, select the SMBus input buffer by using PIM6 register (refer to **4.3 (4) Port input mode register 6 (PIM6)**).

3. During UART communication, set TxD6 to normal output (CMOS output) mode by using POM6 register (refer to **4.3 (5) Port output mode register 6 (POM6)**).

4. The pin function can be selected by using ADPC1 register, PM7 register, and ADS register. Refer to **Table 4-11** of **4.2.5 Port 7**.

5. When using the P121 and P122 pins to connect a resonator for the main system clock (X1, X2) or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode or external clock input mode must be set by using OSCCTL register (for details, refer to **5.3 (1) Clock operation mode select register (OSCCTL)**). The reset value of OSCCTL is 00H (both P121 and P122 are input port pins).

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

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

When a 1-bit memory 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 P20 is an output port, P21 to P27 are input ports (all pin statuses are high level), and the port latch value of port 2 is 00H, if the output of output port P20 is changed from low level to high level via a 1-bit memory manipulation instruction, the output latch value of port 2 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 memory manipulation instruction is executed in the following order in the 78K0/Fx2-L microcontrollers.

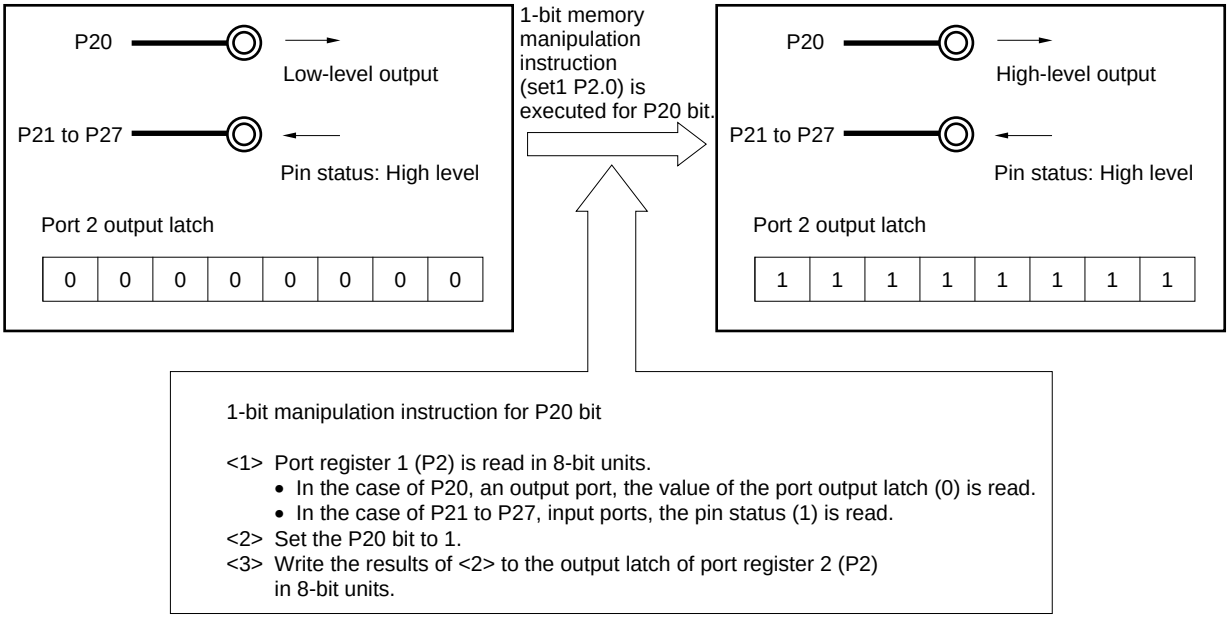
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P20, which is an output port, is read, while the pin statuses of P21 to P27, which are input ports, are read. If the pin statuses of P21 to P27 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 4-35. 1-bit Memory Manipulation Instruction (P20)



Remark The following instructions are 1-bit memory manipulation instructions.

- MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT11, NOT1

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

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).

<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_{IH} = 4 MHz (TYP.)/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/PLL control register (RCM).

<3> External main system clock input

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

<4> Multiplication function using PLL (phase locked loop)

If 4 MHz is selected for the main system clock, the PLL mode can be used. In this mode, a clock of ten times the main system clock times 1/2 (20 MHz) can be supplied.

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

- **Internal low-speed oscillator**

This circuit oscillates a clock of f_{IL} = 30 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/PLL control 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 (when f_{IL}, $f_{IL}/2^6$, or $f_{IL}/2^{15}$ is selected)

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EXCLK}: External main system clock frequency
f_{IL}: 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/PLL control 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	High-speed system clock oscillator Internal high-speed oscillator Internal low-speed oscillator

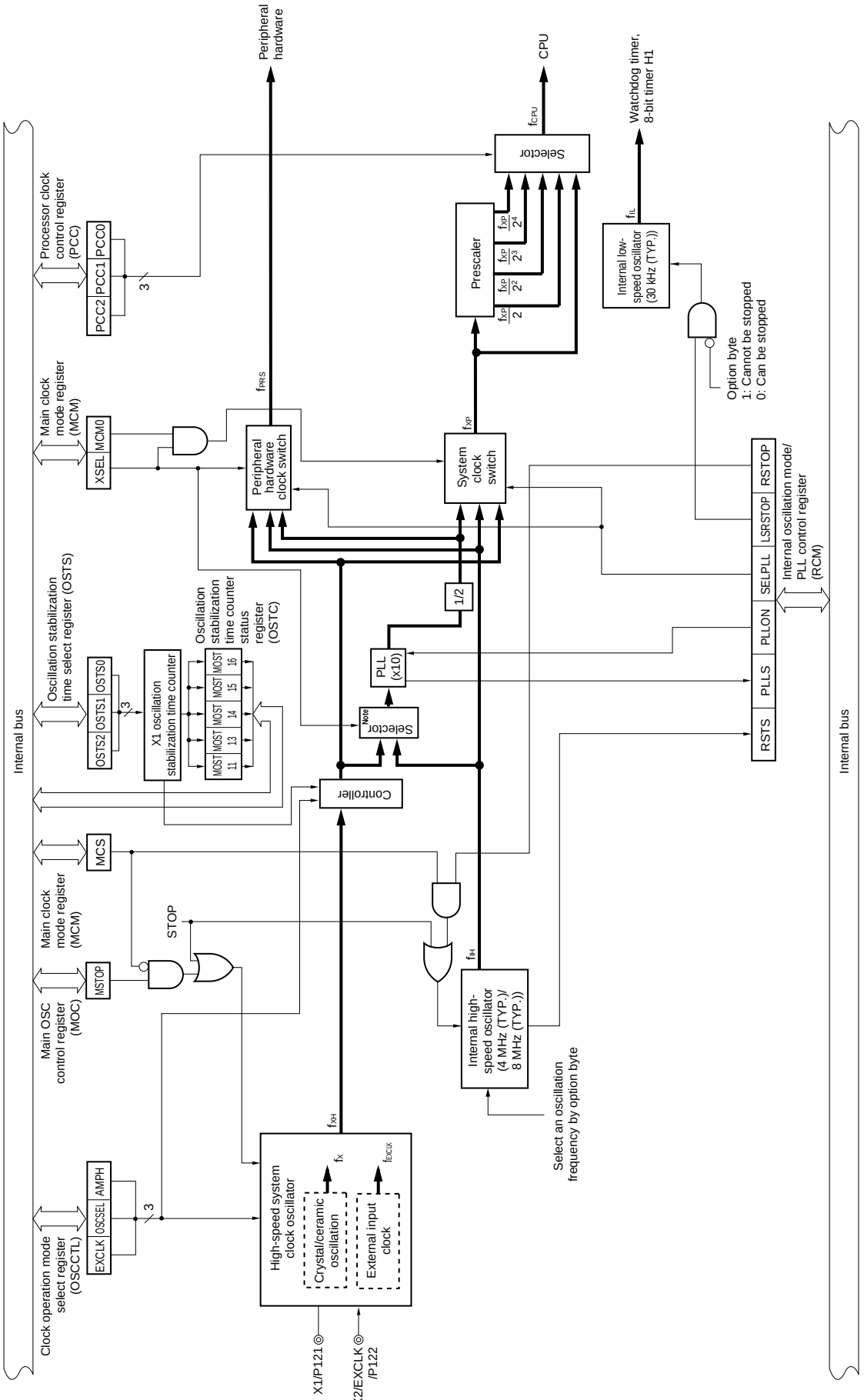
The register settings specify the clocks to be supplied as the main system clock and peripheral hardware clock as follows.

Table 5-2. Clocks Supplied to Main System Clock and Peripheral Hardware Clock

XSEL	MCM0	SELPLL	Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
0	0	0	Internal high-speed oscillation clock (f_{IH})	
	1			
0	0	1	10 times the internal high-speed oscillation clock (f_{IH}) × 1/2	
	1			
1	0	0	Internal high-speed oscillation clock (f_{IH})	High-speed system clock (f_{XH})
1	0	1	Setting prohibited	
1	1	0	High-speed system clock (f_{XH})	
1	1	1	10 times the high-speed system clock (f_{XH}) × 1/2	

Remark XSEL: Bit 2 of the main clock mode register (MCM)
MCM0: Bit 0 of MCM
SELPLL: Bit 3 of the internal oscillation mode/PLL control register (RCM)

Figure 5-1. Block Diagram of Clock Generator



Note Only 4 MHz can be used for the oscillation frequency.

Remark fx: X1 clock oscillation frequency

f_{IH}: Internal high-speed oscillation clock frequency
f_{EXCLK}: External main system clock frequency
f_{XH}: High-speed system clock frequency
f_{XP}: Main system clock frequency
f_{IL}: Internal low-speed oscillation clock frequency
f_{CPU}: CPU clock frequency
f_{PRS}: Peripheral hardware 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/PLL control 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.

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	0	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	Oscillation frequency control
0	2 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 5.00 to 19.35 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 5.00 to 19.35 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 5 to 0.**
7. **Do not set PLLON and AMPH to 1 at the same time.**
8. **AMPH can be set to 1 only once following a reset. When AMPH is 1, therefore, the STOP mode current is not a low current. When using the PLL, set PLLON to 0, enter STOP mode, and then set PLLON to 1. By doing this, a low current consumption of 0.3 μA can be guaranteed.**

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

(2) Processor clock control register (PCC)

This register is used to set the division ratio of the CPU 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

Symbol	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	PCC2	PCC1	PCC0

PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
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

Cautions 1. Be sure to clear bits 3 to 7 to 0.

2. The peripheral hardware clock (f_{PRS}) is not divided when the division ratio of the PCC is set.

Remark f_{XP}: Main system clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/Fx2-L microcontrollers. Therefore, the relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 5-3.

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

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$			
	Main System Clock (f_{XP})			
	High-Speed System Clock (f_{XH})Note 1		Internal High-Speed Oscillation Clock (f_{IH})Note 1	
	At 10 MHz OperationNote 2	At 20 MHz OperationNote 3	At 4 MHz (TYP.) OperationNote 4	At 20 MHz (TYP.) OperationNote 5
f_{XP}	0.2 μs	0.1 μs	0.5 μs (TYP.)	0.1 μs (TYP.)
$f_{XP}/2$	0.4 μs	0.2 μs	1.0 μs (TYP.)	0.2 μs (TYP.)
$f_{XP}/2^2$	0.8 μs	0.4 μs	2.0 μs (TYP.)	0.4 μs (TYP.)
$f_{XP}/2^3$	1.6 μs	0.8 μs	4.0 μs (TYP.)	0.8 μs (TYP.)
$f_{XP}/2^4$	3.2 μs	1.6 μs	8.0 μs (TYP.)	1.6 μs (TYP.)

Notes 1. 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) (refer to **Figure 5-6**).

2. When using clock-through mode (during $f_{XP} = f_{XH}$ = 10 MHz operation)

3. When using PLL mode (during operation at $f_{XP} = f_{XH} \times 5$, f_{XH} = 4 MHz)

4. When using clock-through mode (during $f_{XP} = f_{IH}$ = 4 MHz (TYP.) operation)

5. When using PLL mode (during operation at $f_{XP} = f_{IH} \times 5$, $f_{XP} = f_{IH}$ = 4 MHz (TYP.))

(3) Internal oscillation mode/PLL control register (RCM)

This register sets the operation mode of internal oscillator and controls the PLL function.

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/PLL control 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	PLLS	PLLON	SELPLL	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

PLLS	Status of PLL clock mode
0	Clock-through mode
1	PLL mode

PLLON	Control of PLL operation[Notes 3, 4]
0	Stops PLL operation
1	Enables PLL operation

SELPLL	PLL clock mode selection[Note 5]
0	Clock-through mode
1	PLL mode

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. Bits 7 and 5 are read-only.

3. A 10 μs wait occurs as an internal stabilization wait time after PLLON = 1 is set.

4. Only 4 MHz can be used for the PLL reference clock oscillation frequency.

5. The PLL clock mode is actually switched when the following time has elapsed after SELPLL was set.

- SELPLL 0 → 1: One clock of the clock before the mode was switched to PLL clock mode (MAX.)
- SELPLL 1 → 0: Three clocks of the clock before the mode was switched to PLL clock mode (MAX.)

Cautions 1. When setting RSTOP to 1, be sure to confirm that the CPU operates with a clock other (MCS = 1) than the internal high-speed oscillation clock. Specifically, set under either of the following conditions.
In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock before setting RSTOP to 1.
2. Do not set PLLON and AMPH to 1 at the same time.

Remark The source clock supplied to the peripheral hardware differs depending on the SELPLL setting.

SELPLL	Peripheral hardware
0	$f_{PRS} = f_{XP}$
1	$f_{PRS} = 10\ f_{XP} \times 1/2$ (20 MHz: f_{XP} = 4 MHz operation)

(4) 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. Clear MSTOP to 0 while the regulator mode control register (RMC) is 00H.
2. When setting MSTOP to 1, be sure to confirm that the CPU operates with a clock other (MCS = 0) than the high-speed system clock.
In addition, stop peripheral hardware that is operating on the high-speed system clock before setting MSTOP to 1.
3. Do not clear MSTOP to 0 while bit 6 (OSCSEL) of the clock operation mode select register (OSCCTL) is 0 (input port mode).
4. 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.

(5) 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_{IH})	Internal high-speed oscillation clock (f_{IH})
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. Setting XSEL and MCM0 to 1 and 0, respectively, is prohibited in PLL mode.

3. 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_{TMX}" is selected as the count clock for 16-bit timers X0 and X1 (operates with TMX control clock)**
- **When "f_{IL}", "$f_{IL}/2^6$", or "$f_{IL}/2^{15}$" 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))**

(6) 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 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.

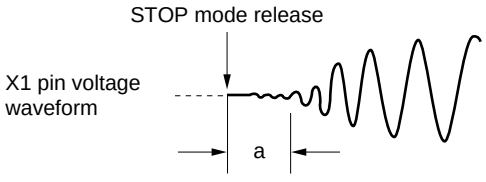
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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

(7) 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		
				f_X = 10 MHz	f_X = 20 MHz
0	0	1	$2^{11}/f_X$	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_X$	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_X$	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_X$	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_X$	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.

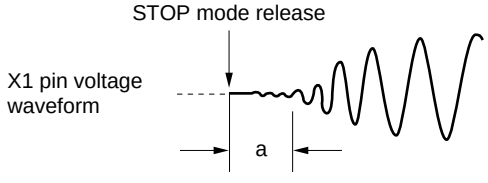
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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 f_X: X1 clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 X1 oscillator

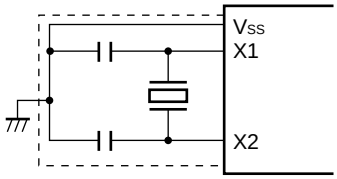
The X1 oscillator oscillates with a crystal resonator or ceramic resonator (clock-through mode: 2 to 20 MHz, PLL mode: 4 MHz) connected to the X1 and X2 pins.

An external clock can also be input. In this case, input the clock signal (clock-through mode: 2 to 20 MHz, PLL mode: 4 MHz) to the EXCLK pin.

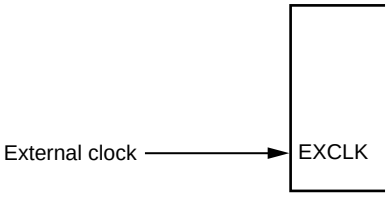
Figure 5-9 shows an example of the external circuit of the X1 oscillator.

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

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

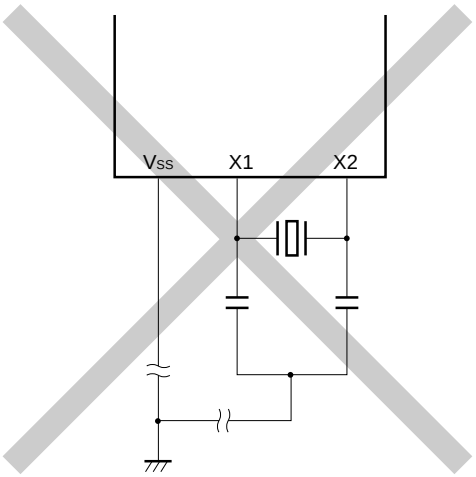
Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-9 to avoid an adverse effect from wiring capacitance.

- **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.**

Figure 5-10 shows examples of incorrect resonator connection.

Figure 5-10. Examples of Incorrect Resonator Connection (1/2)

(a) Too long wiring



(b) Crossed signal line

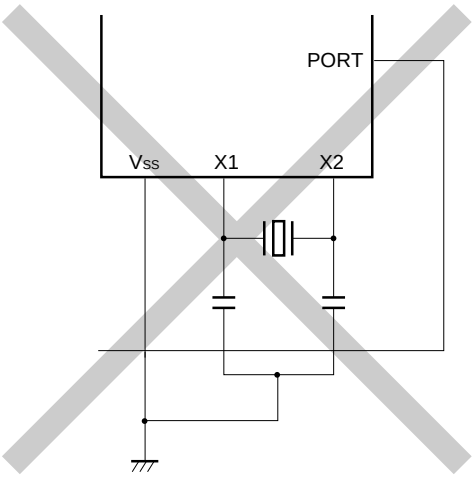
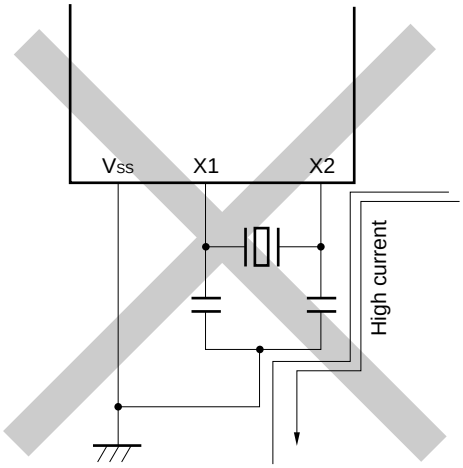
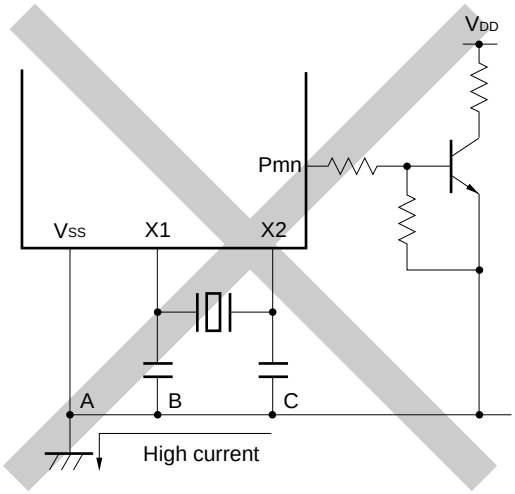


Figure 5-10. Examples of Incorrect Resonator Connection (2/2)

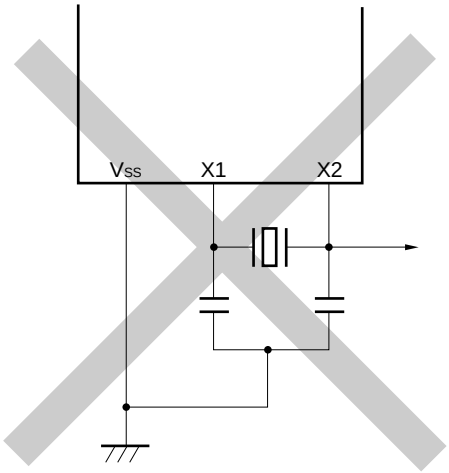
(c) Wiring near high alternating current



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



(e) Signals are fetched



5.4.2 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0/Fx2-L microcontrollers. Oscillation can be controlled by the internal oscillation mode/PLL control register (RCM).

After a reset release, the internal high-speed oscillator automatically starts oscillation.

Internal high-speed oscillation clock frequency (4 MHz (TYP.)/8 MHz (TYP.)) can be set by the option byte.

5.4.3 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0/Fx2-L microcontrollers.

The internal low-speed oscillation clock is only used as the watchdog timer and the clock of 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/PLL control register (RCM).

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

5.4.4 Prescaler

The prescaler generates the clock to be supplied to the CPU by dividing the main system clock.

5.4.5 PLL (Phase Locked Loop)

The PLL can be used to multiply the internal high-speed oscillation clock or high-speed system clock.

Set as follows to use or stop the PLL.

(a) When using PLL

- The PLL is set to be stopped (PLLON = 0) after reset is released. After reset is released, check the oscillation stability of the clock to be multiplied and then set the PLL to operate (PLLON = 1)[Note]. After the PLL starts operating, check that the PLL operation stabilization time (90 μs) has elapsed in clock-through mode (SELPLL = 0) and then change the mode to PLL mode (SELPLL = 1).

- To shift to STOP mode, execute the STOP instruction after making sure that the mode has been changed to clock-through mode (SELPLL = 0) and the PLL has been stopped (PLLON = 0). To return from STOP mode, start operating the PLL (PLLON = 1), check that the PLL operation stabilization time (90 μs) has elapsed, and then change the mode to PLL mode (SELPLL = 1).

(b) When stopping PLL

- Stop the PLL (PLLON = 0) after making sure that the mode has been switched to clock-through mode (SELPLL = 0). Do not simultaneously write 0 to the SELPLL and PLLON bits by using an 8-bit memory manipulation instruction.

Note X1 clock: Use the oscillation stabilization time counter status register (OSTC) to check the oscillation stabilization time.

Internal high-speed oscillator: Use bit 7 (RSTS) of the internal oscillation mode/PLL control register (RCM) to check the accuracy of the oscillation stabilization operation.

Cautions 1. Only 4 MHz can be used for the PLL reference clock oscillation frequency.

2. Do not set PLLON and AMPH to 1 at the same time.

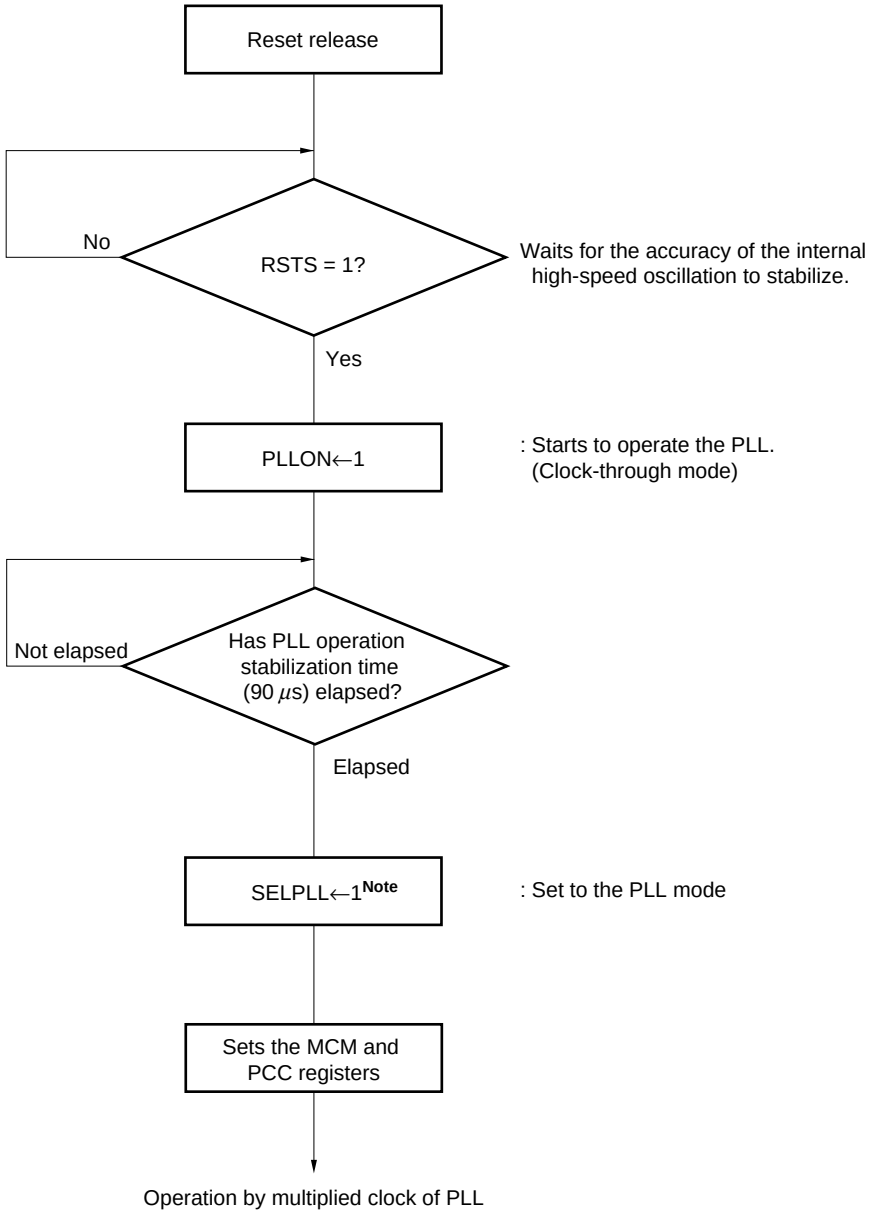
Remarks 1. A 10 μs wait occurs as an internal stabilization wait time after PLLON = 1 is set.

2. SELPLL: Bit 3 of the internal oscillation mode/PLL control register (RCM)

PLLON: Bit 4 of RCM

Here is an example when using PLL (flow chart).

Figure 5-11. Setting Example When Using PLL (Flow Chart)
(When Multiplying Internal High-Speed Oscillation Clock)



Note After starting to operate the PLL, check that the PLL operation stabilization time (90 μs) has elapsed and then set to PLL mode (SELPLL = 1).

Remark A 10 μs wait occurs as an internal stabilization wait time after PLLON = 1 is set.

5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (refer to **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_{IH}
- Internal low-speed oscillation clock f_{IL}
- 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/Fx2-L microcontrollers, thus enabling the following.

(1) Enhancement of security function

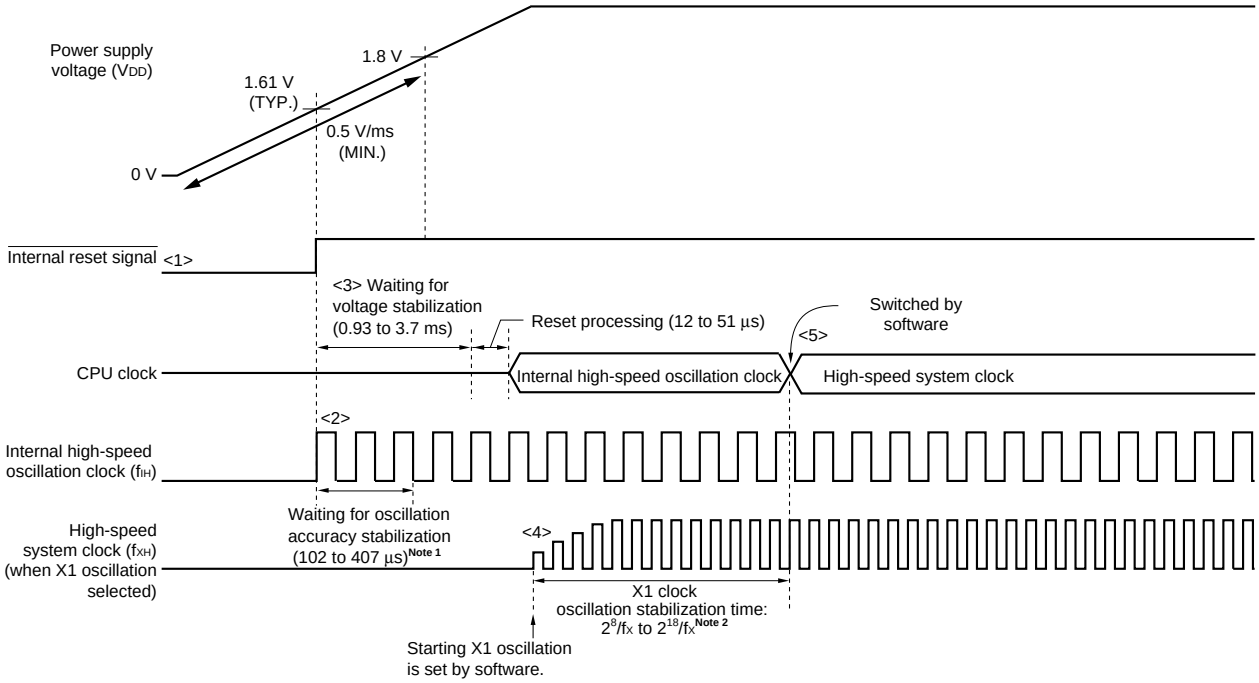
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. 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 Figures 5-12 and 5-13.

Figure 5-12. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVISTART = 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 1.61 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 clock via software (refer to **(1)** in **5.6.1 Example of controlling high-speed system clock**).

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then set switching via software (refer to **(3)** in **5.6.1 Example of controlling high-speed system 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 rise of the voltage to reach 1.8 V after turning on the power is more gradual than 0.5 V/ms (MIN.), perform one of the following operations.

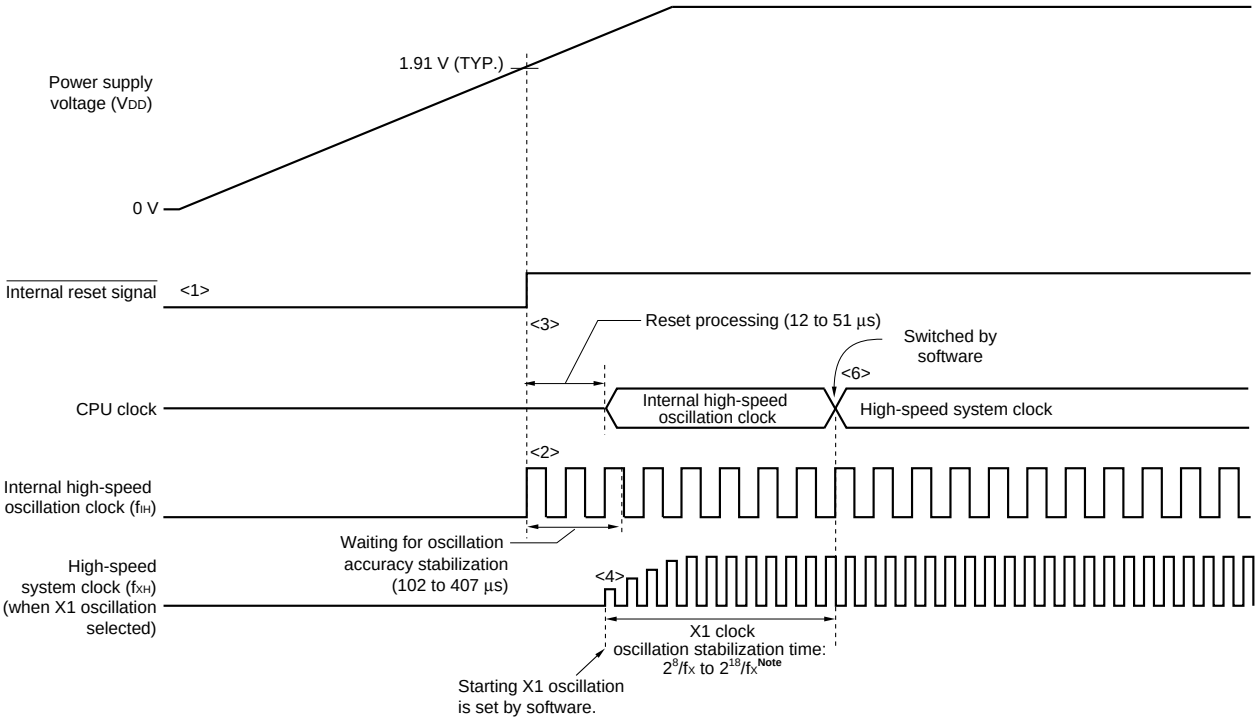
- **Input a low level to the $\overline{\text{RESET}}$ pin until 1.8 V is reached after turning on the power.**
- **Set the LVI default start function to operate (LVISTART = 1) using the option byte.**

When a low level has been input to the $\overline{\text{RESET}}$ pin until the voltage reaches 1.8 V, the CPU operates with the same timing as <2> and thereafter in Figure 5-12, after the reset has been released 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 pins 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 (refer to **(4)** in **5.6.1 Example of controlling high-speed system clock**, and **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**).

Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVISTART = 1))



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

<2> When the power supply voltage exceeds 1.91 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 clock via software (refer to **(1)** in **5.6.1 Example of controlling high-speed system clock**).

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then set switching via software (refer to **(3)** in **5.6.1 Example of controlling high-speed system 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 (0.93 to 3.7 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the supply voltage rises from 1.61 V (TYP.) to 1.91 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.

2. It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pins 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 (refer to **(4)** in **5.6.1 Example of controlling high-speed system clock**, and **(3)** in **5.6.2 Example of controlling internal high-speed oscillation 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 X1/P121 and X2/EXCLK/P122 pins can be used as input port pins.

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

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

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

Remark See **5.4.5 PLL (phase locked loop)** when using the PLL.

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

<1> 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	

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

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

Cautions 1. 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 (refer to CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) and CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)).

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

<1> 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	Input port	External clock input

<2> 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 (refer to CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) and CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)).

(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]

(Refer to **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> Selecting the CPU clock division ratio (PCC register)

To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
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 and stopping the X1 oscillation (disabling clock input if the external clock is used)
- 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, refer to **CHAPTER 18 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 (MCM registers)

Confirm with MCS that the CPU is operating on the internal high-speed oscillation clock.

When MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the internal high-speed oscillation 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 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

Remark See **5.4.5 PLL (phase locked loop)** when using the PLL.

(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]
(Refer to **5.6.2 (1) Example of setting procedure when restarting oscillation of the 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. Refer to **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_{IH})	Internal high-speed oscillation clock (f_{IH})
0	1		
1	0		High-speed system clock (f_{XH})

<3> Selecting the CPU clock division ratio (PCC register)

To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
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, refer to **CHAPTER 18 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. To operate the CPU immediately after the STOP mode has been released, set MCM0 to 0, switch the CPU clock to the internal high-speed oscillation clock, and check that RSTS is 1.

<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 (MCM registers)

Confirm with MCS that the CPU is operating on the high-speed system clock.

When MCS = 0, the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the high-speed system 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 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 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_{IL} 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 (30 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)

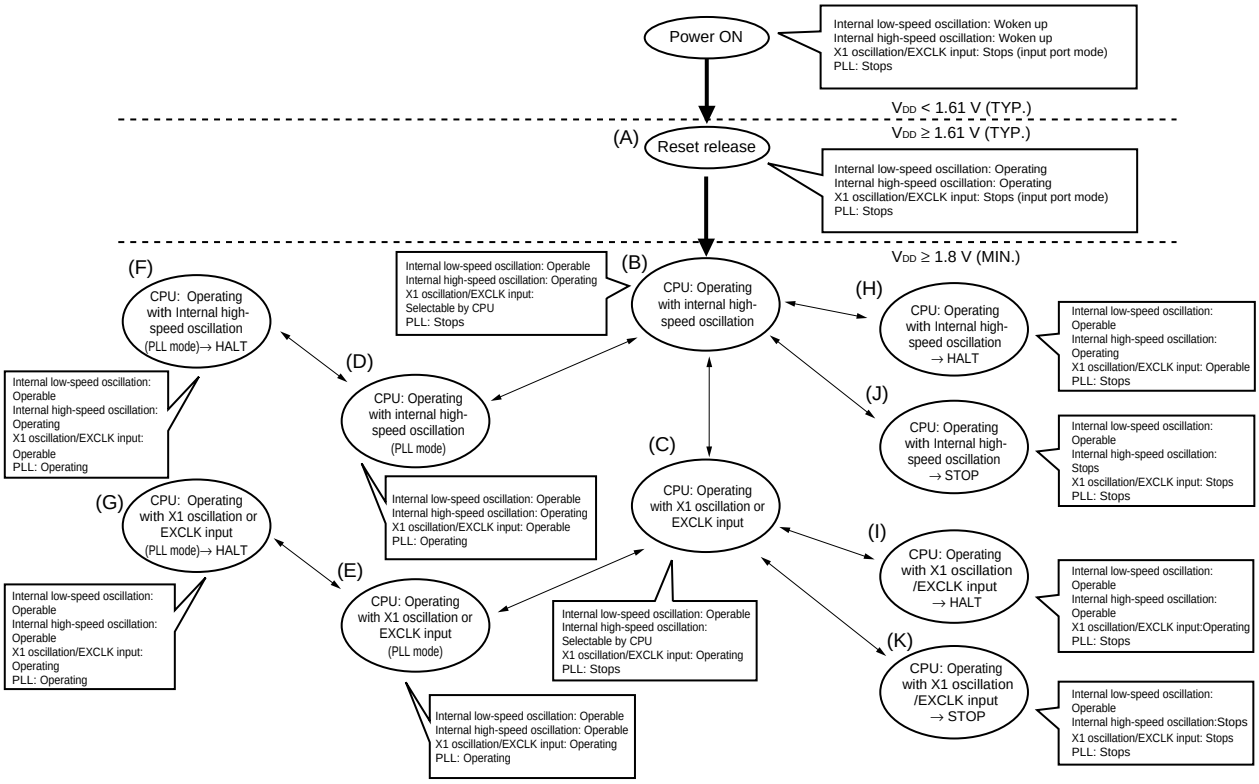
When LSRSTOP is cleared to 0, the internal low-speed oscillation clock is restarted.

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.

5.6.4 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 LVI Default Start Mode Function Stopped Is Set (Option Byte: LVISTART = 0))



Cautions 1. Be sure to stop the operation of the PLL before shifting to STOP mode.

2. When transitioning to the STOP mode, it is possible to achieve low power consumption by setting RMC = 56H.

3. Be sure to stop the operation of the PLL when switching the main system clock.

4. Only 4 MHz can be used for the PLL reference clock oscillation frequency.

Remark When LVI default start function enabled is set (option byte: LVISTART = 1), the CPU clock status changes to (A) in the above figure when the supply voltage exceeds 1.91 V (TYP.), and to (B) after reset processing.

Table 5-4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (1/3)

(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 (B) immediately after a reset release.)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(A) → (B) → (C) (X1 clock)	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main system clock)	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 (refer to CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) and CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)).

(3) 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	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0
(B) → (C) (X1 clock)	0	1	0	Must be checked	1	1
(B) → (C) (external main system clock)	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 (refer to CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) and CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)).

Remarks 1. (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-14.

2. EXCLK, OSCSEL: 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)

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (2/3)

(4) 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

(5) CPU clock changing from internal high-speed oscillation clock (B) to internal high-speed oscillation clock (PLL mode) (D)

CPU clock changing from high-speed system clock (C) to high-speed system clock (PLL mode) (E)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	SELPLL	PLLON	Waiting for Oscillation Stabilization	SELPLL
(B) → (D) (C) → (E)	0	1	Necessary (90 μs)	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the PLL

(6) CPU clock changing from internal high-speed oscillation clock (PLL mode) (D) to internal high-speed oscillation clock (B)

CPU clock changing from high-speed system clock (PLL mode) (E) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	SELPLL	PLLON
(D) → (B) (E) → (C)	0	0

Remarks 1. (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-14.

2. A 10 μs wait occurs as an internal stabilization wait time after PLLON = 1 is set.

3. RSTS, PLLON, SELPLL, RSTOP: Bits 7,4 ,3, and 0 of the internal oscillation mode register (RCM)
MCM0: Bit 0 of the main clock mode register (MCM)

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (3/3)

(7) • HALT mode (H) set while CPU is operating with internal high-speed oscillation clock (B)
• HALT mode (I) set while CPU is operating with high-speed system clock (C)
• HALT mode (F) set while CPU is operating with internal high-speed oscillation clock (PLL mode) (D)
• HALT mode (G) set while CPU is operating with high-speed system clock (PLL mode) (E)

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

(8) • STOP mode (J) set while CPU is operating with internal high-speed oscillation clock (B)
• STOP mode (K) set while CPU is operating with high-speed system clock (C)

(Setting sequence) ——————————▶

Status Transition	Setting	
(B) → (J) (C) → (K)	Stopping peripheral functions that cannot operate in STOP mode	Executing STOP instruction

Caution When transitioning to the STOP mode, it is possible to achieve low power consumption by setting RMC = 56H.

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-14.

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

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

Table 5-5. 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).
	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).
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).

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

By setting bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC), 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 (refer to **Table 5-6**).

Table 5-6. Time Required for Switchover of CPU Clock and Main System Clock Cycle Division Factor

Set Value Before Switchover			Set Value After Switchover														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 clocks			16 clocks			16 clocks			16 clocks		
0	0	1	8 clocks						8 clocks			8 clocks			8 clocks		
0	1	0	4 clocks			4 clocks						4 clocks			4 clocks		
0	1	1	2 clocks			2 clocks			2 clocks						2 clocks		
1	0	0	1 clock			1 clock			1 clock			1 clock					

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

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 (refer to **Table 5-7**).

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.

Caution Be sure to stop the operation of the PLL when switching the main system clock.

Table 5-7. Maximum Time Required for Main System Clock Switchover

Set Value Before Switchover	Set Value After Switchover	
MCM0	MCM0	
	0	1
0		$1 + 2f_{IH}/f_{XH}$ clock
1	$1 + 2f_{XH}/f_{IH}$ 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 5-7 is the number of main system clocks before switchover.

2. Calculate the number of clocks in Table 5-7 by removing the decimal portion.

Example When switching the main system clock from the internal high-speed oscillation clock to the high-speed system clock (@ oscillation with f_{IH} = 4 MHz, f_{XH} = 10 MHz)

$$1 + 2f_{IH}/f_{XH} = 1 + 2 \times 4/10 = 1 + 2 \times 0.4 = 1 + 0.8 = 1.8 \rightarrow 1 \text{ clock}$$

5.6.7 Conditions before clock oscillation is stopped

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

Table 5-8. Conditions Before 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 (The CPU is operating on the high-speed system clock)	RSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on the internal high-speed oscillation clock)	MSTOP = 1
External main system clock		

5.6.8 Peripheral hardware and source clocks

The following lists peripheral hardware and source clocks incorporated in the 78K0/Fx2-L microcontrollers.

Remark The peripheral hardware depends on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

Table 5-9. Peripheral Hardware and Source Clocks

Peripheral Hardware \ Source Clock		Peripheral Hardware Clock (f_{PRS})	Internal Low-Speed Oscillation Clock (f_{IL})	External Clock from Peripheral Hardware Pins
16-bit timers X0 and X1		Y	N	N
16-bit timer/event counter 00		Y	N	Y (TI000 pin)[Note]
8-bit timer/event counter 51		Y	N	Y (TI51 pin)[Note]
8-bit timer H1		Y	Y	N
Watchdog timer		N	Y	N
A/D converter		Y	N	N
Serial interface	UART6	Y	N	N
	CSI11	Y	N	Y ($\overline{SCK11}$ pin)[Note]
	IICA	Y	N	Y (SCLA0 pin)[Note]

Note Do not start the peripheral hardware operation with the external clock from peripheral hardware pins when in the STOP mode.

Remark Y: Can be selected, N: Cannot be selected

CHAPTER 6 16-BIT TIMERS X0 AND X1

	78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
16-bit timer X0	**Note**	Mounted	
16-bit timer X1	Not mounted		Mounted

Note Only the interval timer function is provided.

6.1 Functions of 16-bit Timers X0 and X1

16-bit timers X0 and X1 are mounted onto all 78K0/Fx2-L microcontroller products.

16-bit timers X0 and X1 are dedicated PWM output timers and have two outputs each, enabling the generation of up to four PWM outputs. Complementary PWM output can also be generated to control a half-bridge circuit (2 outputs) or full-bridge circuit (4 outputs). Also, by linking with a comparator or INTP0, PFC control and PWM output can be stopped urgently.

16-bit timer X0 is provided with the following functions.

(1) Interval timer

16-bit timers X0 and X1 generate an interrupt request at the preset time interval.

(2) A/D conversion start timing signal output

The A/D conversion start timing signal can be output by using a compare register (TXnCCR0 register).

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(3) Capture function

This function captures the count value to the capture register by detecting a comparator output or an external interrupt input (INPT0).

(4) PWM output[Note]

- A variable pulse with any duty or cycle can be output while the timer is operating.
- The default timer output level (high or low level) can be set.

Note 78K0/FA2-L and 78K0/FB2-L only.

(5) Timer start synchronization function[Note]

Up to 4 PWM outputs can be simultaneously started by combining two timer units (16-bit timers X0 and X1).

Note 78K0/FB2-L only.

(6) Timer start/clear synchronization function[Note]

Up to 4 PWM output cycles can be synchronized by combining two timer units (16-bit timers X0 and X1).

Note 78K0/FB2-L only.

(7) Timer output gating function (by interlocking with 8-bit timer H1)[Note]

Timer output can be gate-controlled by using the output of 8-bit timer H1 (the TOH1 output).

Note 78K0/FA2-L and 78K0/FB2-L only.

(8) Timer reset mode (comparator, INTP0 interlocking mode 1)[Note]

Timer output can be reset and the timer counter cleared while the comparator 0 to 2 outputs or the INTP0 input is high level. When the comparator 0 to 2 outputs or the INTP0 input go to low level, timer output restarts.

Note 78K0/FA2-L and 78K0/FB2-L only.

(9) Timer restart mode (comparator, INTP0 interlocking mode 2)[Note]

Timer can be restarted upon detection of the rising edge of the comparator 0 to 2 outputs or the INTP0 input.

Note 78K0/FA2-L and 78K0/FB2-L only.

(10) Timer output reset mode (comparator, INTP0 interlocking mode 3)[Note]

Timer output can be reset upon detection of the rising edge of the comparator 0 to 2 outputs or the INTP0 input. The reset status is cleared when the next timer interrupt occurs and timer output restarts.

Note 78K0/FA2-L and 78K0/FB2-L only.

(11) High-impedance output control function (by interlocking with comparator and INTP0)[Note]

Timer output can be made high impedance upon detection of the valid edge of the comparator 0 to 2 outputs or the INTP0 input.

Note 78K0/FA2-L and 78K0/FB2-L only.

6.2 Configuration of 16-bit Timers X0 and X1

16-bit timers X0 and X1 include the following hardware.

Table 6-1. Configuration of 16-bit Timers X0 and X1

(1) 16-bit timer X0

Item	Configuration
Timer/counter	16-bit timer counter X0
Register	16-bit timer X0 capture/compare register 0 (TX0CCR0) 16-bit timer X0 compare registers 0 to 3 (TX0CR0 to TX0CR3)
Timer output	TOX00[Note], TOX01[Note]
Control registers	16-bit timer X0 operation control registers 0 to 3 (TX0CTL0 to TX0CTL3) 16-bit timer X0 operation control register 4 (TX0CTL4)[Note] 16-bit timer X0 output control register 0 (TX0IOC0)[Note] Port mode register 3 (PM3) Port register 3 (P3)

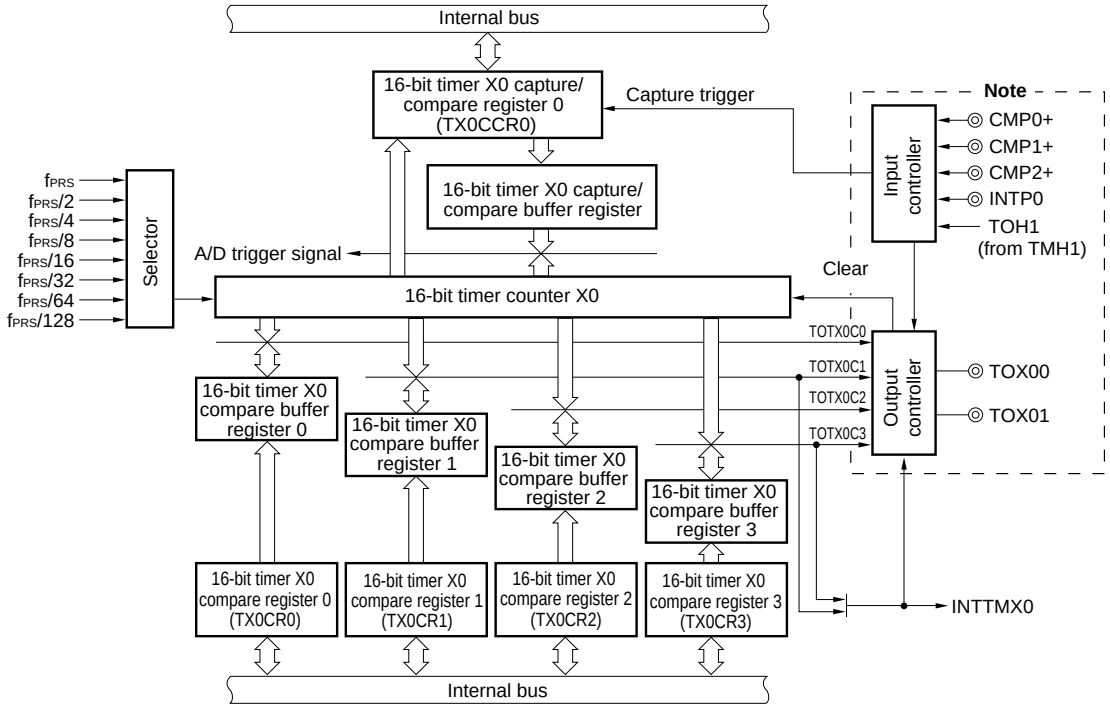
Note 78K0/FA2-L and 78K0/FB2-L only

(2) 16-bit timer X1 (78K0/FB2-L only)

Item	Configuration
Timer/counter	16-bit timer counter X1
Register	16-bit timer X1 capture/compare register 0 (TX1CCR0) 16-bit timer X1 compare registers 0 to 3 (TX1CR0 to TX1CR3)
Timer output	TOX10, TOX11
Control registers	16-bit timer X1 operation control registers 0 to 2, 4 (TX1CTL0 to TX1CTL2, TX1CTL4) 16-bit timer X1 output control register 0 (TX1IOC0) Port mode register 3 (PM3) Port register 3 (P3)

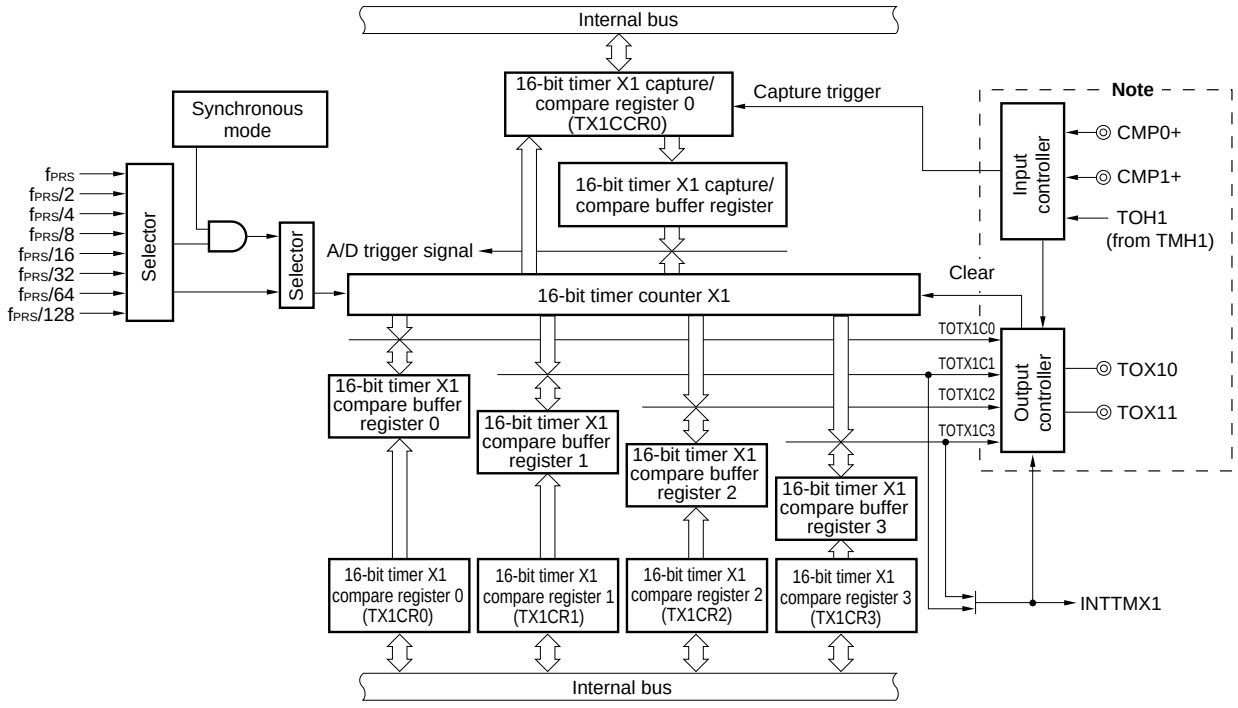
Figures 6-1 to 6-3 show the block diagrams.

Figure 6-1. Block Diagram of 16-bit Timer X0



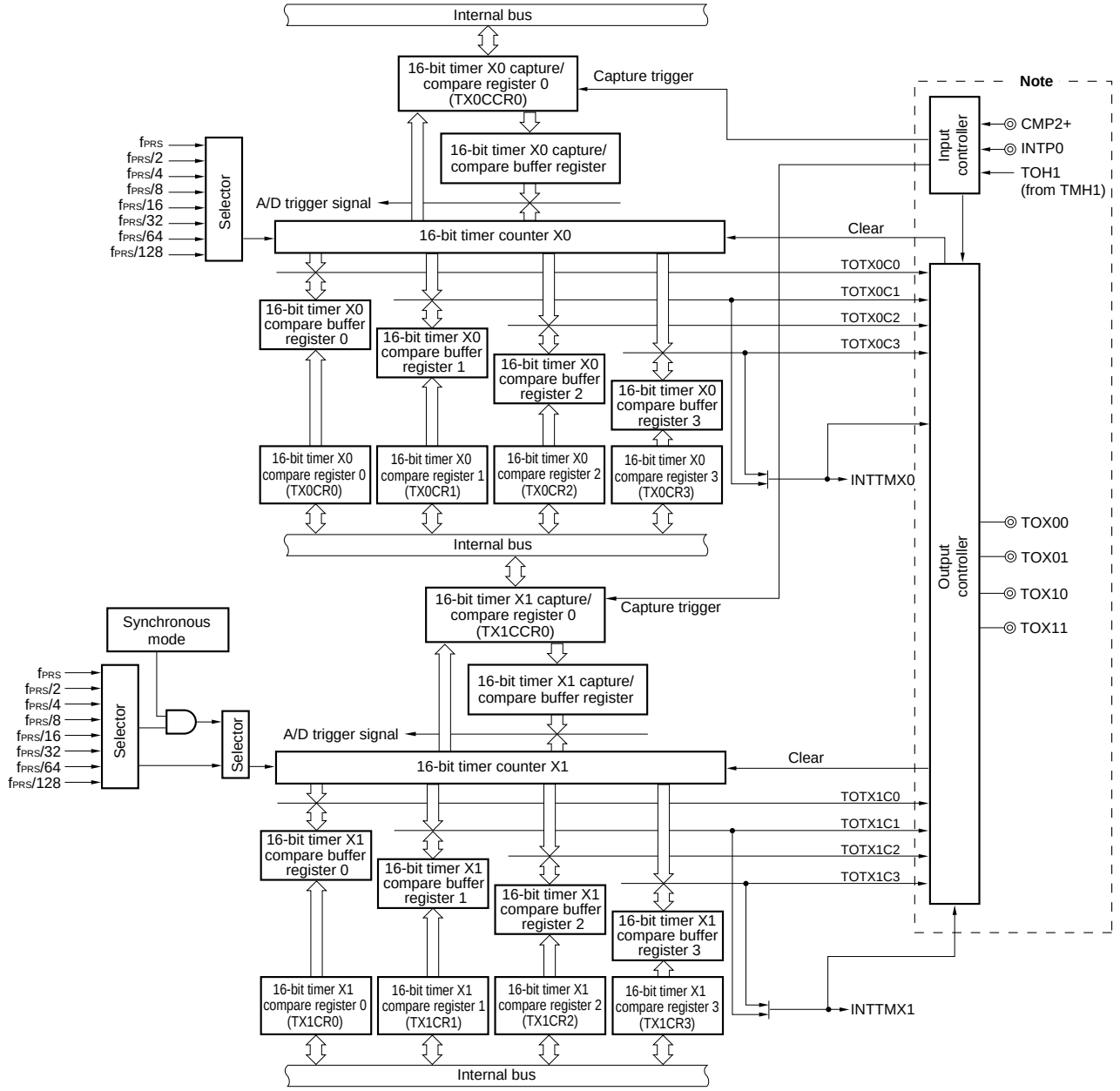
Note For details, see **Figure 6-43 lock Diagram of 16-bit Timer X0 Output Configuration**.

Figure 6-2. Block Diagram of 16-bit Timer X1



Note For details, see **Figure 6-44 Block Diagram of 16-bit Timer X1 Output Configuration**.

Figure 6-3. Block Diagram of 16-bit Timers X0 and X1
(TMX0 and TMX1 Synchronous Start/Clear Mode, TMX0 and TMX1 Synchronous Start Mode)



Note For details, see **Figure 6-45 Block Diagram of 16-bit Timers X0 and X1 Output Configuration**.

(1) 16-bit timer Xn capture/compare register 0 (TXnCCR0)

This is a 16-bit register that can switch between being used for the capture function and the compare function. TXnCTL2 is used to switch between the capture function and compare function.

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

Reset signal generation clears this register to 0000H.

Figure 6-4. Format of 16-bit Timer Xn Capture/Compare Register 0 (TXnCCR0)

Address: FF92H, FF93H (TX0CCR0), FFB6H, FFB7H (TX1CCR0) After reset: 0000H R/W

	FF93H (TX0CCR0), FFB7H (TX1CCR0)								FF92H (TX0CCR0), FFB6H (TX1CCR0)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXnCCR0																

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(i) Using TXnCCR0 as a compare register

TXnCCR0 can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TXnTMC = 1). When the value of TXnCCR0 is rewritten while the timer is operating, that value is latched, transferred to TXnCCR0 at the following timing, and the value of TXnCCR0 is changed.

- The counter value and TXnCR1 setting value match (TXnPWM = 0)
- The counter value and TXnCR3 setting value match (TXnPWM = 1)

In interlocking mode 2, the latched value is transferred to TXnCCR0 and the value of TXnCCR0 is changed at the timing of the comparator output by setting the TX0CMPLDSET1 and TX0CMPLDSET0 bits.

(ii) Using TXnCCR0 as a capture register

The count value is captured to TXnCCR0 by inputting a capture trigger.

Caution When reading TXnCCR0 continuously, wait for at least 3 clock cycles of the 16-bit timer Xn count clock between reads.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(2) 16-bit timer Xn compare register m (TXnCRm)

TXnCRm can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TXnTMC = 1). When the value of TXnCRm is rewritten while the timer is operating, that value is latched, transferred to TXnCRm at the following timing, and the value of TXnCRm is changed.

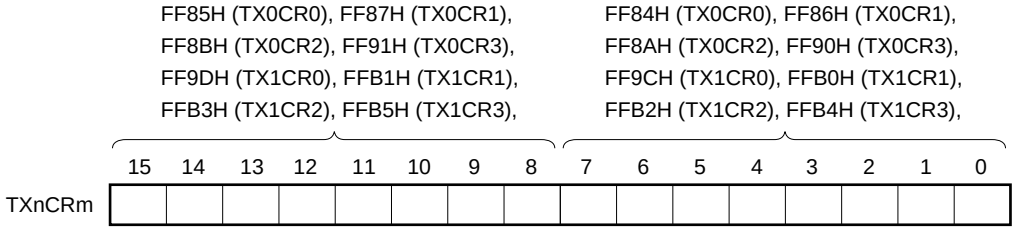
- The counter value and TXnCR1 setting value match (TXnPWM = 0)
- The counter value and TXnCR3 setting value match (TXnPWM = 1)

In interlocking mode 2, the latched value is transferred to TXnCRm and the value of TXnCRm is changed at the timing of the comparator output by setting the TX0CMPLDSET1 and TX0CMPLDSET0 bits.
This register can be read or written in 16-bit units.
Reset signal generation clears this register to 0000H.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-5. Format of 16-bit Timer Xn Compare Register m (TXnCRm)

Address: FF84H, FF85H (TX0CR0), FF86H, FF87H (TX0CR1), FF8AH, FF8BH (TX0CR2), FF90H, FF91H (TX0CR3), FF9CH, FF9DH (TX1CR0), FFB0H, FFB1H (TX1CR1), FFB2H, FFB3H (TX1CR2), FFB4H, FFB5H (TX1CR3)
After reset: 0000H R/W



Remark m = 0 to 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

6.3 Registers Controlling 16-bit Timers X0 and X1

Registers used to control 16-bit timers X0 and X1 are shown below.

- 16-bit timer X0
  - 16-bit timer X0 operation control registers 0 to 3 (TX0CTL0 to TX0CTL3)
  - 16-bit timer X0 operation control register 4 (TX0CTL4)Note
  - 16-bit timer X0 output control register 0 (TX0IOC0)Note
  - Port mode register 3 (PM3)
  - Port register 3 (P3)

Note 78K0/FA2-L and 78K0/FB2-L only

- 16-bit timer X1 (78K0/FB2-L only)
  - 16-bit timer X1 operation control registers 0 to 2, 4 (TX1CTL0 to TX1CTL2, TX1CTL4)
  - 16-bit timer X1 output control register 0 (TX1IOC0)
  - Port mode register 3 (PM3)
  - Port register 3 (P3)

(1) 16-bit timer Xn operation control register 0 (TXnCTL0)

TXnCTL0 is a register that controls the count operation and sets the count clock of 16-bit timer Xn.

TXnCTL0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TXnCTL0 to 00H.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-6. Format of 16-bit Timer X0 Operation Control Register 0 (TX0CTL0)

Address: FF7EH After reset: 00H R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
TX0CTL0	TX0TMC	0	0	0	0	TX0CKS2	TX0CKS1	TX0CKS0

TX0TMC	TMX0 count operation control
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation

SELPLL	TX0CKS2	TX0CKS1	TX0CKS0	TMX0 count clock selection		
					f_{PRS} = 4 MHz	f_{PRS} = 20 MHz (when using PLL)
0	0	0	0	f_{PRS}	4 MHz	–
0	0	0	1	$f_{PRS}/2$	2 MHz	–
0	0	1	0	$f_{PRS}/2^2$	1 MHz	–
0	0	1	1	$f_{PRS}/2^3$	500 kHz	–
0	1	0	0	$f_{PRS}/2^4$	250 kHz	–
0	1	0	1	$f_{PRS}/2^5$	125 kHz	–
0	1	1	0	$f_{PRS}/2^6$	62.5 kHz	–
0	1	1	1	$f_{PRS}/2^7$	31.25 kHz	–
1	0	0	1	f_{PRS}	–	20 MHz
Other than above				Setting prohibited		

Cautions 1. Only 4 MHz can be used for the PLL reference clock oscillation frequency.

2. When rewriting TX0CKS2 to TX0CKS0 bits to other data, stop the timer operation beforehand (TX0TMC = 0).

Remark SELPLL: Bit 3 of the internal oscillation mode/PLL control register (RCM)

f_{PRS}: Peripheral hardware clock frequency

f_{XP}: Main system clock frequency

Figure 6-7. Format of 16-bit Timer X1 Operation Control Register 0 (TX1CTL0) (78K0/FB2-L Only)

Address: FF94H After reset: 00H R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
TX1CTL0	TX1TMC	0	0	0	0	TX1CKS2	TX1CKS1	TX1CKS0

TX1TMC	TMX1 count operation control
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation

TX1 MD1	TX1 MD0	SEL PLL	TX1 CKS2	TX1 CKS1	TX1 CKS0	TMX1 count clock selection		
							f_{PRS} = 4 MHz	f_{PRS} = 20 MHz (when using PLL)
0	0/1	0	0	0	0	f_{PRS}	4 MHz	–
		0	0	0	1	$f_{PRS}/2$	2 MHz	–
		0	0	1	0	$f_{PRS}/2^2$	1 MHz	–
		0	0	1	1	$f_{PRS}/2^3$	500 kHz	–
		0	1	0	0	$f_{PRS}/2^4$	250 kHz	–
		0	1	0	1	$f_{PRS}/2^5$	125 kHz	–
		0	1	1	0	$f_{PRS}/2^6$	62.5 kHz	–
		0	1	1	1	$f_{PRS}/2^7$	31.25 kHz	–
		1	0	0	1	f_{PRS}	–	20 MHz
1	0	x	x	x	x	TMX0 count clock (see **Figure 6-6 Format of 16-bit Timer X0 Operation Control Register 0 (TX0CTL0)**)		
Other than above						Setting prohibited		

Cautions 1. Only 4 MHz can be used for the PLL reference clock oscillation frequency.

2. When rewriting TX1CKS2 to TX1CKS0 bits to other data, stop the timer operation beforehand (TX1TMC = 0).

Remark SELPLL: Bit 3 of the internal oscillation mode/PLL control register (RCM)

TX1MD1, TX1MD0: Bits 1 and 0 of 16-bit timer X1 operation control register 1

f_{PRS}: Peripheral hardware clock frequency

f_{XP}: Main system clock frequency

(2) 16-bit timer Xn operation control register 1 (TXnCTL1)

TXnCTL1 is a register that sets timer start via detection of INTP0 rising edge, output gate function by TOH1 output, the PWM output operation, and synchronous operation mode.

TXnCTL1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TXnCTL1 to 00H.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-8. Format of 16-bit Timer X0 Operation Control Register 1 (TX0CTL1)

Address: FF7FH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	0
TX0CTL1	TX0INTPST	0	TX0PWM CE Note 3	TX0PWM CINV Note 3	TX0PWM Note 3	0	0	0

TX0INTPST	Control of timer start operation via detection of INTP0 rising edge
0	Disables timer start operation via detection of INTP0 rising edge (starts timer via setting (1) of TX0TMC) Note 1.
1	Enables timer start operation via detection of INTP0 rising edge Note 2.

TX0PWM CE Note 3	Control of TOX0n output gate function by TOH1 output (n = 0, 1)
0	Does not use output gate function.
1	Use output gate function.

TX0PWM CINV Note 3	Setting of TOXmn output by TOH1 output (mn = 00, 01, 10, 11)
0	• Performs PWM output from TOXmn while the TOH1 output is high level. • Outputs a default level of TOXmn while the TOH1 output is low level.
1	• Performs PWM output from TOXmn while the TOH1 output is low level. • Outputs a level that is the inverse of the default level of TOXmn while the TOH1 output is high level.

TX0PWM Note 3	TMX0 PWM output operation setting
0	• Single output (TOX00 pin only) • INTTMX0 is generated upon match of counter and TX0CR1 register
1	• Dual output (TOX00 and TOX01 pins) • INTTMX0 is generated upon match of counter and TX0CR3 register

Notes 1. In TMX0 or TMX1 synchronous start mode (available only in 78K0/FB2-L), a timer start operation via detection of INTP0 rising edge cannot be performed, so set TX0INTPST to 0.

2. If 1 is set to TX0TMC after setting 1 to TX0INTPST, detection of INTP0 rising edge will be waited for. If the INTP0 rising edge is detected, 16-bit timer X0 will start counting up.

3. 78K0/FA2-L and 78K0/FB2-L only.

(**Caution** is listed on next page.)

Cautions 1. During timer operation, setting the other bits of TX0CTL1 is prohibited. However, TX0CTL1 can be refreshed (the same value is written).

2. Be sure to clear bits 0 to 2 and 6 to 0.

3. When using the output gate function, set bit 0 (TOEN1) of the TMHMD1 register to 1 (enable the TOH1 output).

Figure 6-9. Format of 16-bit Timer X1 Operation Control Register 1 (TX1CTL1) (78K0/FB2-L Only)

Address: FF95H After reset: 00H R/W

Symbol	7	6	<5>	4	<3>	2	<1>	<0>
TX1CTL1	0	0	TX1PWM CE	0	TX1PWM	0	TX1MD1	TX1MD0

TX1PWM CE	Control of TOX1n output gate function by TOH1 output (n = 0, 1)
0	Does not use output gate function.
1	Use output gate function.

TX1PWM	TMX1 PWM output operation setting
0	• Single output (TOX10 pin only) • INTTMX1 is generated upon match of counter and TX1CR1 register
1	• Dual output (TOX10 and TOX11 pins) • INTTMX1 is generated upon match of counter and TX1CR3 register

TX1MD1	TX1MD0	Operation mode setting
0	0	TMX1-only start mode
0	1	TMX0 and TMX1 synchronous start mode
1	0	TMX0 and TMX1 synchronous start/clear mode
1	1	Setting prohibited

Cautions 1. During the timer operation, setting the other bits of TX1CTL1 is prohibited. However, TX1CTL1 can be refreshed (the same value is written).

2. Be sure to clear bits 2, 4, 6 and 7 to 0.

3. When using the output gate function, set bit 0 (TOEN1) of the TMHMD1 register to 1 (enable the TOH1 output).

(3) 16-bit timer Xn operation control register 2 (TXnCTL2)

TXnCTL2 is a register that selects the capture trigger source, controls the generation of the A/D conversion synchronization trigger, and sets the TXnCCR0 register.

TXnCTL2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TXnCTL2 to 00H.

Remarks 1. The capture trigger source differs as follows, according to the operation mode.

Operation mode		Capture trigger sources
TMX0-only start mode		INTCMP2, INTP0
TMX1-only start mode (78K0/FB2-L only)		INTCMP1
TMX0 and TMX1 synchronous start mode (78K0/FB2-L only)	TMX0	INTCMP2, INTP0
	TMX1	INTCMP1
TMX0 and TMX1 synchronous start/clear mode (78K0/FB2-L only)	TMX0	INTCMP2, INTP0
	TMX1	INTCMP1

2. n = 0, 1

Figure 6-10. Format of 16-bit Timer X0 Operation Control Register 2 (TX0CTL2)

Address: FF80H After reset: 00H R/W

Symbol	7	<6>	5	4	3	2	<1>	<0>
TX0CTL2	0	TX0TRGS	0	0	0	0	TX0ADEN	TX0CCS

TX0TRGS	TMX0 capture trigger source selection
0	INTCMP2
1	INTP0

TX0ADEN	Control of generating A/D conversion synchronization trigger from TMX0
0	Disables generating A/D conversion synchronization trigger
1	Enables generating A/D conversion synchronization trigger[Note]

TX0CCS	TX0CCR0 register control
0	Operates as compare register[Note]
1	Operates as capture register

Note When enabling generation of the A/D conversion synchronization trigger (TX0ADEN = 1), set the TX0CCR0 register to operate as a compare register (TX0CCS = 0), because the A/D conversion synchronization trigger is generated upon a match between the counter and the TX0CCR0 register.

Cautions 1. During the 16-bit timer X0 operation, setting the other bits of TX0CTL2 is prohibited. However, TX0CTL2 can be refreshed (the same value is written).

2. The registers used by the A/D converter (ADM0, ADPC0, ADPC1, ADS) can be rewritten while the 16-bit timer X0 is operating.

3. A/D conversion synchronization triggers that occur while A/D conversion is stopped (ADCS = 0) are invalid. A/D conversion synchronization triggers that occur after A/D conversion has been enabled (ADCS = 1) are valid.

Figure 6-11. Format of 16-bit Timer X1 Operation Control Register 2 (TX1CTL2) (78K0/FB2-L only)

Address: FF96H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TX1CTL2	0	0	0	0	0	0	TX1ADEN	TX1CCS

TX1ADEN	Control of generating A/D conversion synchronization trigger from TMX1
0	Disables generating A/D conversion synchronization trigger
1	Enables generating A/D conversion synchronization trigger[Note]

TX1CCS	TX1CCR0 register operation
0	Operates as compare register[Note]
1	Operates as capture register

Note When enabling generation of the A/D conversion synchronization trigger (TX1ADEN = 1), set the TX1CCR0 register to operate as a compare register (TX1CCS = 0), because the A/D conversion synchronization trigger is generated upon a match between the counter and the TX1CCR0 register.

Cautions 1. During the 16-bit timer operation, setting the other bits of TX1CTL2 is prohibited. However, TX1CTL2 can be refreshed (the same value is written).

2. The registers used by the A/D converter (ADM0, ADPC0, ADPC1, ADS) can be rewritten while the 16-bit timer X1 is operating.

3. A/D conversion synchronization triggers that occur while A/D conversion is stopped (ADCS = 0) are invalid. A/D conversion synchronization triggers that occur after A/D conversion has been enabled (ADCS = 1) are valid.

(4) 16-bit timer X0 operation control register 3 (TX0CTL3)

TX0CTL3 is a register that sets the mode of the interlocking function with comparator 2 and INTP0, and sets the operation when restarting upon comparator output.

TX0CTL3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TX0CTL3 to 00H.

Figure 6-12. Format of 16-bit Timer X0 Operation Control Register 3 (TX0CTL3)

Address: FF81H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	<1>	<0>
TX0CTL3	0	TX0CMPLD SET1[Note 5]	TX0CMPLD SET0[Note 5]	0	TX0INTP0R M1	TX0INTP0R M0	TX0CMP2R M1	TX0CMP2R M0

TX0CMPLD SET1[Note 5]	TX0CMPLD SET0[Note 5]	Change of compare register when restarting upon comparator output (When interlocking mode 2 (timer restart mode) is set)
0	0	Does not change compare register value when restarting.
0	1	Rewrites values of all compare registers (TXnCR0 to TXnCR3, TXnCCR0)[Note 1] at once when restarting upon comparator 0 output[Notes 2, 3].
1	0	Rewrites values of all compare registers (TXnCR0 to TXnCR3, TXnCCR0)[Note 1] at once when restarting upon comparator 1 output[Notes 2, 3].
1	1	Rewrites values of all compare registers (TXnCR0 to TXnCR3, TXnCCR0)[Note 1] at once when restarting upon comparator 2 output[Notes 2, 3].

TX0INTP0R M1	TX0INTP0R M0	Operation mode of interlocking function via INTP0 (interlocking with TMX0 timer)
0	0	Disables operation of interlocking function via INTP0.
0	1	Interlocking mode 1 (timer reset mode): Resets timer when INTP0 is at high level.
1	0	Interlocking mode 2 (timer restart mode): Restarts timer when INTP0 rising edge is detected.
1	1	Interlocking mode 3 (timer output reset mode): Resets timer output from detection of INTP0 rising edge to generation of next interrupt[Note 4].

TX0CMP2R M1	TX0CMP2R M0	Operation mode of interlocking function via comparator 2 output (interlocking with TMX0 timer)
0	0	Disables operation of interlocking function via comparator 2 output.
0	1	Interlocking mode 1 (timer reset mode): Resets timer when comparator 2 output is at high level (CMP2F flag = 1).
1	0	Interlocking mode 2 (timer restart mode): Restarts timer when rising edge of comparator 2 output is detected.
1	1	Interlocking mode 3 (timer output reset mode): Resets timer output from detection of rising edge of comparator 2 output to generation of next interrupt[Note 4].

Notes 1. The values of TX1CR0 to TX1CR3 and TX1CCR0 are rewritten at once only in the TMX0 and TMX1 synchronous start/clear mode.

2. For TMX0 single output, the values of the compare registers are rewritten at once upon restart by comparator output after TX0CR1 is rewritten.

3. For TMX0 dual output, the values of the compare registers are rewritten at once upon restart by comparator output after TX0CR3 is rewritten.

4. Do not set to interlocking mode 3 when in TMX0 and TMX1 synchronous start/clear mode (available only in 78K0/FB2-L).

5. 78K0/FA2-L and 78K0/FB2-L only.

Caution During the timer operation, setting the other bits of TX0CTL3 is prohibited. However, TX0CTL3 can be refreshed (the same value is written).

(5) 16-bit timer Xn operation control register 4 (TXnCTL4)

TXnCTL4 is a register that sets the mode of the interlocking function with comparator 0 and comparator 1.

TXnCTL4 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TXnCTL4 to 00H.

Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-13. Format of 16-bit Timer X0 Operation Control Register 4 (TX0CTL4) (78K0/FA2-L and 78K0/FB2-L only)

Address: FF82H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
TX0CTL4	0	0	TX0CMP1RP	TX0CMP1RM1	TX0CMP1RM0	TX0CMP0RP	TX0CMP0RM1	TX0CMP0RM0

TX0CMP1RP	Selection of timer interlocking with comparator 1 output
0	Comparator 1 output interlocks with TMX0 timer.
1	Comparator 1 output interlocks with TMX1 timer (available only in 78K0/FB2-L).

TX0CMP1RM1	TX0CMP1RM0	Operation mode of interlocking function via comparator 1 output (interlocking with TMX0 timer)
0	0	Disables operation of interlocking function via comparator 1 output.
0	1	Interlocking mode 1 (timer reset mode): Resets timer when comparator 1 output is at high level (CMP1F flag = 1).
1	0	Interlocking mode 2 (timer restart mode): Restarts timer when rising edge of comparator 1 output is detected.
1	1	Interlocking mode 3 (timer output reset mode): Resets timer output from detection of rising edge of comparator 1 output to generation of next interrupt[Note].

TX0CMP0RP	Selection of timer interlocking with comparator 0 output
0	Comparator 0 output interlocks with TMX0 timer.
1	Comparator 0 output interlocks with TMX1 timer (available only in 78K0/FB2-L).

TX0CMP0RM1	TX0CMP0RM0	Operation mode of interlocking function via comparator 0 output (interlocking with TMX0 timer)
0	0	Disables operation of interlocking function via comparator 0 output.
0	1	Interlocking mode 1 (timer reset mode): Resets timer when comparator 0 output is at high level (CMP0F flag = 1).
1	0	Interlocking mode 2 (timer restart mode): Restarts timer when rising edge of comparator 0 output is detected.
1	1	Interlocking mode 3 (timer output reset mode): Resets timer output from detection of rising edge of comparator 0 output to generation of next interrupt[Note].

Note Do not set to interlocking mode 3 when in TMX0 and TMX1 synchronous start/clear mode (available only in 78K0/FB2-L).

Caution During the timer operation, setting the other bits of TX0CTL4 is prohibited. However, TX0CTL4 can be refreshed (the same value is written).

Figure 6-14. Format of 16-bit Timer X1 Operation Control Register 4 (TX1CTL4) (78K0/FB2-L Only)

Address: FF9AH After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	2	<1>	<0>
TX1CTL4	0	0	0	TX1CMP1RM1	TX1CMP1RM0	0	TX1CMP0RM1	TX1CMP0RM0

TX1CMP1RM1	TX1CMP1RM0	Operation mode of interlocking function via comparator 1 output (interlocking with TMX1 timer)
0	0	Disables operation of interlocking function via comparator 1 output.
0	1	Interlocking mode 1 (timer reset mode): Resets timer when comparator 1 output is at high level (CMP1F flag = 1).
1	0	Interlocking mode 2 (timer restart mode): Restarts timer when rising edge of comparator 1 output is detected.
1	1	Interlocking mode 3 (timer output reset mode): Resets timer output from detection of rising edge of comparator 1 output to generation of next interrupt[Note].

TX1CMP0RM1	TX1CMP0RM0	Operation mode of interlocking function via comparator 0 output (interlocking with TMX1 timer)
0	0	Disables operation of interlocking function via comparator 0 output.
0	1	Interlocking mode 1 (timer reset mode): Resets timer when comparator 0 output is at high level (CMP0F flag = 1).
1	0	Interlocking mode 2 (timer restart mode): Restarts timer when rising edge of comparator 0 output is detected.
1	1	Interlocking mode 3 (timer output reset mode): Resets timer output from detection of rising edge of comparator 0 output to generation of next interrupt[Note].

Note Do not set to interlocking mode 3 when in TMX0 and TMX1 synchronous start/clear mode.

Caution During the timer operation, setting the other bits of TX1CTL4 is prohibited. However, TX1CTL4 can be refreshed (the same value is written).

(6) 16-bit timer Xn output control register 0 (TXnIOC0)

TXnIOC0 is a register that sets the timer output.

TXnIOC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TXnIOC0 to 00H.

Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-15. Format of 16-bit Timer X0 Output Control Register 0 (TX0IOC0) (78K0/FA2-L and 78K0/FB2-L only)

Address: FF83H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
TX0IOC0	0	0	0	0	TX0TOC1	TX0TOC0	TX0TOL1	TX0TOL0

TX0TOC1	TOX01 output control
0	Disables timer output (Fixes to low-level output when TX0TOL1 = 0, and fixes to high-level output when TX0TOL1 = 1.)
1	Enables timer output (PWM output)

TX0TOC0	TOX00 output control
0	Disables timer output (Fixes to low-level output when TX0TOL0 = 0, and fixes to high-level output when TX0TOL0 = 1.)
1	Enables timer output (PWM output)

TX0TOL1	Default TOX01 output state setting
0	Normal output (low level)
1	Inverted output (high level)

TX0TOL0	Default TOX00 output state setting
0	Normal output (low level)
1	Inverted output (high level)

Cautions 1. During the timer operation, setting the other bits of TX0IOC0 is prohibited. However, TX0IOC0 can be refreshed (the same value is written).

2. The actual TOX00/P31/INTP2/TOOLC1 and TOX01/P32/INTP3/TOOLD1 pin outputs are determined depending on PM31, P31, PM32, and P32 besides TOX00 and TOX01 outputs.

Figure 6-16. Format of 16-bit Timer X1 Output Control Register 0 (TX1IOC0) (78K0/FB2-L Only)

Address: FF9BH After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
TX1IOC0	0	0	0	0	TX1TOC1	TX1TOC0	TX1TOL1	TX1TOL0

TX1TOC1	TOX11 output control
0	Disables timer output (Fixes to low-level output when TX1TOL1 = 0, and fixes to high-level output when TX1TOL1 = 1.)
1	Enables timer output (PWM output)

TX1TOC0	TOX10 output control
0	Disables timer output (Fixes to low-level output when TX1TOL0 = 0, and fixes to high-level output when TX1TOL0 = 1.)
1	Enables timer output (PWM output)

TX1TOL1	Default TOX11 output state setting
0	Normal output (low level)
1	Inverted output (high level)

TX1TOL0	Default TOX10 output state setting
0	Normal output (low level)
1	Inverted output (high level)

Cautions 1. During the timer operation, setting the other bits of TX1IOC0 is prohibited. However, TX1IOC0 can be refreshed (the same value is written).

2. The actual TOX10/P33 and TOX11/P34/INTP4 pin outputs are determined depending on PM33, P33, PM34, and P34 besides TOX10 and TOX11 outputs.

(7) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P31/TOX00/INTP2/TOOLC1, P32/TOX01/INTP3/TOOLD1, P33/TOX10, and P34/TOX11/INTP4 pins for timer output, set PM31 to PM34 and the output latches of P31 to P34 to 0.

PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM3 to FFH.

Figure 6-17. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode register 3 of the 78K0/FB2-L products. For the format of port mode register 3 of other products, refer to **(1) Port mode registers (PMxx)** in **4.3 Registers Controlling Port Function**.

The following table shows how the operation modes (TMX0-only mode, TMX1-only mode, TMX0 and TMX1 synchronous start mode, TMX0 and TMX1 synchronous start/clear mode) and register setting bits controlling 16-bit timers X0 and X1 relate to each other.

Table 6-2. Register Setting Bits Controlling Operation Mode and 16-bit Timers X0 and X1 (1/2)

Register	Bit	Operation Mode					
		TMXn-Only Mode (n = 0, 1)		Synchronous Start Mode[Note]		Synchronous Start/Clear Mode[Note]	
		TMX0	TMX1[Note]	Master (TMX0)	Slave (TMX1)	Master (TMX0)	Slave (TMX1)
TX0CTL0	TX0TMC	Setting	–	Setting		Setting	
	TX0CKS2 to TX0CKS0	Setting	–	Setting	–	Setting	
TX1CTL0	TX1TMC	–	Setting	–	–	–	–
	TX1CKS2 to TX1CKS0	–	Setting	–	Setting	–	–
TX0CTL1	TX0INTPST	Setting	–	0		Setting	
	TX0PWMCE	Setting	–	Setting	–	Setting	–
	TX0PWMCINV	Setting	–	Setting		Setting	
	TX0PWM	Setting	–	Setting	–	Setting	–
TX1CTL1	TX1PWMCE	–	Setting	–	Setting	–	Setting
	TX1PWM	–	Setting	–	Setting	–	Setting
	TX1MD1	–	0	–	0	–	1
	TX1MD0	–	0	–	1	–	0
TX0CTL2	TX0TRGS	Setting	–	Setting	–	Setting	–
	TX0ADEN	Setting	–	Setting	–	Setting	–
	TX0CCS	Setting	–	Setting	–	Setting	–
TX1CTL2	TX1ADEN	–	Setting	–	Setting	–	Setting
	TX1CCS	–	Setting	–	Setting	–	Setting
TX0CTL3	TX0CMPLDSET1, TX0CMPLDSET0	Setting	–	Setting	–	Setting	
	TX0INTP0RM1, TX0INTP0RM0	Setting	–	Setting	–	Setting (setting TX0INTP0RM1 = 1 and TX0INTP0RM0 = 1 is prohibited)	
	TX0CMP2RM1, TX0CMP2RM0	Setting	–	Setting	–	Setting (setting TX0CMP2RM1 = 1 and TX0CMP2RM0 = 1 is prohibited)	

Note 78K0/FB2-L only

Table 6-2. Register Setting Bits Controlling Operation Mode and 16-bit Timers X0 and X1 (2/2)

Register	Bit	Operation mode					
		TMXn-Only Mode (n = 0, 1)		Synchronous Start Mode[Note]		Synchronous Start/Clear Mode[Note]	
		TMX0	TMX1[Note]	Master (TMX0)	Slave (TMX1)	Master (TMX0)	Slave (TMX1)
TX0CTL4	TX0CMP1RP	Setting		Setting		–	–
	TX0CMP1RM1, TX0CMP1RM0	Setting is valid when TX0CMP 1RP = 0	–	Setting is valid when TX0CMP 1RP = 0	–	–	–
	TX0CMP0RP	Setting		Setting		–	–
	TX0CMP0RM1, TX0CMP0RM0	Setting is valid when TX0CMP 0RP = 0	–	Setting is valid when TX0CMP 0RP = 0	–	–	–
TX1CTL4	TX1CMP1RM1, TX1CMP1RM0	–	Setting is valid when TX0CMP 1RP = 1	–	Setting is valid when TX0CMP 1RP = 1	–	–
	TX1CMP0RM1, TX1CMP0RM0	–	Setting is valid when TX0CMP 0RP = 1	–	Setting is valid when TX0CMP 0RP = 1	–	–
TX0IOC0	TX0TOC1	Setting	–	Setting	–	Setting	–
	TX0TOC0	Setting	–	Setting	–	Setting	–
	TX0TOL1	Setting	–	Setting	–	Setting	–
	TX0TOL0	Setting	–	Setting	–	Setting	–
TX1IOC0	TX1TOC1	–	Setting	–	Setting	–	Setting
	TX1TOC0	–	Setting	–	Setting	–	Setting
	TX1TOL1	–	Setting	–	Setting	–	Setting
	TX1TOL0	–	Setting	–	Setting	–	Setting

Note 78K0/FB2-L only

The following tables show how the operation modes (TMX0-only mode, TMX1-only mode, TMX0 and TMX1 synchronous start mode, TMX0 and TMX1 synchronous start/clear mode) and the trigger source of each operation (start, capture, interlocking function) relate to each other.

Table 6-3. Operation Mode and Trigger Source

(1) Timer start

Trigger Source	Operation Mode					
	TMXn-Only Mode (n = 0, 1)		Synchronous Start Mode[Note]		Synchronous Start/Clear Mode[Note]	
	TMX0	TMX1[Note]	Master (TMX0)	Slave (TMX1)	Master (TMX0)	Slave (TMX1)
INTP0	Usable	Not usable	Not usable		Usable	

(2) Capture

Trigger Source	Operation Mode					
	TMXn-Only Mode (n = 0, 1)		Synchronous Start Mode[Note]		Synchronous Start/Clear Mode[Note]	
	TMX0	TMX1[Note]	Master (TMX0)	Slave (TMX1)	Master (TMX0)	Slave (TMX1)
INTP0	Usable	Not usable	Usable	Not usable	Usable	Not usable
INTCMP1	Not usable	Usable	Not usable	Usable	Not usable	Usable
INTCMP2	Usable	Not usable	Usable	Not usable	Usable	Not usable

(3) Interlocking function

Trigger Source	Operation Mode					
	TMXn-Only Mode (n = 0, 1)		Synchronous Start Mode[Note]		Synchronous Start/Clear Mode[Note]	
	TMX0	TMX1[Note]	Master (TMX0)	Slave (TMX1)	Master (TMX0)	Slave (TMX1)
INTP0	Usable	Not usable	Usable	Not usable	Usable	
INTCMP0		Usable		Usable	Not usable	
INTCMP1						
INTCMP2		Not usable		Not usable	Usable	

Note Available only in 78K0/FB2-L

6.4 Operation of 16-Bit Timer/Event Counter 00

(1) Interval timer operation

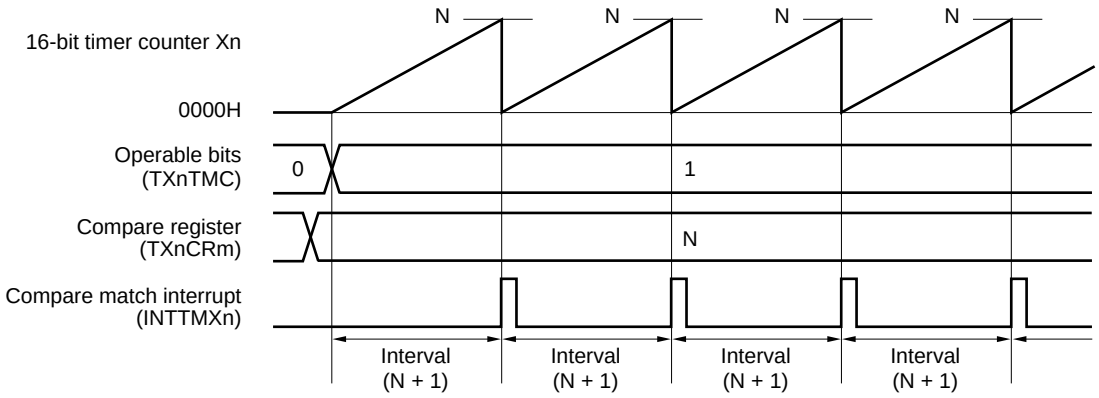
If bit 7 (TXnTMC) of the 16-bit timer Xn operation control register 0 (TXnCTL0) is set to 1, the count operation is started in synchronization with the count clock.

When the value of the 16-bit timer counter Xn (TMXn) later matches the value of TXnCRm, TMXn is cleared to 0000H and a match interrupt signal (INTTMXn) is generated. This INTTMXn signal enables TMXn to operate as an interval timer.

Remarks 1. For how to enable the INTTMXn interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

2. m = 1, 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

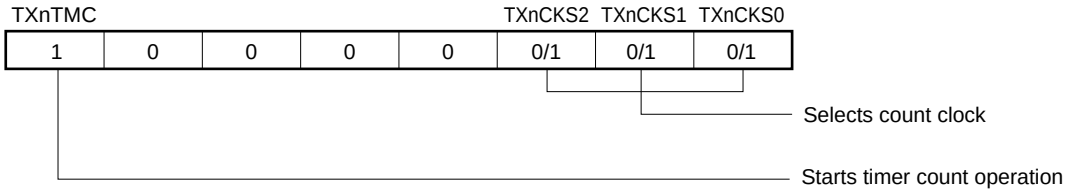
Figure 6-18. Basic Timing Example of Interval Timer Operation



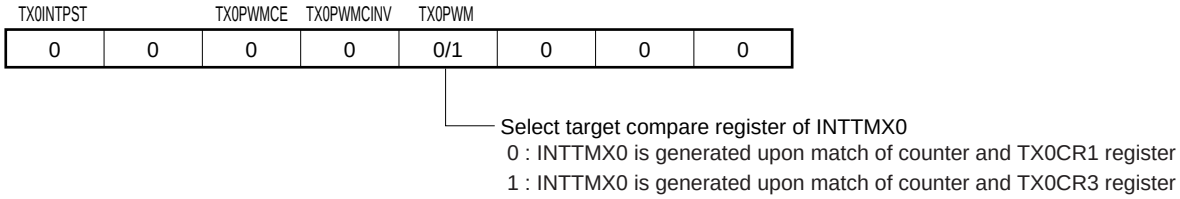
Remark m = 1, 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-19. Example of Register Settings for Interval Timer Operation

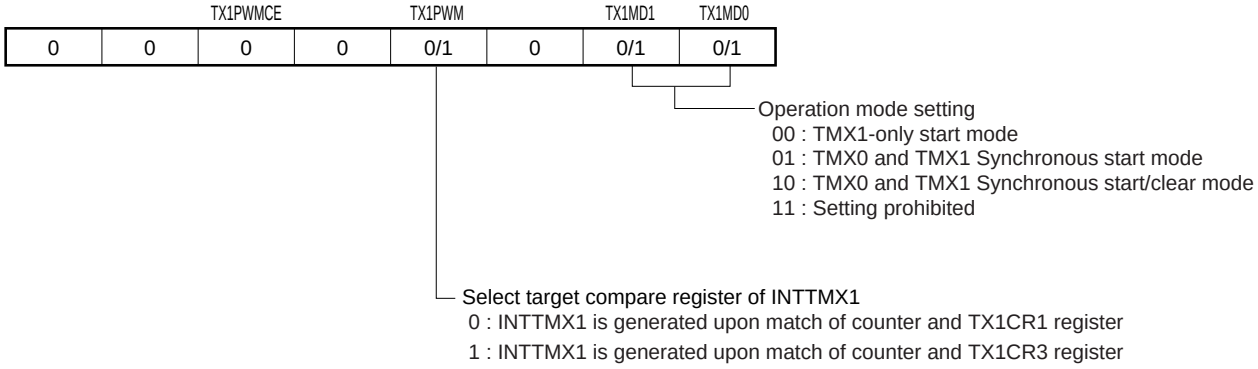
(a) 16-bit timer Xn operation control register 0 (TXnCTL0)



(b) 16-bit timer X0 operation control register 1 (TX0CTL1)



(c) 16-bit timer X1 operation control register 1 (TX1CTL1) (78K0/FB2-L only)



(d) 16-bit timer Xn capture/compare register m (TXnCRm)

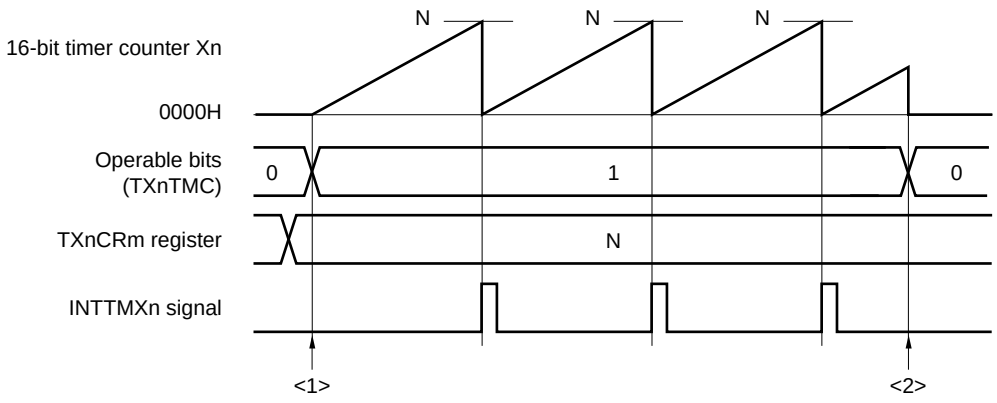
If N is set to TXnCRm, the interval time is as follows.

- Interval time = (N + 1) × Count clock cycle

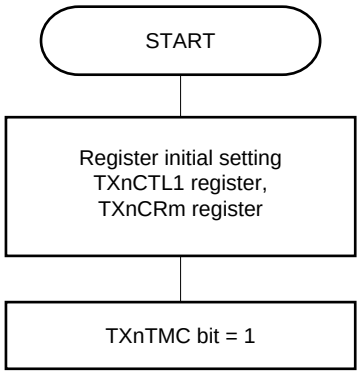
Setting TXnCRm to 0000H is prohibited.

Remark m = 1, 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-20. Example of Software Processing for Interval Timer Function



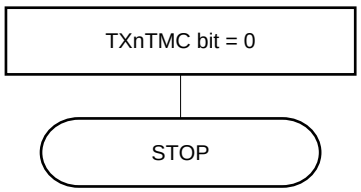
<1> Count operation start flow



Initial setting of these registers is performed before setting the TXnTMC bit to 1.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TXnTMC bit bit to 0.

Remark m = 1, 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(2) A/D conversion start timing signal output

If bit 1 (TXnADEN) of the 16-bit timer Xn operation control register 2 (TXnCTL2) is set to 1, the generation of the A/D conversion synchronization trigger is enabled. If bit 7 (TXnTMC) of the 16-bit timer Xn operation control register 0 (TXnCTL0) is set to 1, the count operation is started in synchronization with the count clock.

When the value of the 16-bit timer counter Xn (TMXn) later matches the value of TXnCCR0, the A/D conversion synchronization trigger is generated. When the value of TMXn matches the value of TXnCRm, TMXn is cleared to 0000H.

To output the A/D conversion start timing signal, satisfy the relationship between TXnCCR0 and TXnCRm as follows.

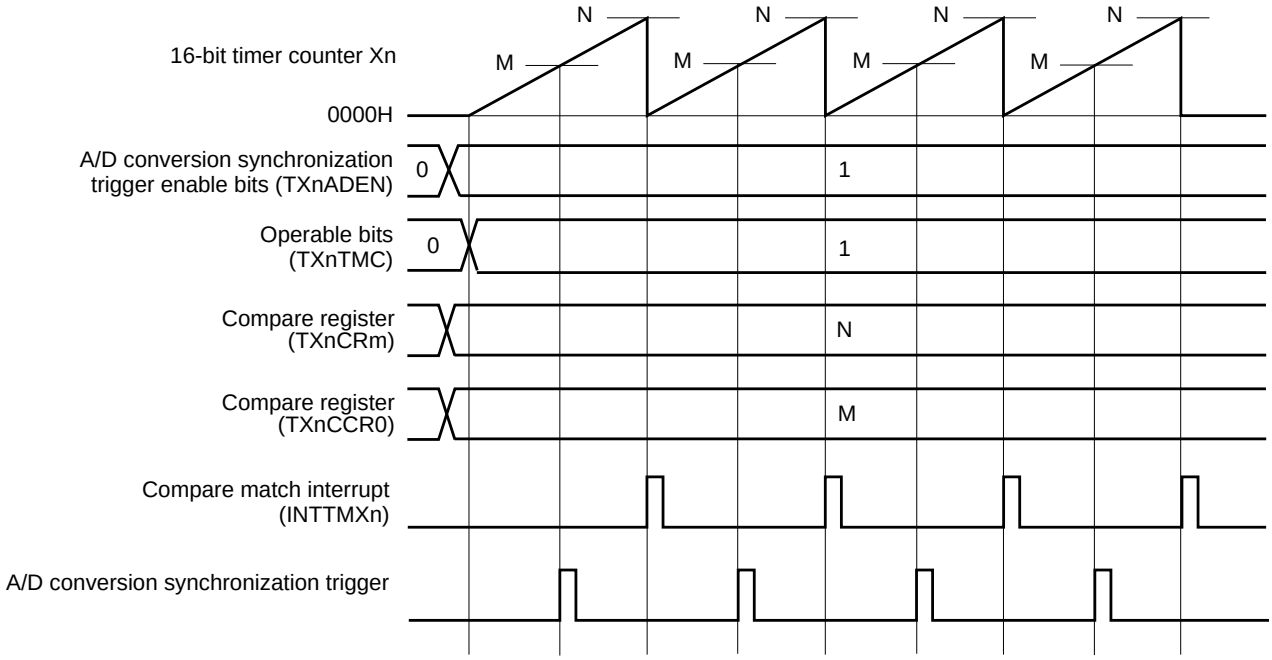
- TXnCCR0 < TXnCRm

If this relationship is not satisfied, the A/D conversion trigger is not generated.

Remarks 1. For details of the A/D conversion in combination with 16 bit timer X0 or X1, refer to **11.4.2 Basic operation of A/D converter (timer trigger mode)**.

2. m = 1, 3
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

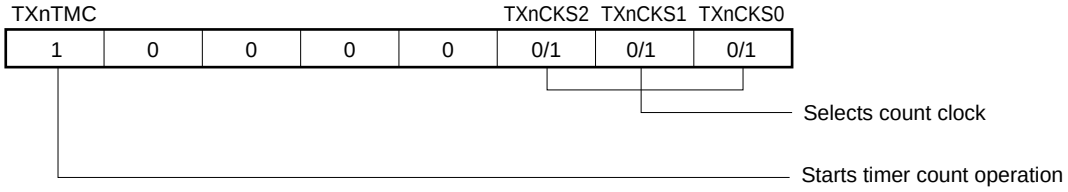
Figure 6-21. Basic Timing Example of A/D Conversion Start Timing Signal Output



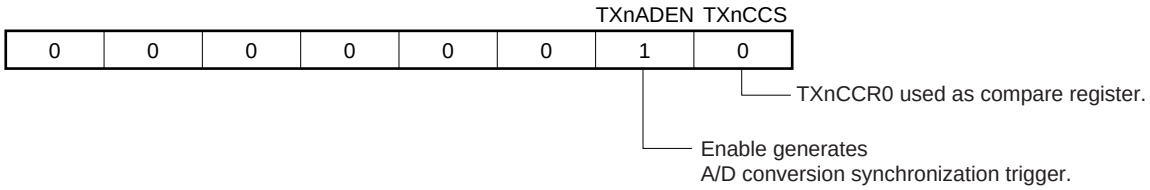
Remark m = 1, 3
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-22. Example of Register Settings for A/D Conversion Start Timing Signal Output

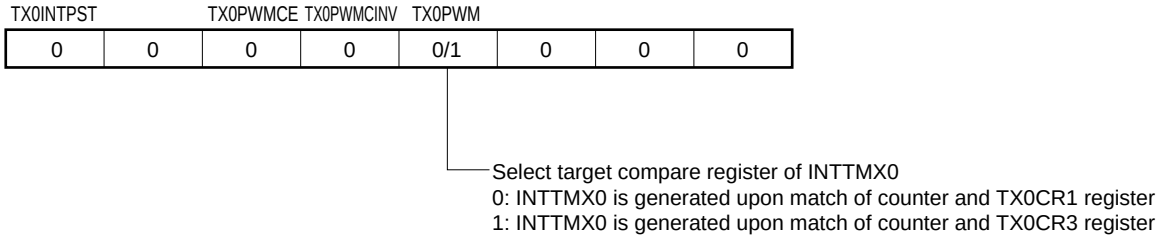
(a) 16-bit timer Xn operation control register 0 (TXnCTL0)



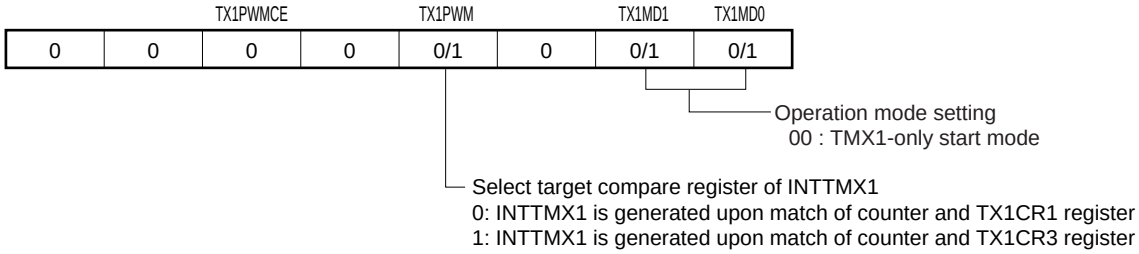
(b) 16-bit timer Xn operation control register 2 (TXnCTL2)



(c) 16-bit timer X0 operation control register 1 (TX0CTL1)



(d) 16-bit timer X1 operation control register 1 (TX1CTL1) (78K0/FB2-L only)



(e) 16-bit timer Xn compare register m (TXnCRm)

If N is set to TXnCRm, the A/D conversion synchronization trigger generation period is as follows.

- The A/D conversion synchronization trigger generation period = (N + 1) × Count clock cycle

Setting TXnCRm to 0000H is prohibited.

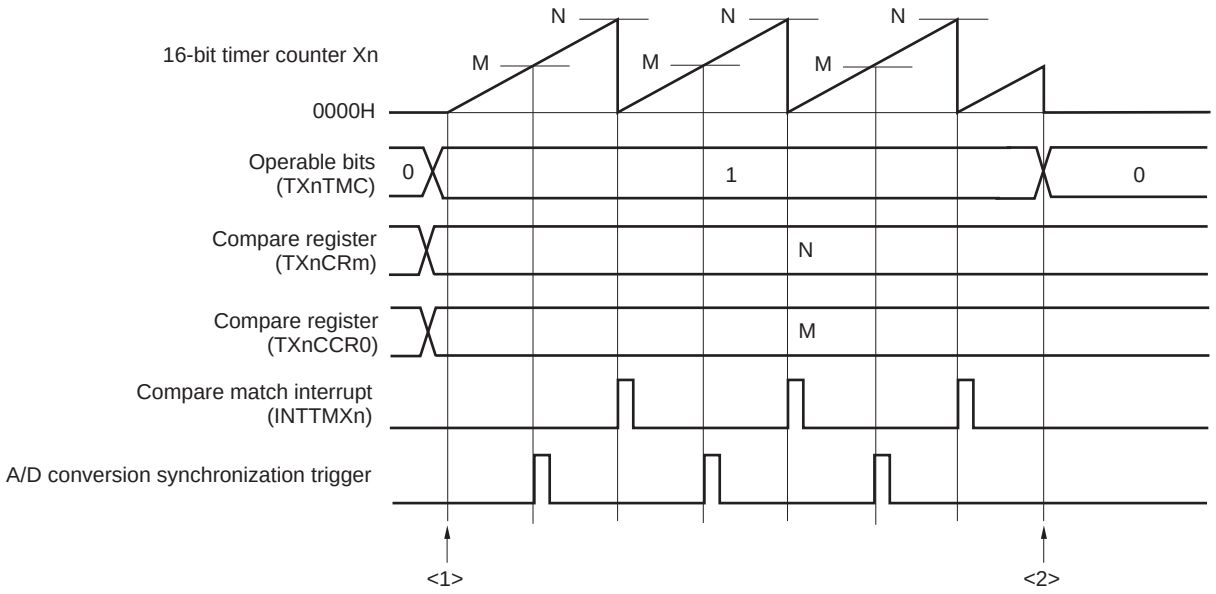
(f) 16-bit timer Xn capture/compare register 0 (TXnCCR0)

If M is set to TXnCCR0, the A/D conversion synchronization trigger is generated at a time later than counting 0000H only for M.

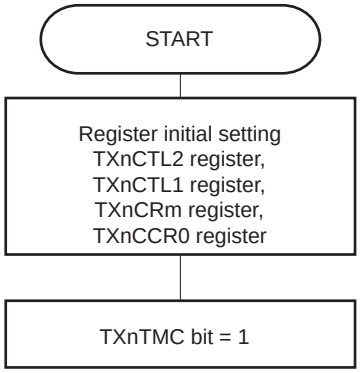
Remarks 1. m = 1, 3
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

2. For details of A/D conversion in combination with 16 bit timer X0 or X1, refer to **11.4.2 Basic operation of A/D converter (timer trigger mode)**.

Figure 6-23. Example of Software Processing for A/D Conversion Start Timing Signal Output



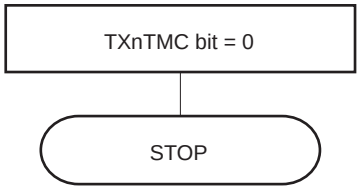
<1> Count operation start flow



Initial setting of these registers is performed before setting the TXnTMC bit to 1.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TXnTMC bit bit to 0.

Remarks 1. m = 1, 3

n = 0 : 78K0/FY2-L, 78K0/FA2-L

n = 0, 1 : 78K0/FB2-L

2. For details of A/D conversion in combination with 16 bit timer X0 or X1, refer to **11.4.2 Basic operation of A/D converter (timer trigger mode)**.

(3) Capture function

If bit 7 (TXnTMC) of the 16-bit timer Xn operation control register 0 (TXnCTL0) is set to 1, the count operation is started in synchronization with the count clock.

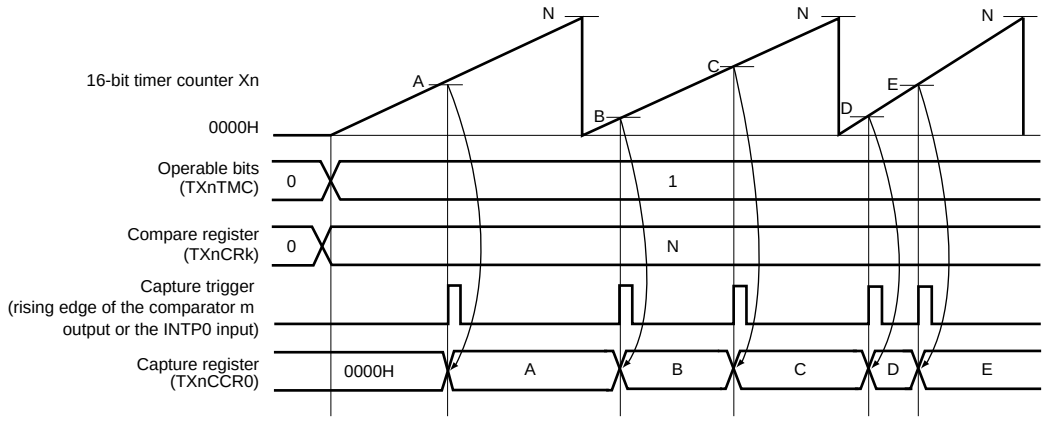
Then, when the rising edge of the comparator m output or the INTP0 input is detected, the count value of the 16-bit timer counter Xn (TMXn) is captured to a 16-bit timer Xn capture/compare register 0 (TXnCCR0) .

When the value of TMXn later matches the value of TXnCRk, TMXn is cleared to 0000H.

Remarks 1. m = 1, 2 : 78K0/FB2-L
m = 2 : 78K0/FY2-L, 78K0/FA2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L
k = 1, 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L

2. Capture trigger of 16-bit timer X0 : Rising edge of the comparator 2 output or the INTP0 input
Capture trigger of 16-bit timer X1 : Rising edge of the comparator 1 output

Figure 6-24. Basic Timing Example of Capture Function Operation

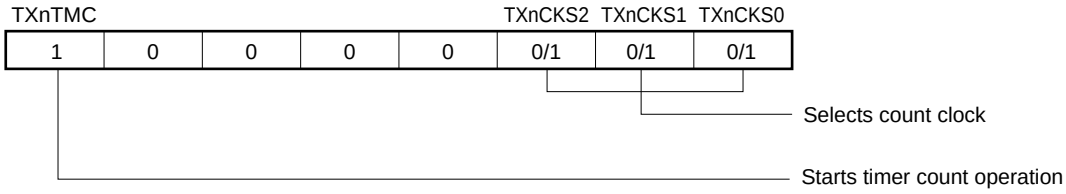


Remarks 1. m = 1, 2 : 78K0/FB2-L
m = 2 : 78K0/FY2-L, 78K0/FA2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L
k = 1, 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L

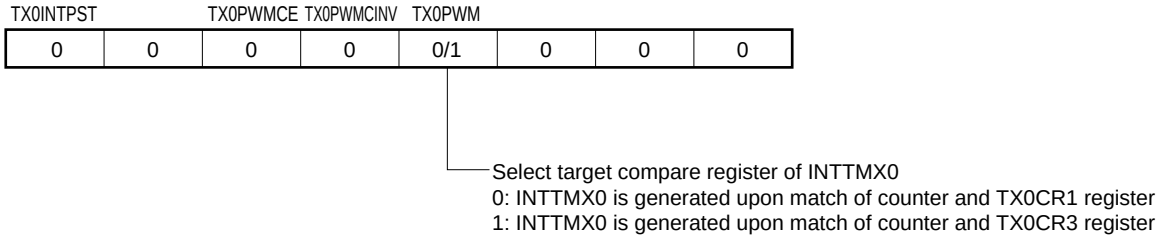
2. Capture trigger of 16-bit timer X0 : Rising edge of the comparator 2 output or the INTP0 input
Capture trigger of 16-bit timer X1 : Rising edge of the comparator 1 output

Figure 6-25. Example of Register Settings for Capture Function Operation

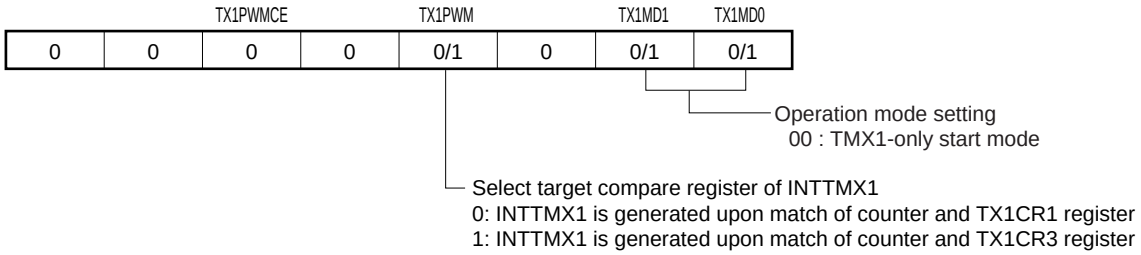
(a) 16-bit timer Xn operation control register 0 (TXnCTL0)



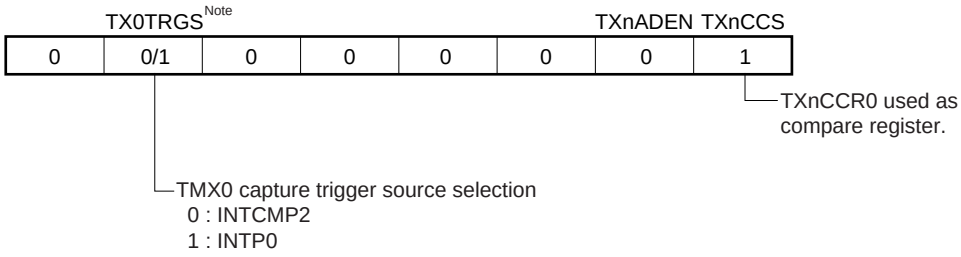
(b) 16-bit timer X0 operation control register 1 (TX0CTL1)



(c) 16-bit timer X1 operation control register 1 (TX1CTL1) (78K0/FB2-L only)



(d) 16-bit timer Xn operation control register 2 (TXnCTL2)

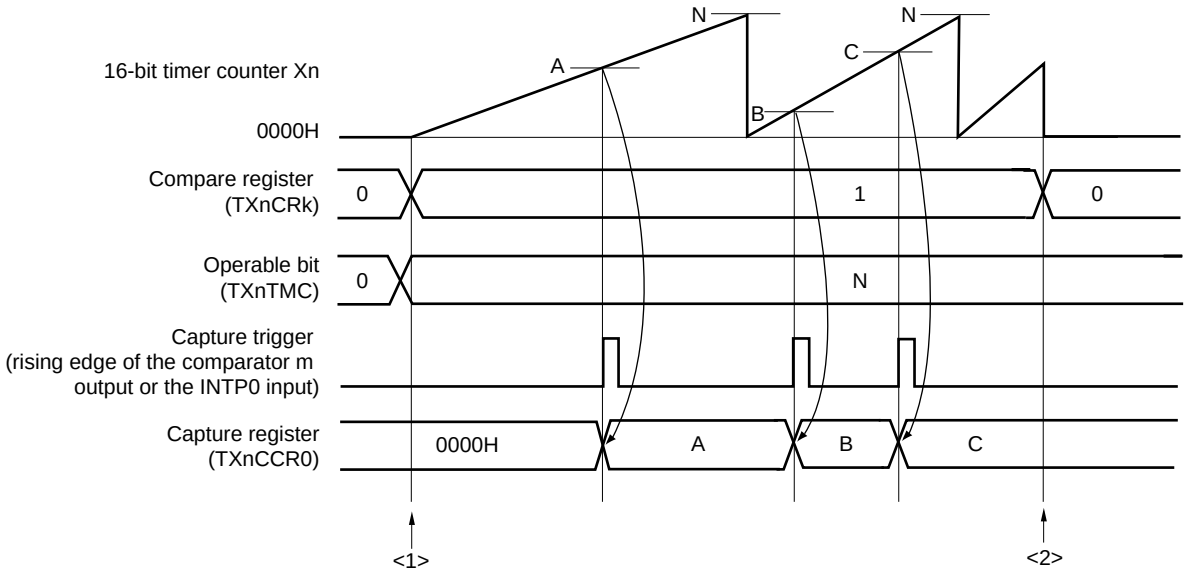


Note The TX0TRGS bit is not provided in the TX1CTL2 register.

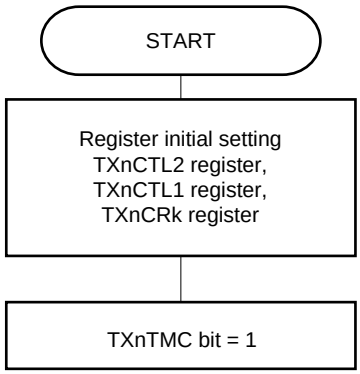
Remarks 1. m = 1, 2 : 78K0/FB2-L
m = 2 : 78K0/FY2-L, 78K0/FA2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-
n = 0, 1 : 78K0/FB2-L

2. Capture trigger of 16-bit timer X0 : INTCMP2 or INTP0
Capture trigger of 16-bit timer X1 : INTCMP1

Figure 6-26. Example of Software Processing for Capture Function Operation



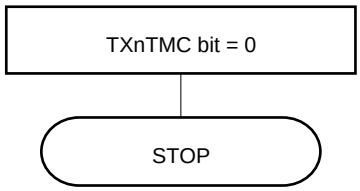
<1> Count operation start flow



Initial setting of these registers is performed before setting the TXnTMC bit to 1.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TXnTMC bit bit to 0.

Remarks 1. m = 1, 2 : 78K0/FB2-L
m = 2 : 78K0/FY2-L, 78K0/FA2-L
n = 0 : 78K0/FY2-L, 78K0/FA2-
n = 0, 1 : 78K0/FB2-L
k = 1, 3 : 78K0/FY2-L, 78K0/FA2-L, 78K0/FB2-L

2. Capture trigger of 16-bit timer X0 : Rising edge of the comparator 2 output or the INTP0 input
Capture trigger of 16-bit timer X1 : Rising edge of the comparator 1 output

6.5 Operation of PWM output operation of 16-Bit Timers X0 and X1

(1) PWM output operation (TMXn-only mode, single output)

PWM output is started from TOXn0 if bit 7 (TXnTMC) of TXnCTL0 is set to 1 after setting the count value of the inverted output to TXnCR0 and the count value of the cycle to TXnCR1.

- Pulse cycle = (Set value of TXnCR1 + 1) × Count clock cycle
- Duty = (Set value of TXnCR1 – Set value of TXnCR0) / (Set value of TXnCR1 + 1)

TXnCR0 and TXnCR1 can be rewritten while the timers are operating, and the duty and the pulse cycle can be changed. When rewriting both TXnCR0 and TXnCR1, be sure to rewrite TXnCR0 before rewriting TXnCR1. If only TXnCR1 needs to be changed, there is no need to rewrite TXnCR0. If only TXnCR0 needs to be changed, rewrite TXnCR0 and then write the same value to TXnCR1.

The output is changed when the INTTMXn interrupt is generated immediately after TXnCR1 is written. However, if TXnCR1 is rewritten during the clock[Note] cycle in which the INTTMXn interrupt was generated, or during the previous two clock[Note] cycles, the output will be changed when the INTTMXn interrupt subsequent to this INTTMXn interrupt is generated. Note also that if TXnCR0 or TXnCR1 is written with a different value in the period between when TXnCR1 is written and when the output changes, the output will be changed to this different value, not the originally specified value.

To specify PWM output from TOXn0, set TXnCR0 and TXnCR1 to a value in the following range:

0000H ≤ TXnCR0 ≤ TXnCR1 ≤ FFFFH

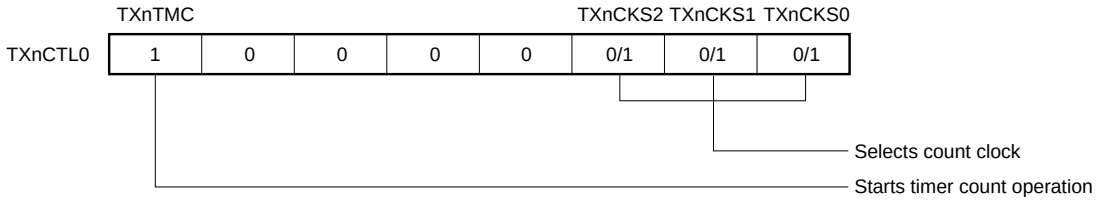
If TXnCR0 = TXnCR1 is specified, the output will be set to the default status (fixed).

Note Count clock of 16-bit timer Xn

Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-27. Example of Register Settings for PWM Output Operation (Single Mode) (1/2)

(a) 16-bit timer Xn operation control register 0

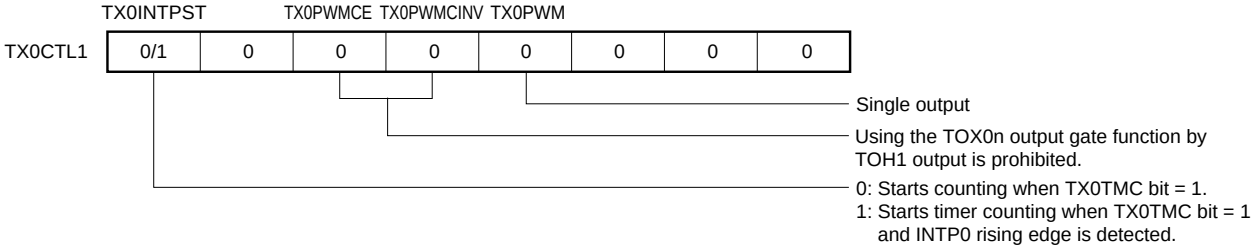


Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

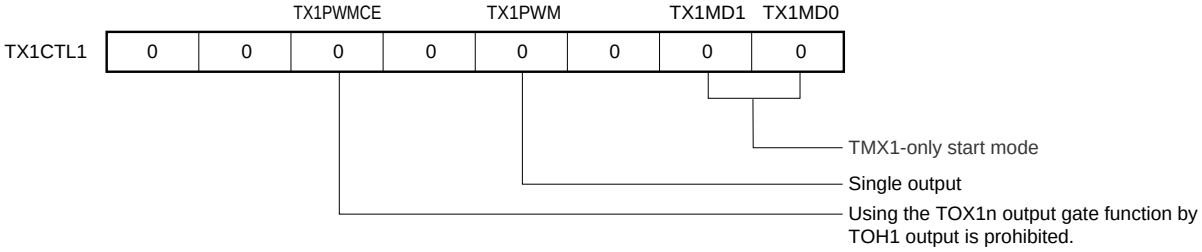
Figure 6-27. Example of Register Settings for PWM Output Operation (Single Mode) (2/2)

(b) 16-bit timer Xn operation control register 1

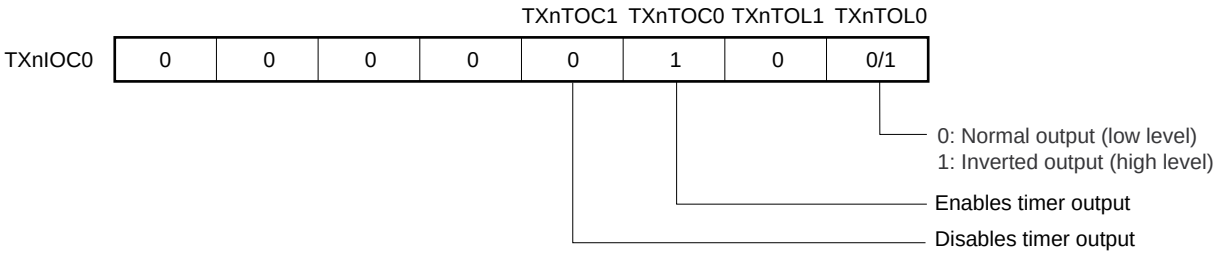
- PWM output: TOX00 pin



- PWM output: TOX10 pin

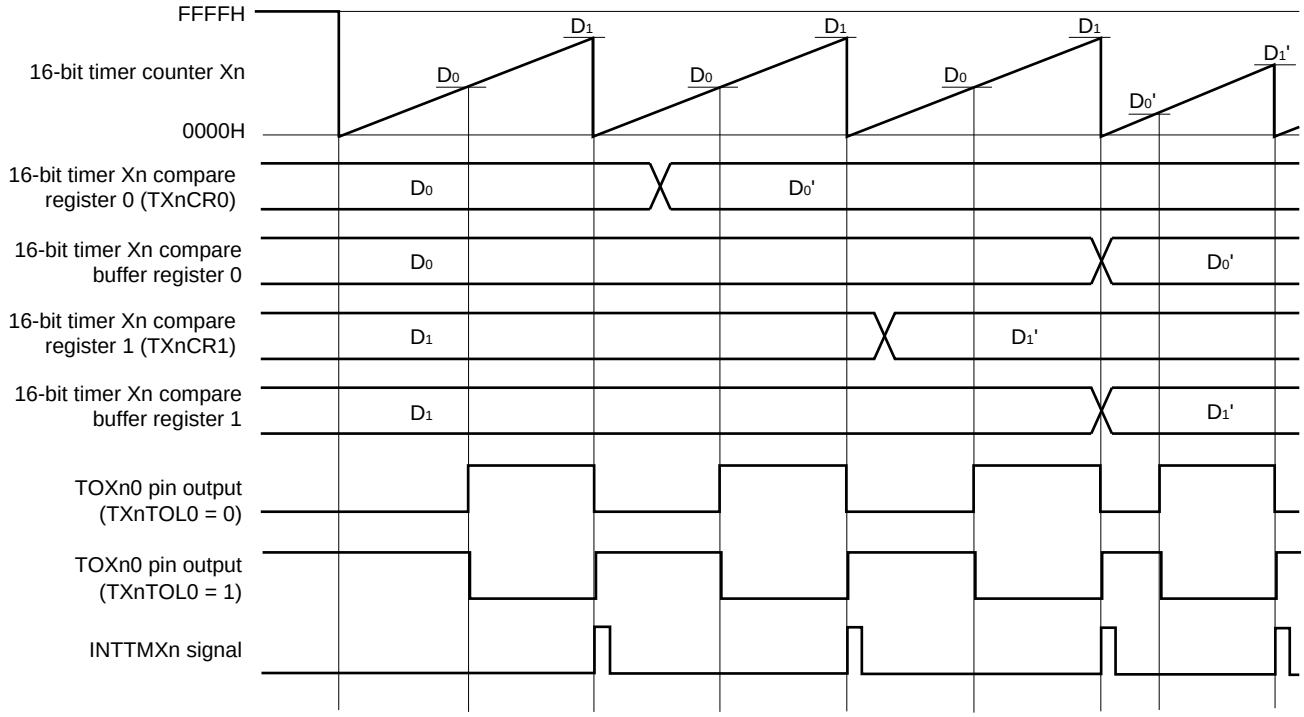


(c) 16-bit timer Xn output control register 0



Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-28. PWM Output Timing (TMXn-only Operation, PWM output: TOXn0 pin)



Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(2) PWM output operation (TMXn-only mode, dual output)

PWM outputs are started when the count value of the inverted output of TOXn0 is set to TXnCR0 and TXnCR1, the count value of the inverted output of TOXn1 and the count value of the cycle are set to TXnCR2 and TXnCR3, and 1 is set to bit 7 (TXnTMC) of TXnCTL0.

- Pulse cycle and duty of TOXn0
  - Pulse cycle = (Set value of TXnCR3 + 1) × Count clock cycle
  - Duty = (Set value of TXnCR1 − Set value of TXnCR0) / (Set value of TXnCR3 + 1)

- Pulse cycle and duty of TOXn1
  - Pulse cycle = (Set value of TXnCR3 + 1) × Count clock cycle
  - Duty = (Set value of TXnCR3 − Set value of TXnCR2) / (Set value of TXnCR3 + 1)

TXnCR0 to TXnCR3 can be rewritten while the timers are operating, and the duty and the pulse cycle can be changed. When rewriting TXnCR0 to TXnCR3, be sure to rewrite TXnCR0 to TXnCR2 before rewriting TXnCR3. Registers among TXnCR0 to TXnCR2 that are not to be changed do not have to be rewritten. If one of TXnCR0 to TXnCR2 needs to be changed but TXnCR3 does not need to be changed, be sure to write the same value to TXnCR3.

The output is changed when the INTTMXn interrupt is generated immediately after TXnCR3 is written. However, if TXnCR3 is rewritten during the clock[Note] cycle in which the INTTMXn interrupt was generated, or during the previous two clock[Note] cycles, the output will be changed when the INTTMXn interrupt subsequent to this INTTMXn interrupt is generated. Note also that if TXnCR0 to TXnCR3 are written with a different value in the period between when TXnCR3 is written and when the output changes, the output will be changed to this different value, not the originally specified value.

To specify PWM output from TOXn0, set TXnCR0 and TXnCR1 to a value in the following range:

0000H ≤ TXnCR0 ≤ TXnCR1 ≤ TXnCR3

If TXnCR0 = TXnCR1 is specified, the output will be set to the default status (fixed).

To specify PWM output from TOXn1, set TXnCR2 and TXnCR3 to a value in the following range:

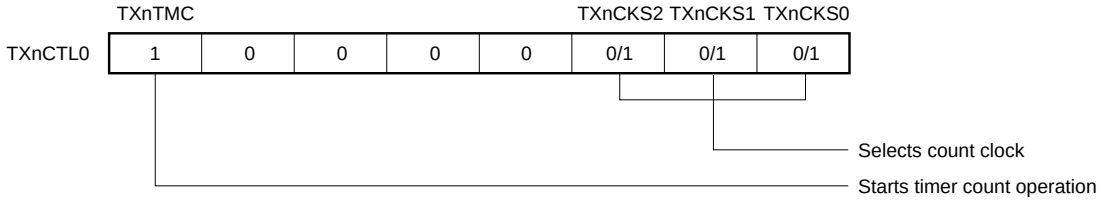
0000H ≤ TXnCR2 ≤ TXnCR3 ≤ FFFFH

If TXnCR2 = TXnCR3 is specified, the output will be set to the default status (fixed).

Note Count clock of 16-bit timer Xn

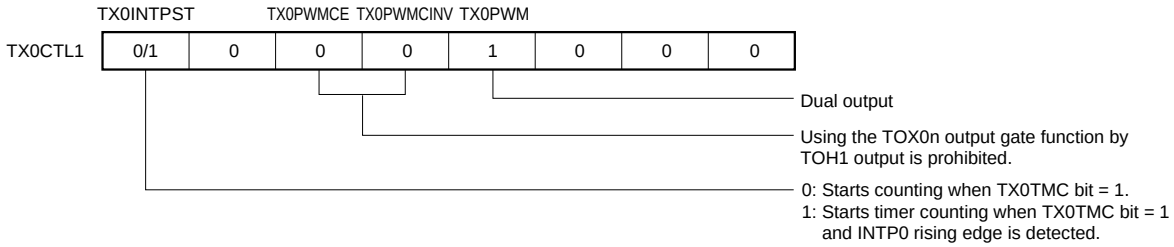
Figure 6-29. Example of Register Settings for PWM Output Operation (Dual Mode) (1/2)

(a) 16-bit timer Xn operation control register 0

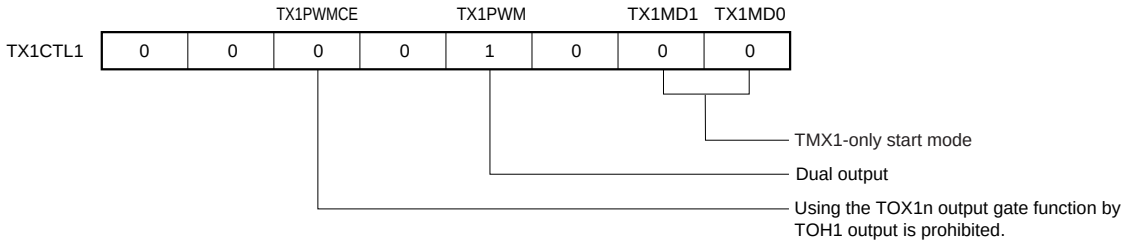


(b) 16-bit timer Xn operation control register 1

- PWM output: TOX00 and TOX01 pins



- PWM output: TOX10 and TOX11 pins

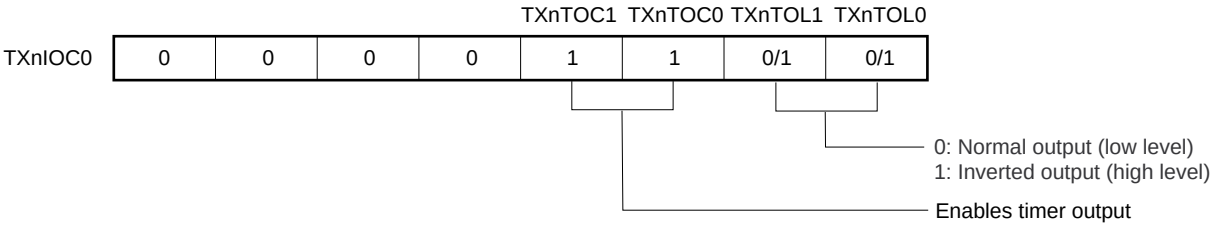


Remark n = 0 : 78K0/FA2-L

n = 0, 1 : 78K0/FB2-L

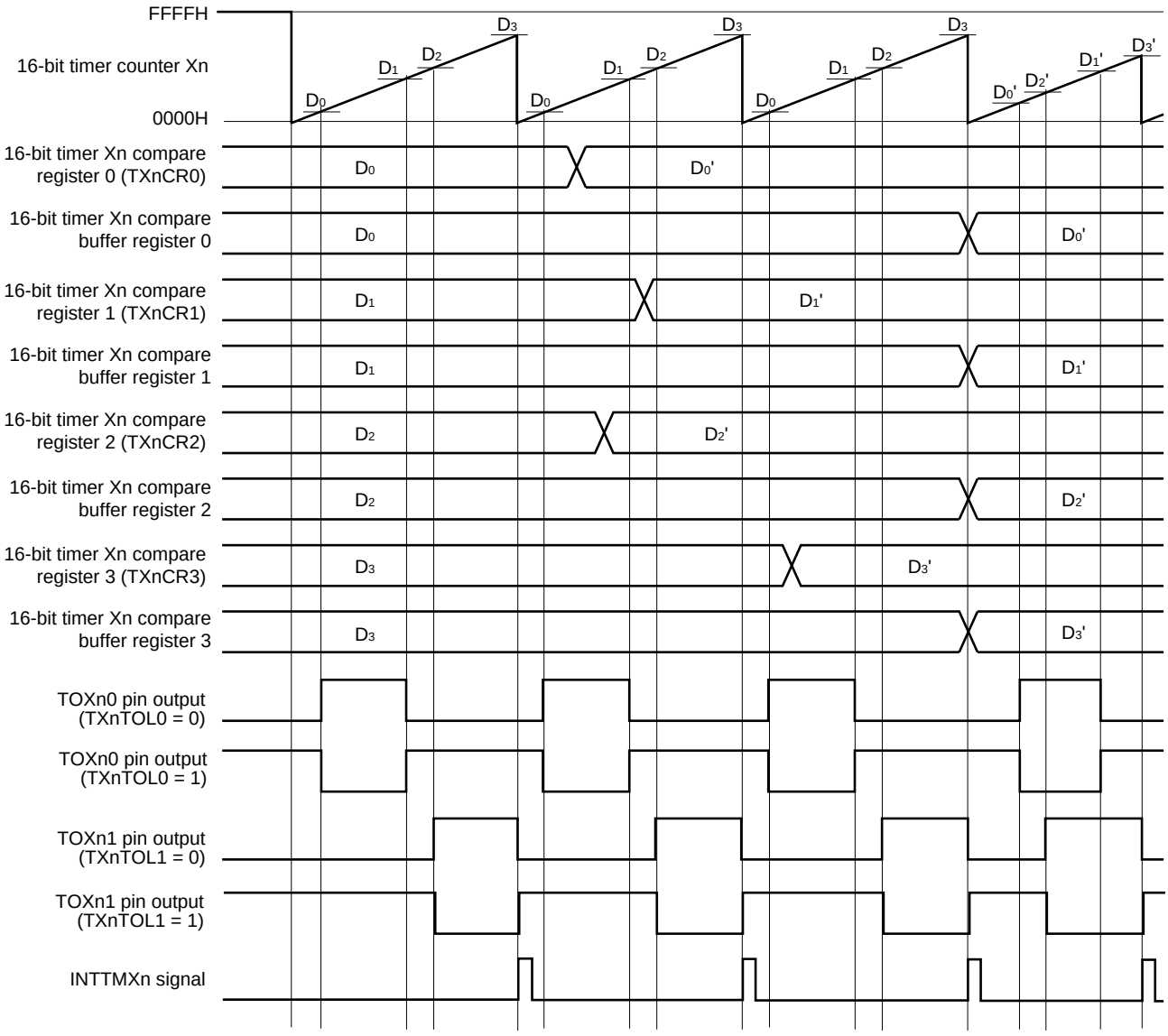
Figure 6-29. Example of Register Settings for PWM Output Operation (Dual Mode) (2/2)

(c) 16-bit timer Xn output control register 0



Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Figure 6-30. PWM Output Timing (TMXn-only Operation, PWM output: TOXn0 and TOXn1 pins)



Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

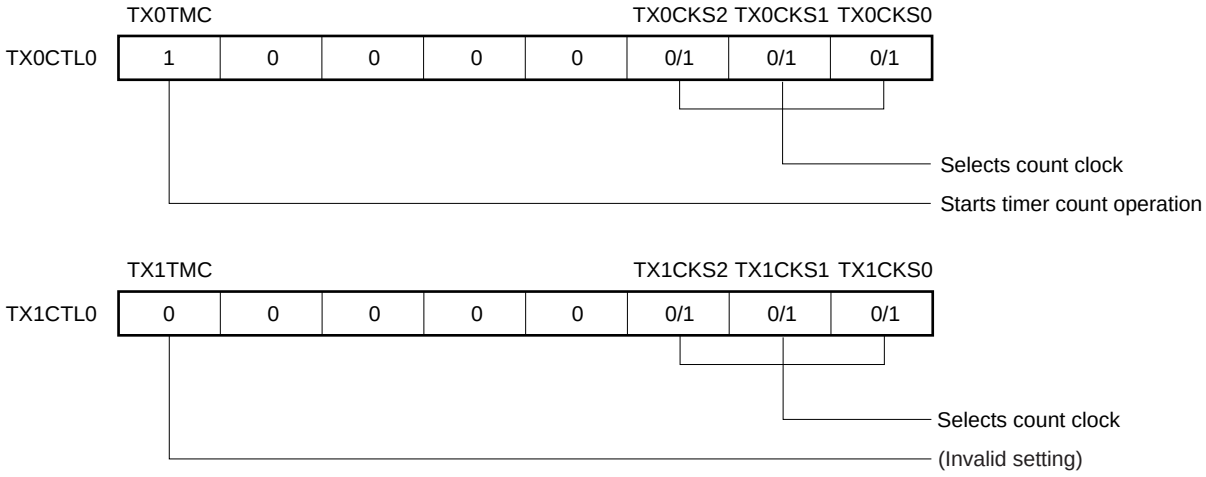
(3) PWM output operation (TMX0 and TMX1 synchronous start mode) (78K0/FB2-L only)

Output from the two timer outputs of TMX0 and TMX1 (up to 4 outputs) is simultaneously started.
Setting bit 7 (TX0TMC) of TX0CTL1 to 1 starts PWM output.
Setting bit 7 (TX0TMC) of TX0CTL1 to 0 stops PWM output.

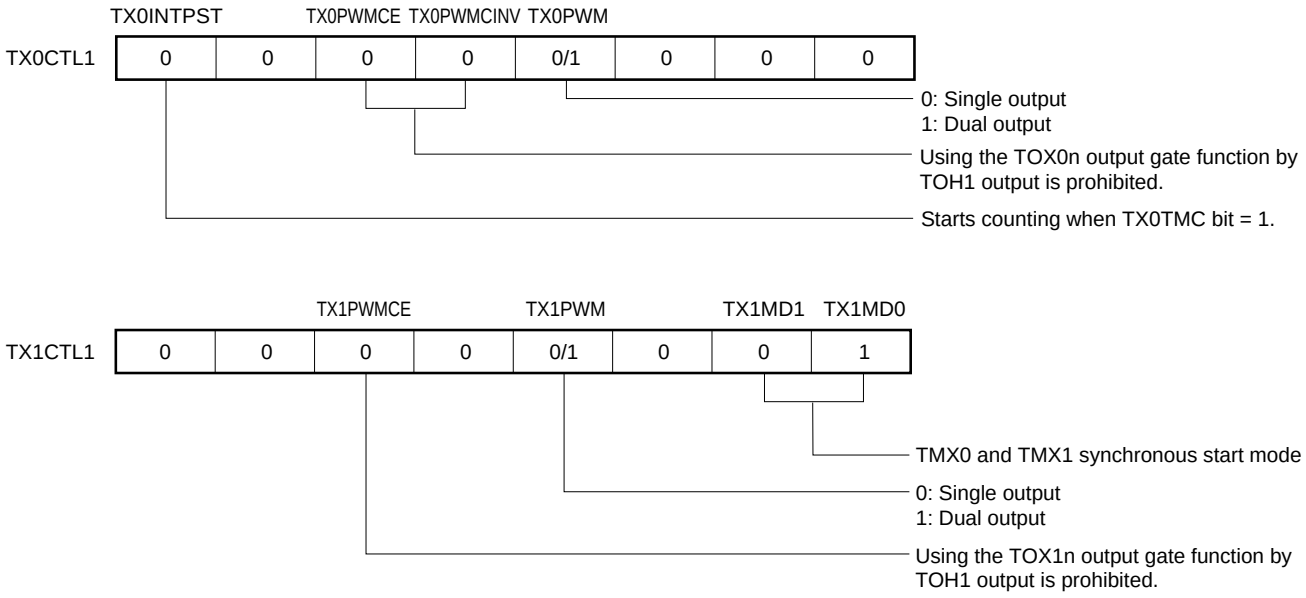
Remark This mode is simultaneously used when starting or stopping output.
While the TMX0 and TMX1 are operating, output control is performed according to the setting of each timer. See **(1) PWM output operation (TMXn-only mode, single output)** and **(2) PWM output operation (TMXn-only mode, dual output)** for the setting of TXnCRm.

Figure 6-31. Example of Register Settings for PWM Output Operation (TMX0 and TMX1 Synchronous Start Mode) (1/2)

(a) 16-bit timer Xn operation control register 0



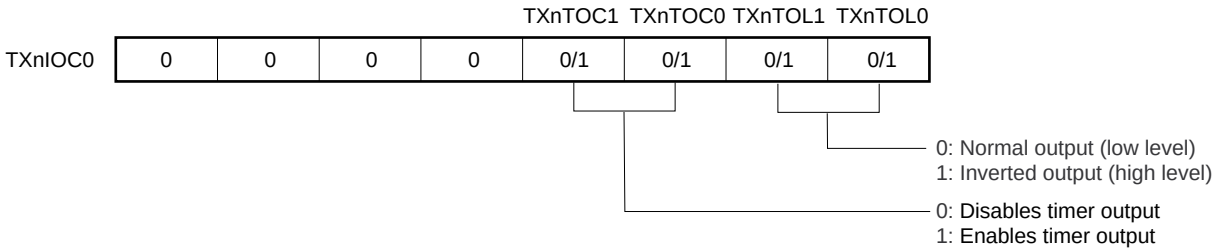
(b) 16-bit timer Xn operation control register 1



Remark n = 0, 1

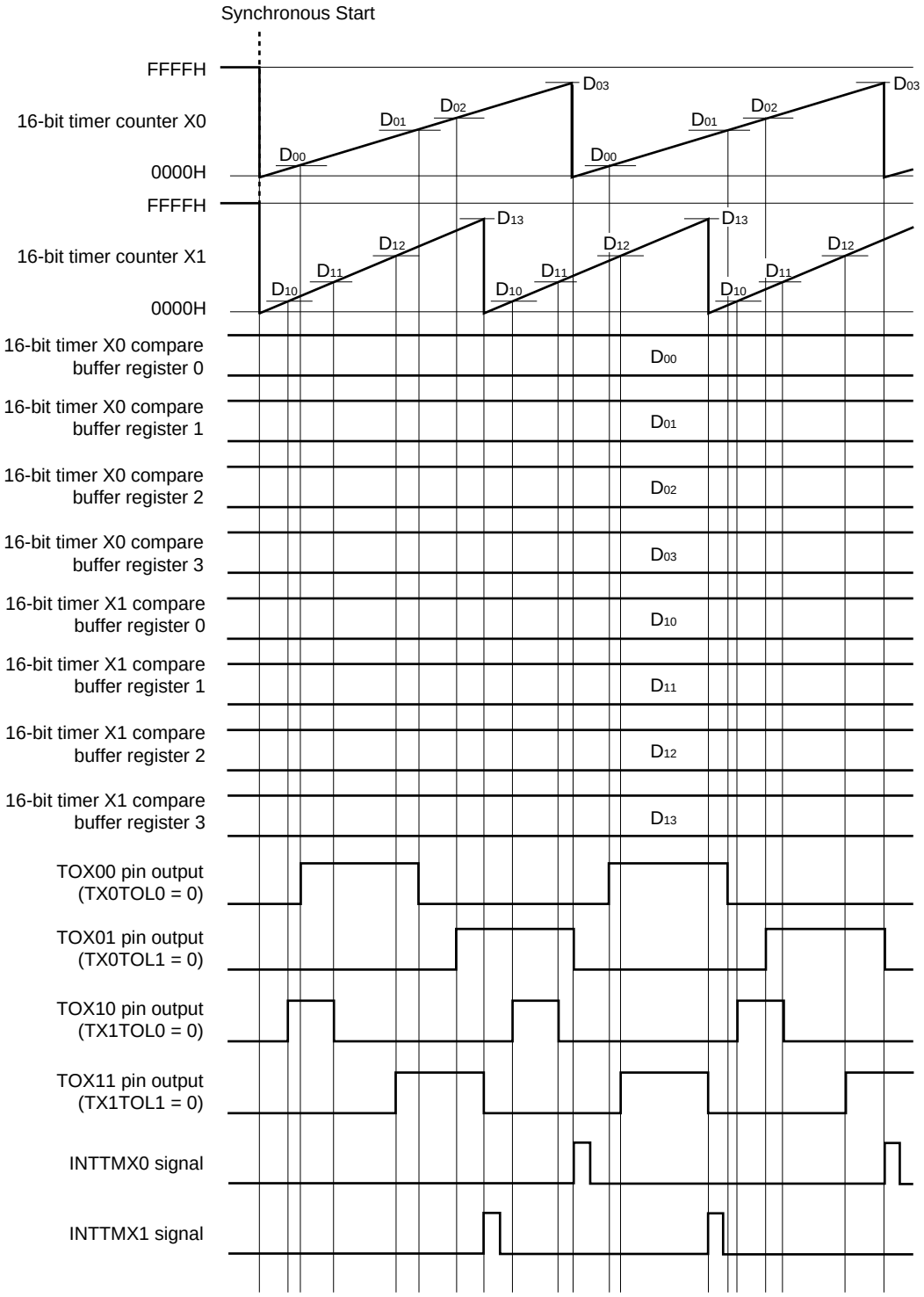
Figure 6-31. Example of Register Settings for PWM Output Operation (TMX0 and TMX1 Synchronous Start Mode) (2/2)

(c) 16-bit timer Xn output control register 0



Remark n = 0, 1

Figure 6-32. PWM Output Timing (Synchronous Start mode, TMX0 dual output, TMX1 dual output)



(4) PWM output operation (TMX0 and TMX1 synchronous start/clear mode) (78K0/FB2-L only)

Output cycles from the two timer outputs of TMX0 and TMX1 (up to 4 outputs) are synchronized.
Setting bit 7 (TX0TMC) of TX0CTL1 to 1 starts PWM output.
Setting bit 7 (TX0TMC) of TX0CTL1 to 0 stops PWM output.

TXnCRm can be rewritten while the timers are operating, and the duty and the pulse cycle can be changed.

(a) TMX0 single output, TMX1 single output or dual output

- Pulse cycle and duty of TOX00
  - Pulse cycle = (Set value of TX0CR1 + 1) × Count clock cycle
  - Duty = (Set value of TX0CR1 − Set value of TX0CR0) / (Set value of TX0CR1 + 1)
- Pulse cycle and duty of TOX10
  - Pulse cycle = (Set value of TX0CR1 + 1) × Count clock cycle
  - Duty = (Set value of TX1CR1 − Set value of TX1CR0) / (Set value of TX0CR1 + 1)
- Pulse cycle and duty of TOX11[Note 1]
  - Pulse cycle = (Set value of TX0CR1 + 1) × Count clock cycle
  - Duty = (Set value of TX1CR3 − Set value of TX1CR2) / (Set value of TX0CR1 + 1)

When changing TXnCRm, be sure to rewrite TX0CR1 after rewriting the registers other than TX0CR1. The registers other than TX0CR1 that are not to be changed do not have to be rewritten. If one of the registers other than TX0CR1 needs to be changed but TX0CR1 does not need to be changed, be sure to write the same value to TX0CR1.

The output is changed when the INTTMX0 interrupt is generated immediately after TX0CR1 is written. However, if TX0CR1 is rewritten during the clock[Note 2] cycle in which the INTTMX0 interrupt was generated, or during the previous two clock[Note 2] cycles, the output will be changed when the INTTMX0 interrupt subsequent to this INTTMX0 interrupt is generated. Note also that if TXnCRm is written with a different value in the period between when TX0CR1 is written and when the output changes, the output will be changed to this different value, not the originally specified value.

To specify PWM output from TOX00, set TX0CR0 and TX0CR1 to a value in the following range:
0000H ≤ TX0CR0 ≤ TX0CR1 ≤ FFFFH
If TX0CR0 = TX0CR1 is specified, the output will be set to the default status (fixed).

To specify PWM output from TOX10, set TX1CR0 and TX1CR1 to a value in the following range:
0000H ≤ TX1CR0 ≤ TX1CR1 ≤ TX0CR1
If TX1CR0 = TX1CR1 is specified, the output will be set to the default status (fixed).

To specify PWM output from TOX11[Note 1], set TX1CR2 and TX1CR3 to a value in the following range:
0000H ≤ TX1CR2 ≤ TX1CR3 ≤ TX0CR1
If TX1CR2 = TX1CR3 is specified, the output will be set to the default status (fixed).

Notes 1. TMX1 dual output only
2. Count clock of 16-bit timer X0

(b) TMX0 dual output, TMX1 single output or dual output

- Pulse cycle and duty of TOX00
  - Pulse cycle = (Set value of TX0CR3 + 1) × Count clock cycle
  - Duty = (Set value of TX0CR1 − Set value of TX0CR0) / (Set value of TX0CR3 + 1)
- Pulse cycle and duty of TOX01
  - Pulse cycle = (Set value of TX0CR3 + 1) × Count clock cycle
  - Duty = (Set value of TX0CR3 − Set value of TX0CR2) / (Set value of TX0CR3 + 1)
- Pulse cycle and duty of TOX10
  - Pulse cycle = (Set value of TX0CR3 + 1) × Count clock cycle
  - Duty = (Set value of TX1CR1 − Set value of TX1CR0) / (Set value of TX0CR3 + 1)
- Pulse cycle and duty of TOX11[Note 1]
  - Pulse cycle = (Set value of TX0CR3 + 1) × Count clock cycle
  - Duty = (Set value of TX1CR3 − Set value of TX1CR2) / (Set value of TX0CR3 + 1)

When changing TXnCRm, be sure to rewrite TX0CR3 after rewriting the registers other than TX0CR3. The registers other than TX0CR3 that are not to be changed do not have to be rewritten. If one of the registers other than TX0CR3 needs to be changed but TX0CR3 does not need to be changed, be sure to write the same value to TX0CR3.

The output is changed when the INTTMX0 interrupt is generated immediately after TX0CR3 is written. However, if TX0CR3 is rewritten during the clock[Note 2] cycle in which the INTTMX0 interrupt was generated, or during the previous two clock[Note 2] cycles, the output will be changed when the INTTMX0 interrupt subsequent to this INTTMX0 interrupt is generated. Note also that if TXnCRm is written with a different value in the period between when TX0CR3 is written and when the output changes, the output will be changed to this different value, not the originally specified value.

To specify PWM output from TOX00, set TX0CR0 and TX0CR1 to a value in the following range:

0000H ≤ TX0CR0 ≤ TX0CR1 ≤ TX0CR3

If TX0CR0 = TX0CR1 is specified, the output will be set to the default status (fixed).

To specify PWM output from TOX01, set TX0CR2 and TX0CR3 to a value in the following range:

0000H ≤ TX0CR2 ≤ TX0CR3 ≤ FFFFH

If TX0CR2 = TX0CR3 is specified, the output will be set to the default status (fixed).

To specify PWM output from TOX10, set TX1CR0 and TX1CR1 to a value in the following range:

0000H ≤ TX1CR0 ≤ TX1CR1 ≤ TX0CR3

If TX1CR0 = TX1CR1 is specified, the output will be set to the default status (fixed).

To specify PWM output from TOX11[Note 1], set TX1CR2 and TX1CR3 to a value in the following range:

0000H ≤ TX1CR2 ≤ TX1CR3 ≤ TX0CR3

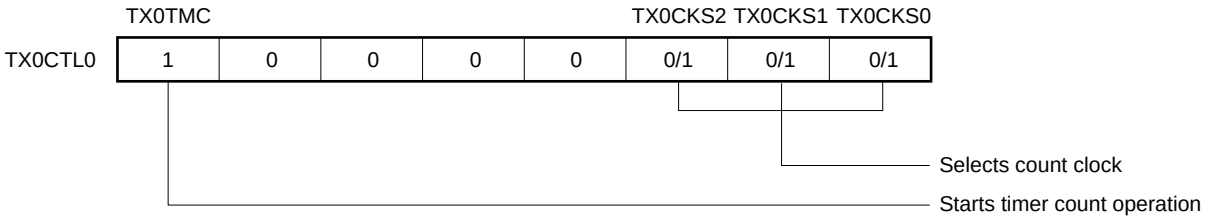
If TX1CR2 = TX1CR3 is specified, the output will be set to the default status (fixed).

Notes 1. TMX1 dual output only

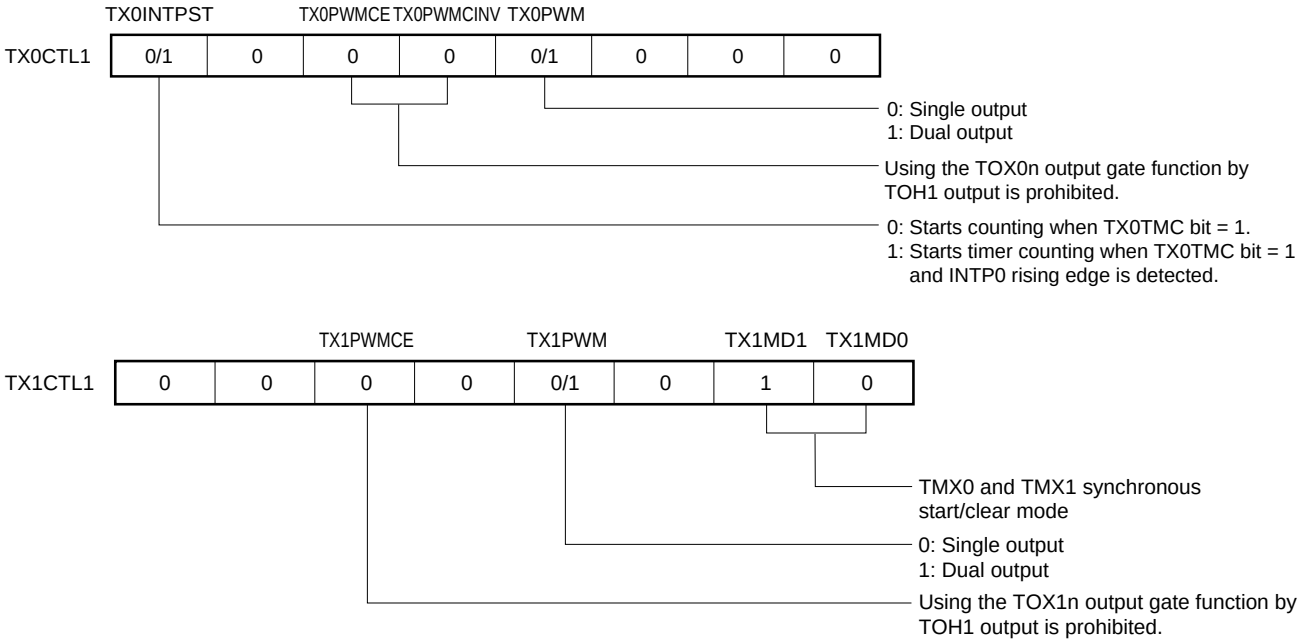
2. Count clock of 16-bit timer X0

Figure 6-33. Example of Register Settings for PWM Output Operation (TMX0 and TMX1 synchronous start/clear mode)

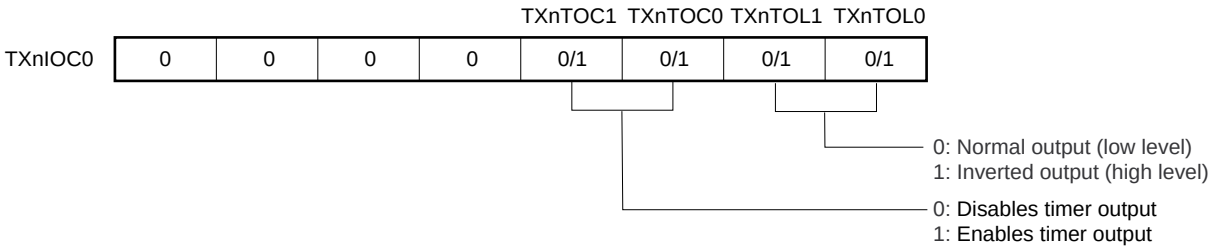
(a) 16-bit timer X0 operation control register 0



(b) 16-bit timer Xn operation control register 1

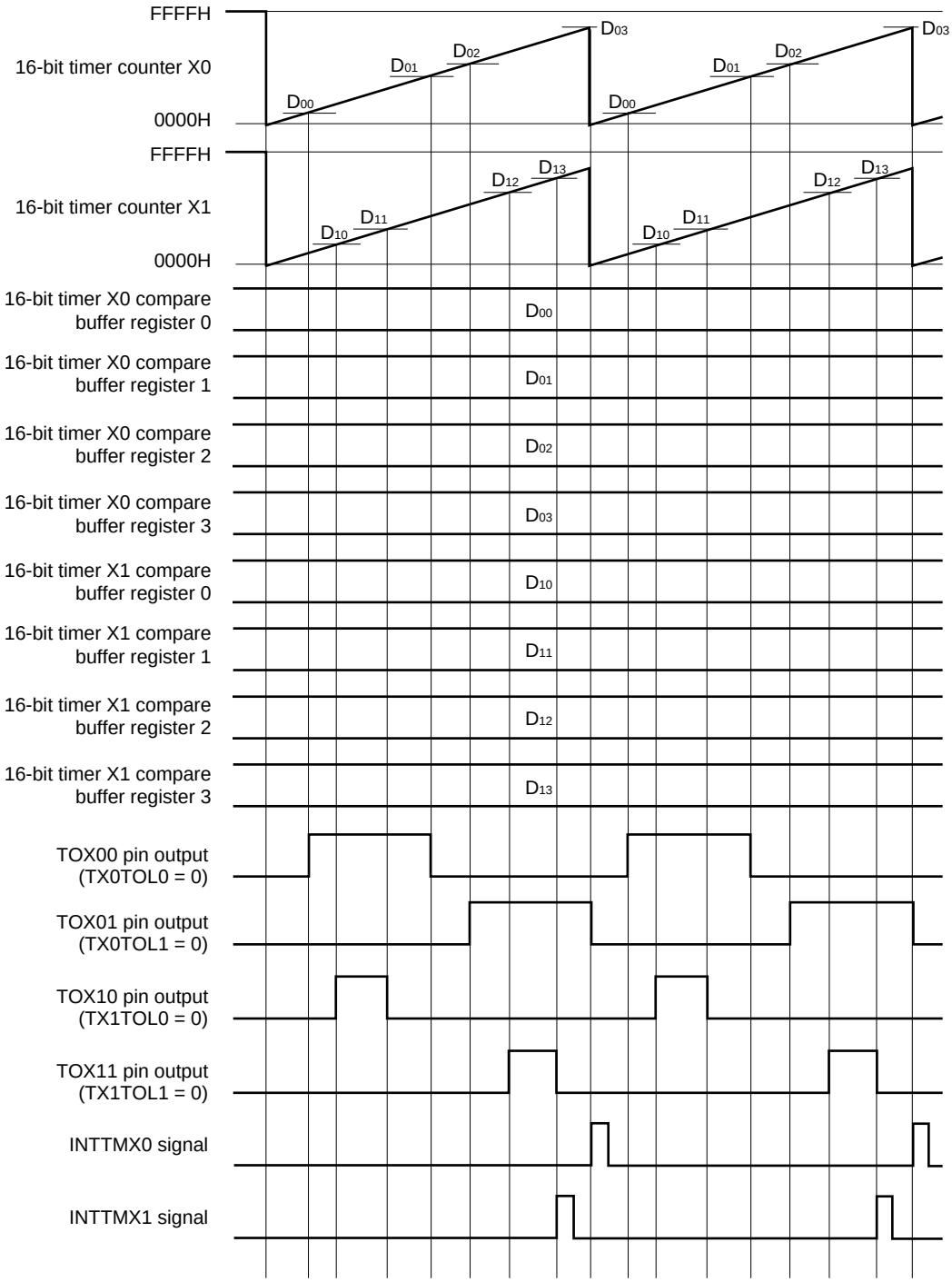


(c) 16-bit timer Xn output control register 0



Remark n = 0, 1

Figure 6-34. PWM Output Timing (Synchronous start/clear mode, TMX0 dual output, TMX1 dual output)



(5) PWM output operation (PWM output from TOX0n when TOH1 output is at high level)

A square wave is output from the TOX0n pin by combining 8-bit timer H1 and 16-bit timer X0, only when the TOH1 output is at high level.

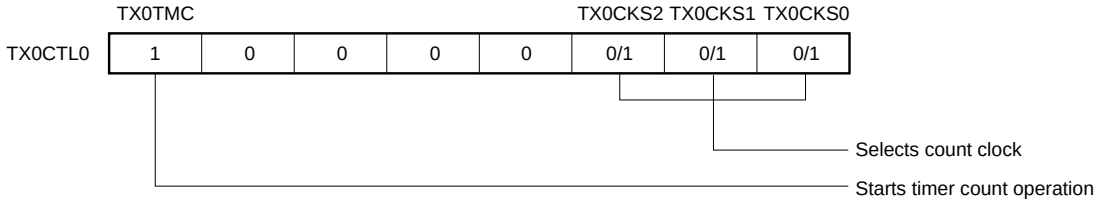
See **(1) PWM output operation (single output)** through **(4) PWM output operation (TMX0 and TMX1 synchronous start mode)** for the setting of outputting a square wave.

Remarks 1. n = 0, 1

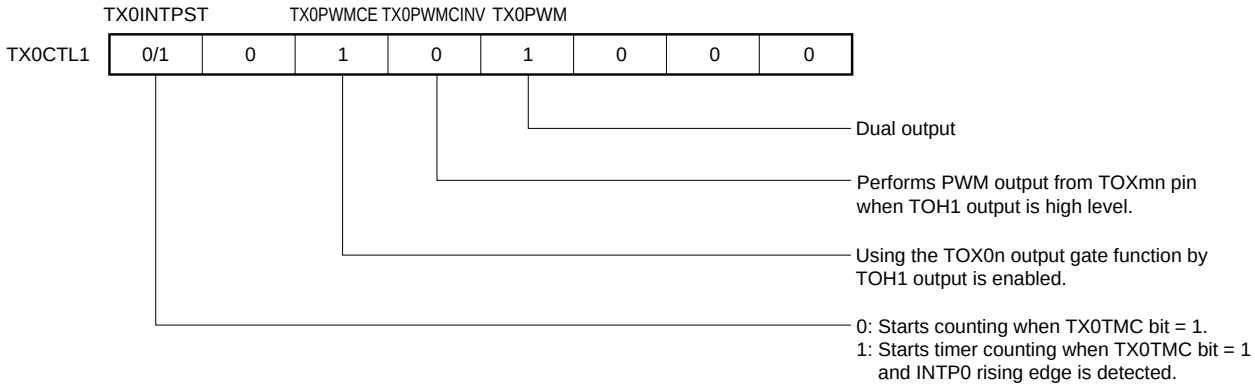
2. For the setting of TOH1 output, see **CHAPTER 9 8-BIT TIMER H1.**

Figure 6-35. Example of Register Settings for PWM Output Operation (TMX0-Only Operation (Dual Output), PWM Output from TOX00 and TOX01 When TOH1 Output Is at High Level)

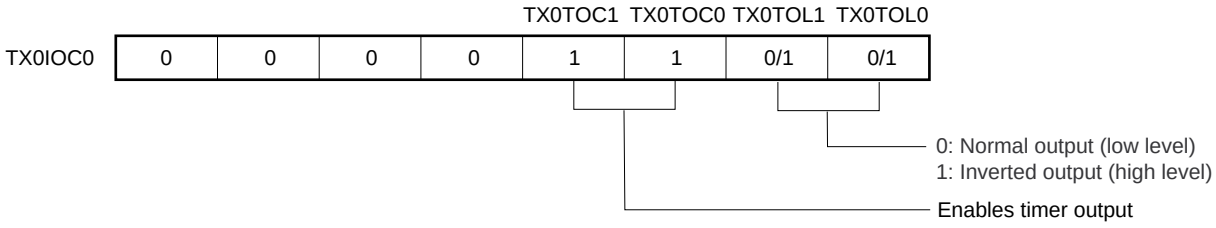
(a) 16-bit timer X0 operation control register 0



(b) 16-bit timer X0 operation control register 1

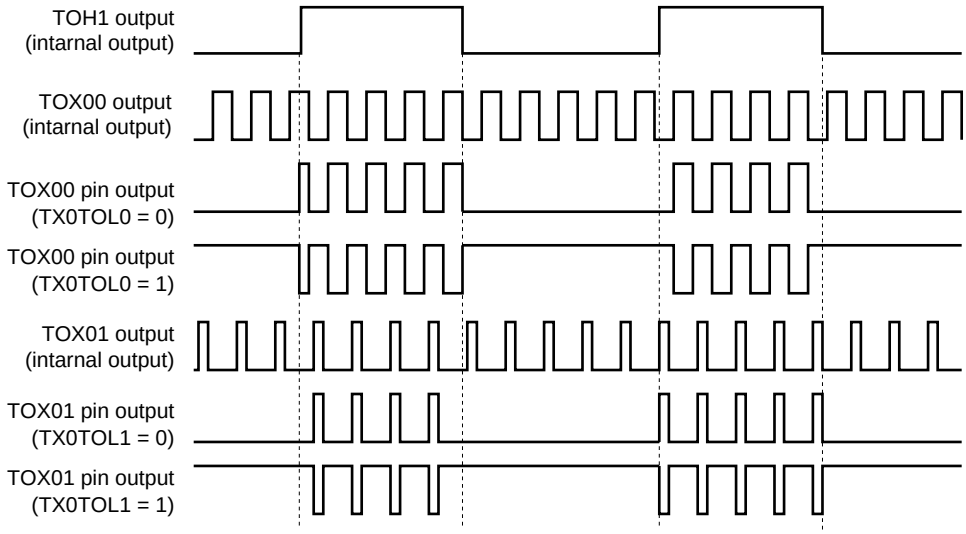


(c) 16-bit timer X0 output control register 0



Remark n = 0, 1, mn = 00, 01, 10, 11

Figure 6-36. PWM Output Timing (TMX0-Only Operation (Dual Output), PWM Output from TOX00 and TOX01 When TOH1 Output Is at High Level)



(6) PWM output operation (PWM output from TOX0n when TOH1 output is at low level)

A square wave is output from the TOX0n pin by combining 8-bit timer H1 and 16-bit timer X0, only when the TOH1 output is at low level.

See **(1) PWM output operation (single output)** through **(4) PWM output operation (TMX0 and TMX1 synchronous start mode)** for the setting of outputting a square wave.

Remarks 1. n = 0, 1

2. For the setting of TOH1 output, see **CHAPTER 9 8-BIT TIMER H1**.

3. The above functions cannot be used in the 78K0/FY2-L. In the 78K0/FY2-L, only the interval timer can be used.

Figure 6-37. Example of Register Settings for PWM Output Operation (TMX0-Only Operation (Dual Output), PWM Output from TOX00 and TOX01 When TOH1 Output Is at Low Level) (1/2)

(a) 16-bit timer X0 operation control register 0

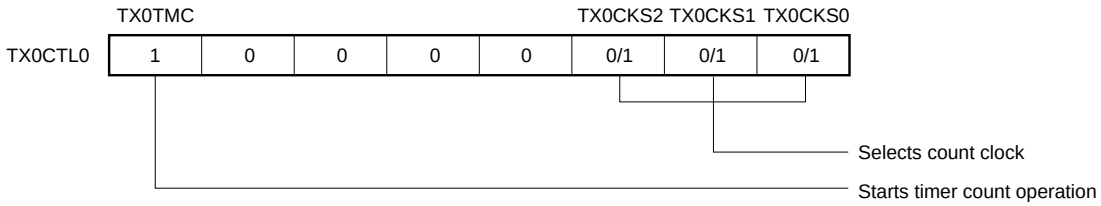
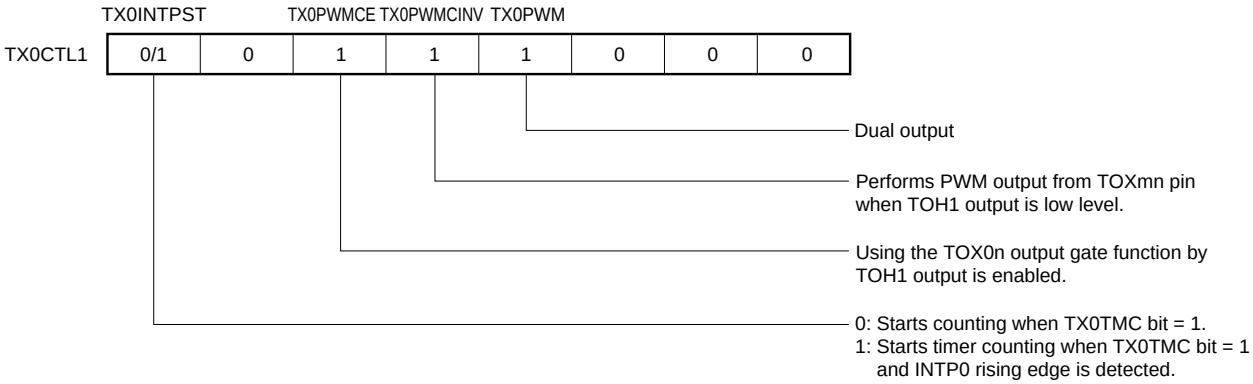
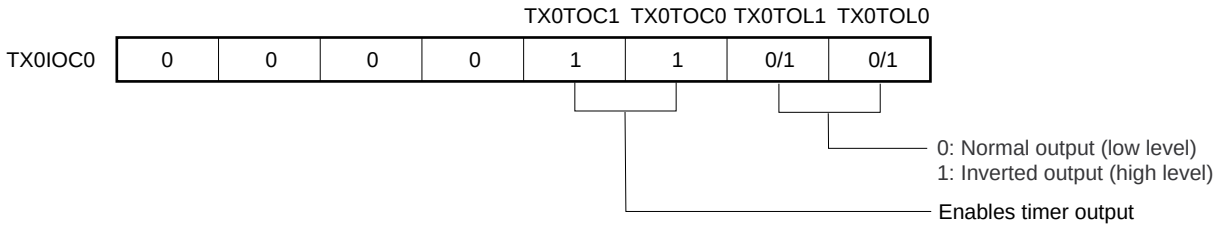


Figure 6-37. Example of Register Settings for PWM Output Operation (TMX0-Only Operation (Dual Output), PWM Output from TOX00 and TOX01 When TOH1 Output Is at Low Level) (2/2)

(b) 16-bit timer X0 operation control register 1

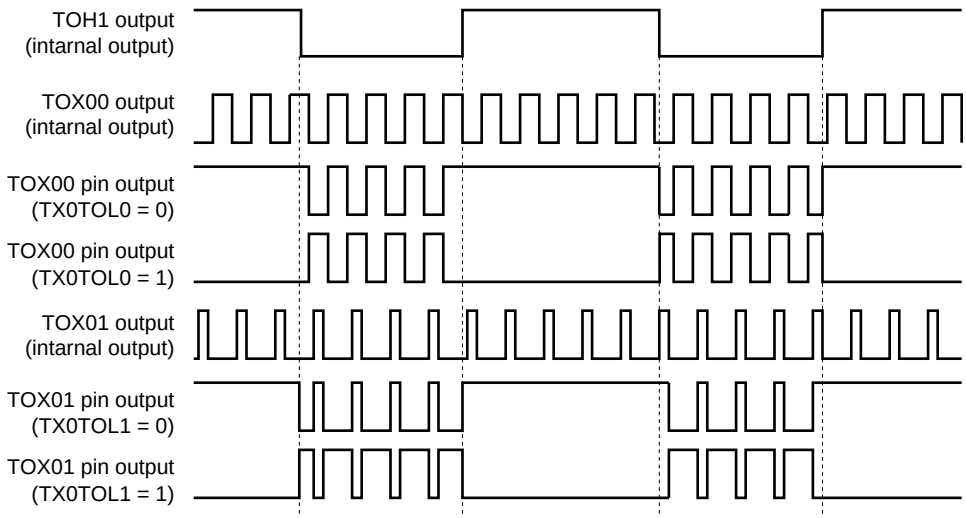


(c) 16-bit timer X0 output control register 0



Remark n = 0, 1, mn = 00, 01, 10, 11

Figure 6-38. PWM Output Timing (TMX0-Only Operation (Dual Output), PWM Output from TOX00 and TOX01 When TOH1 Output Is at Low Level)



(7) PWM output operation (PWM output from TOX1n when TOH1 output is at high level) (78K0/FB2-L only)

A square wave is output from the TOX1n pin by combining 8-bit timer H1 and 16-bit timer X1, only when the TOH1 output is at high level.

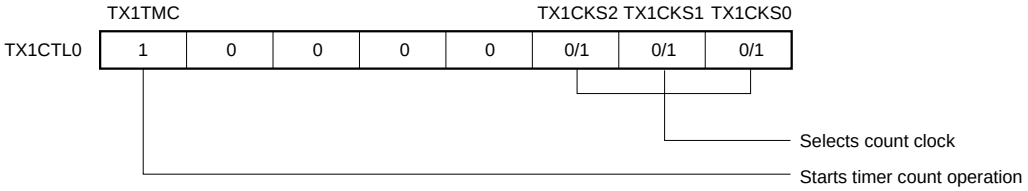
See **(1) PWM output operation (single output)** through **(4) PWM output operation (TMX0 and TMX1 synchronous start mode)** for the setting of outputting a square wave.

Remarks 1. n = 0, 1

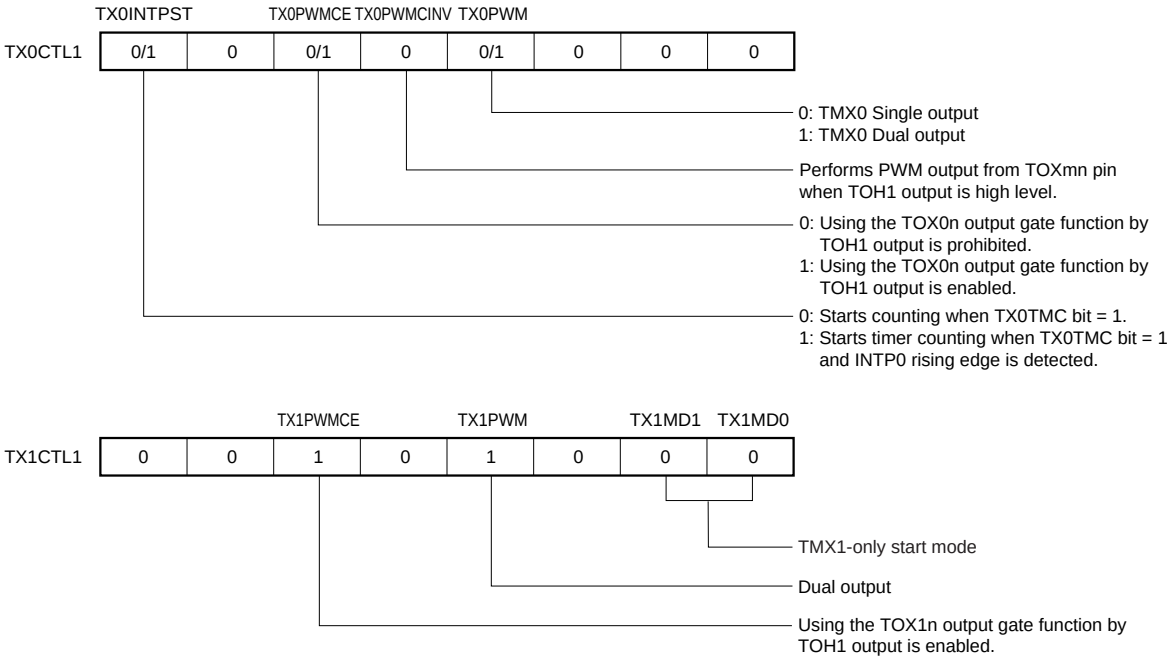
2. For the setting of TOH1 output, see **CHAPTER 9 8-BIT TIMER H1**.

Figure 6-39. Example of Register Settings for PWM Output Operation (TMX1-Only Operation (Dual Output), PWM Output from TOX10 and TOX11 When TOH1 Output Is at High Level) (1/2)

(a) 16-bit timer X1 operation control register 0



(b) 16-bit timer Xn operation control register 1



Remark n = 0, 1, mn = 00, 01, 10, 11

Figure 6-39. Example of Register Settings for PWM Output Operation (TMX1-Only Operation (Dual Output), PWM Output from TOX10 and TOX11 When TOH1 Output Is at High Level) (2/2)

(c) 16-bit timer X1 output control register 0

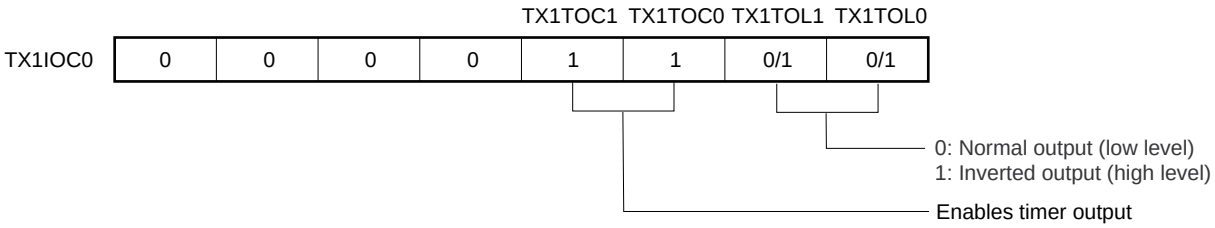
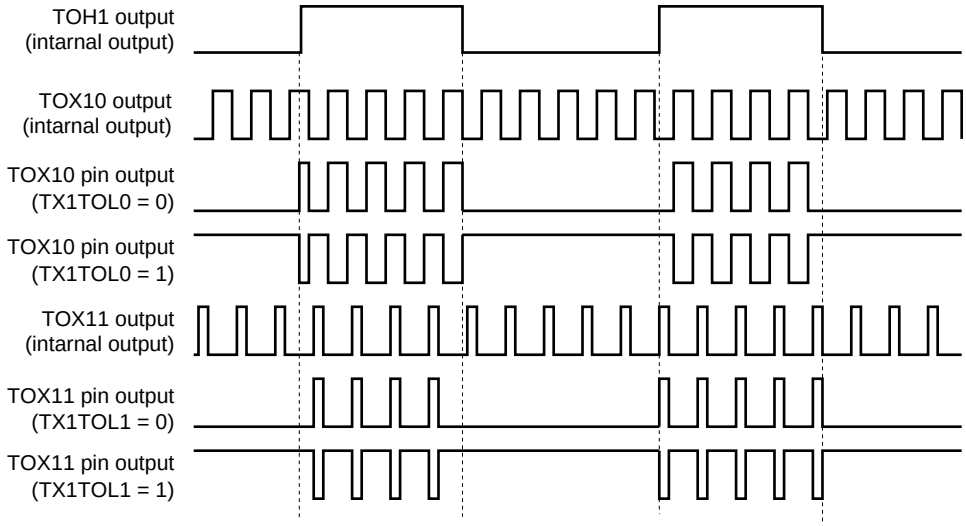


Figure 6-40. PWM Output Timing (TMX1-Only Operation (Dual Output), PWM Output from TOX10 and TOX11 When TOH1 Output Is at High Level)



(8) PWM output operation (PWM output from TOX00, TOX01, TOX10, and TOX11 when TOH1 output is at high level) (78K0/FB2-L only)

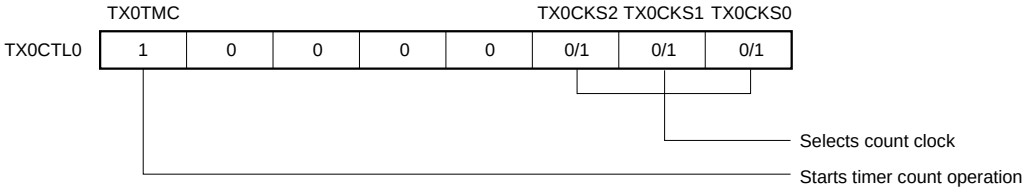
A square wave is output from the TOX00, TOX01, TOX10, and TOX11 pins by combining 8-bit timer H1 and 16-bit timers X0 and X1, only when the TOH1 output is at high level.

See **(1) PWM output operation (single output)** through **(4) PWM output operation (TMX0 and TMX1 synchronous start mode)** for the setting of outputting a square wave.

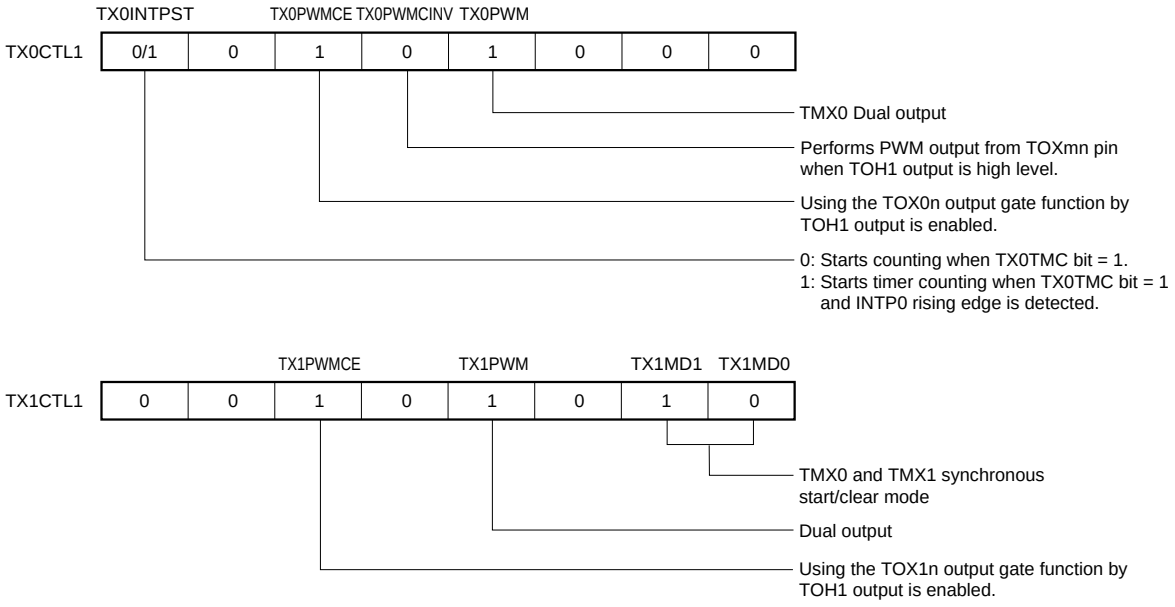
Remark For the setting of TOH1 output, see **CHAPTER 9 8-BIT TIMER H1.**

Figure 6-41. Example of Register Settings for PWM Output Operation (TMX0 and TMX1 Synchronous Start/Clear Mode, PWM Output from TOX00, TOX01, TOX10, and TOX11 When TOH1 Output Is at High Level) (1/2)

(a) 16-bit timer X0 operation control register 0



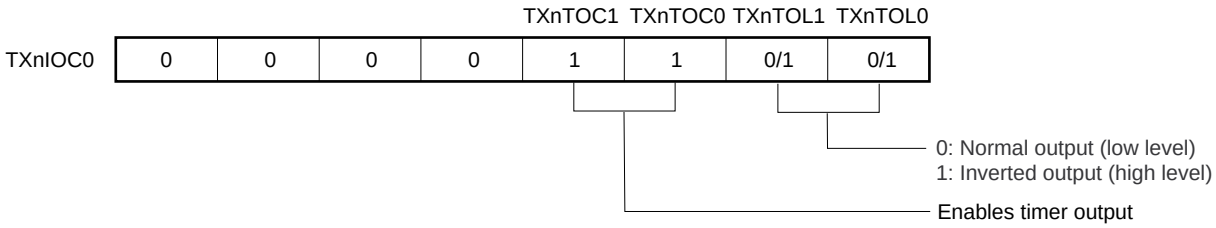
(b) 16-bit timer Xn operation control register 1



Remark n = 0, 1, mn = 00, 01, 10, 11

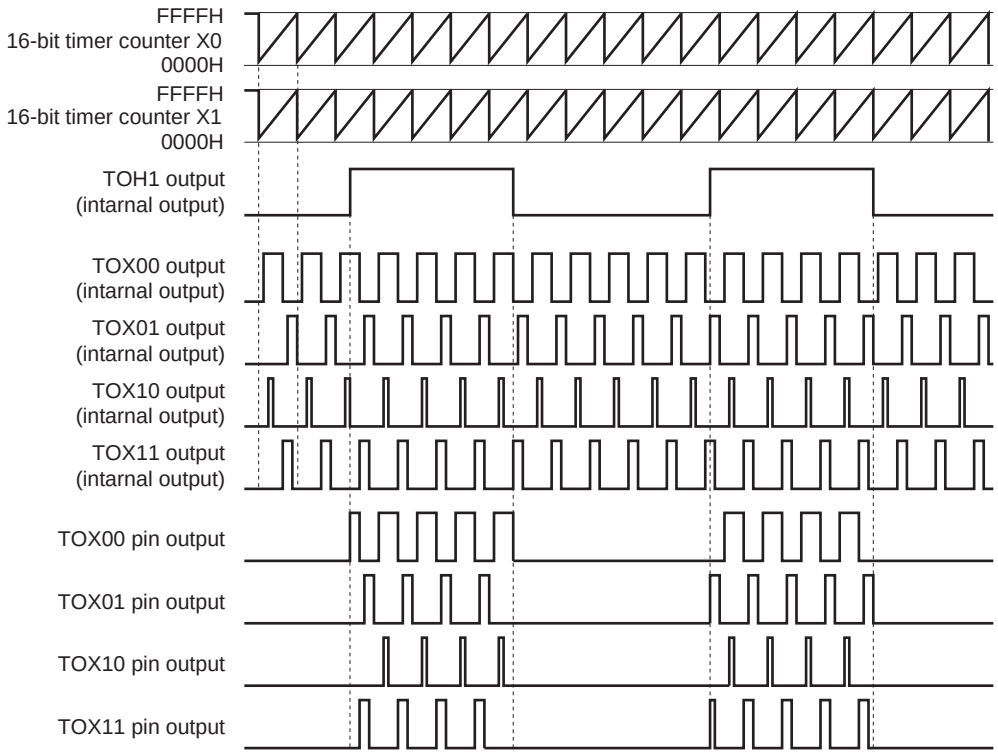
Figure 6-41. Example of Register Settings for PWM Output Operation (TMX0 and TMX1 Synchronous Start/Clear Mode, PWM Output from TOX00, TOX01, TOX10, and TOX11 When TOH1 Output Is at High Level) (2/2)

(c) 16-bit timer Xn output control register 0



Remark n = 0, 1

Figure 6-42. PWM Output Timing (TMX0 and TMX1 Synchronous Start/Clear Mode, PWM Output from TOX00, TOX01, TOX10, and TOX11 When TOH1 Output Is at High Level)



6.6 Interlocking Function with Comparator or INTP0

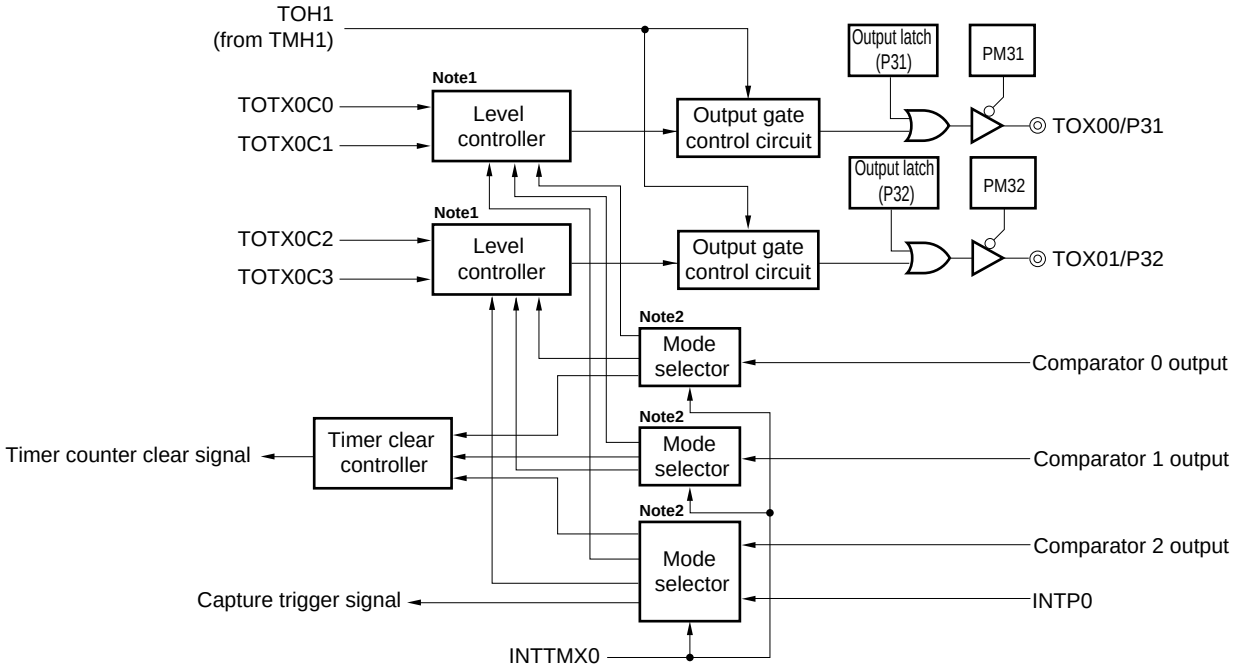
16-bit timers X0 and X1 can control PWM waveforms by interlocking with the output of comparators 0 to 2 or the INTP0 input signal, without involving the CPU.

TMX0 and TMX1, comparators 0 to 2, and INTP0 can be combined as follows.

- 16-bit timer X0 (TMX0-only operation mode, synchronous start mode[Note]): CMP0, CMP1, CMP2, INTP0
- 16-bit timer X1 (TMX1-only operation mode, synchronous start mode)[Note]: CMP0, CMP1
- 16-bit timers X0 and X1 (synchronous start/clear mode)[Note]: CMP2, INTP0

Note Available only in 78K0/FB2-L

Figure 6-43. Block Diagram of 16-bit Timer X0 Output Configuration



Notes 1. Timer output is controlled by the level controller according to the value of the compare register and the mode selector output.

2. Resetting timer output (interlocking modes 1 and 3) and clearing the timer counter (interlocking modes 1 and 2) are controlled by the mode selector according to the comparator output or INTP0 input.

Figure 6-44. Block Diagram of 16-Bit Timer X1 Output Configuration (78K0/FB2-L only)

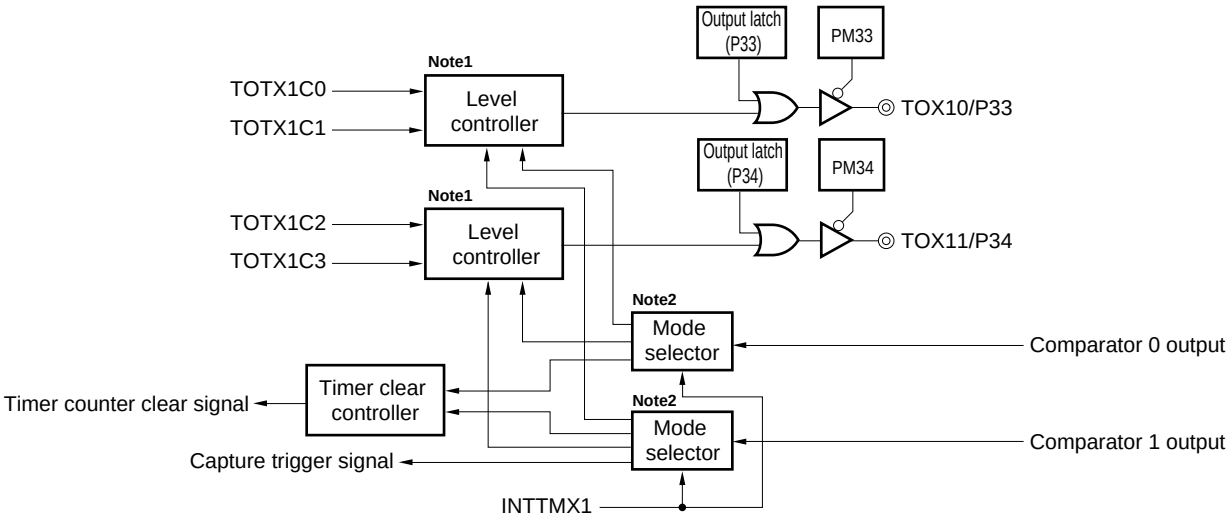
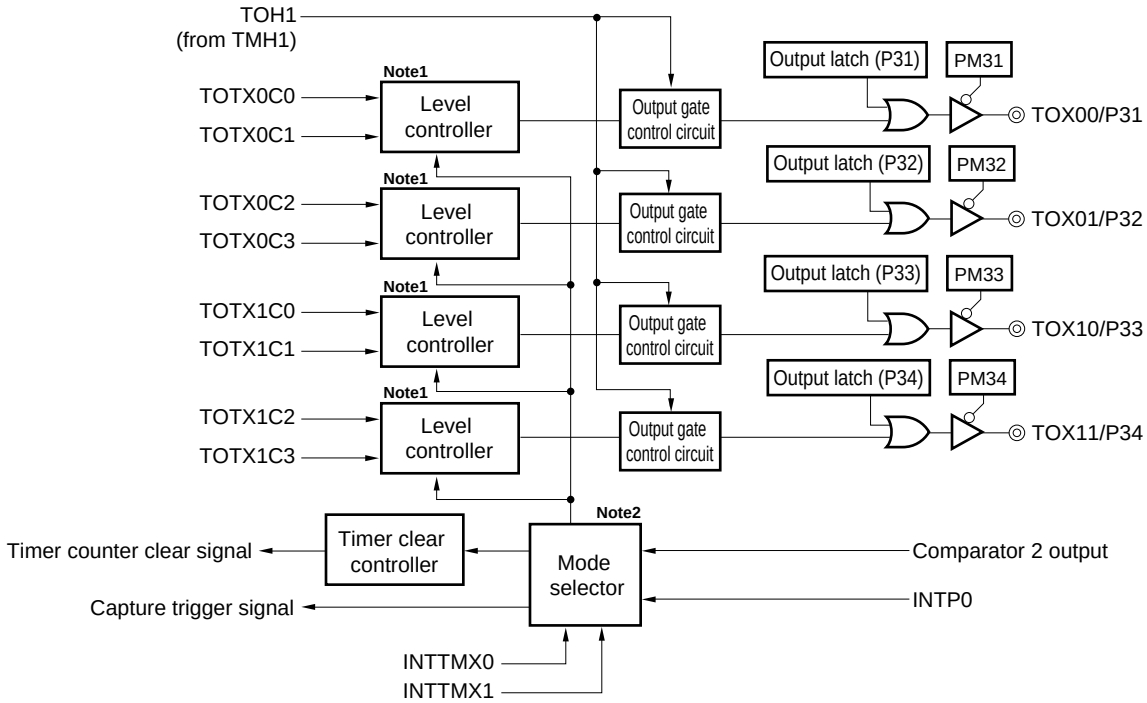


Figure 6-45. Block Diagram of 16-Bit Timers X0 and X1 Output Configuration (78K0/FB2-L only)



Notes 1. Timer output is controlled by the level controller according to the value of the compare register and the mode selector output.

2. Resetting timer output (interlocking modes 1 and 3) and clearing the timer counter (interlocking modes 1 and 2) are controlled by the mode selector according to the comparator output or INTP0 input.

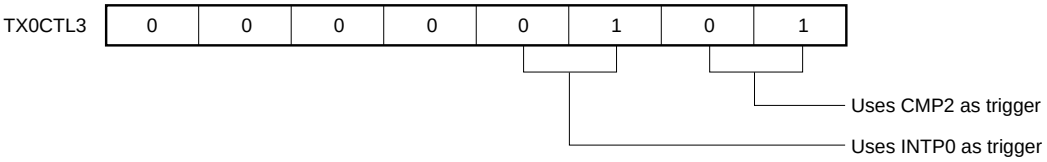
The modes controlled by comparator output or external interrupt input (INTP0) are described below.

(1) Interlocking mode 1 (timer reset mode)

This mode sets the output of the corresponding timer to the reset state while the output of comparators 0 to 2 or INTP0 input is at high level, and restarts the timer when the detection signal is stopped.

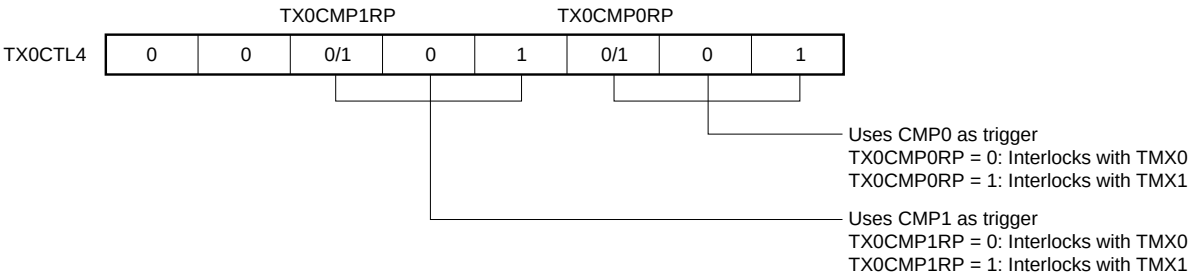
Figure 6-46 Example of Register Settings for Interlocking Mode 1 (Timer Reset Mode)

- Using CMP2 and INTP0 as triggers



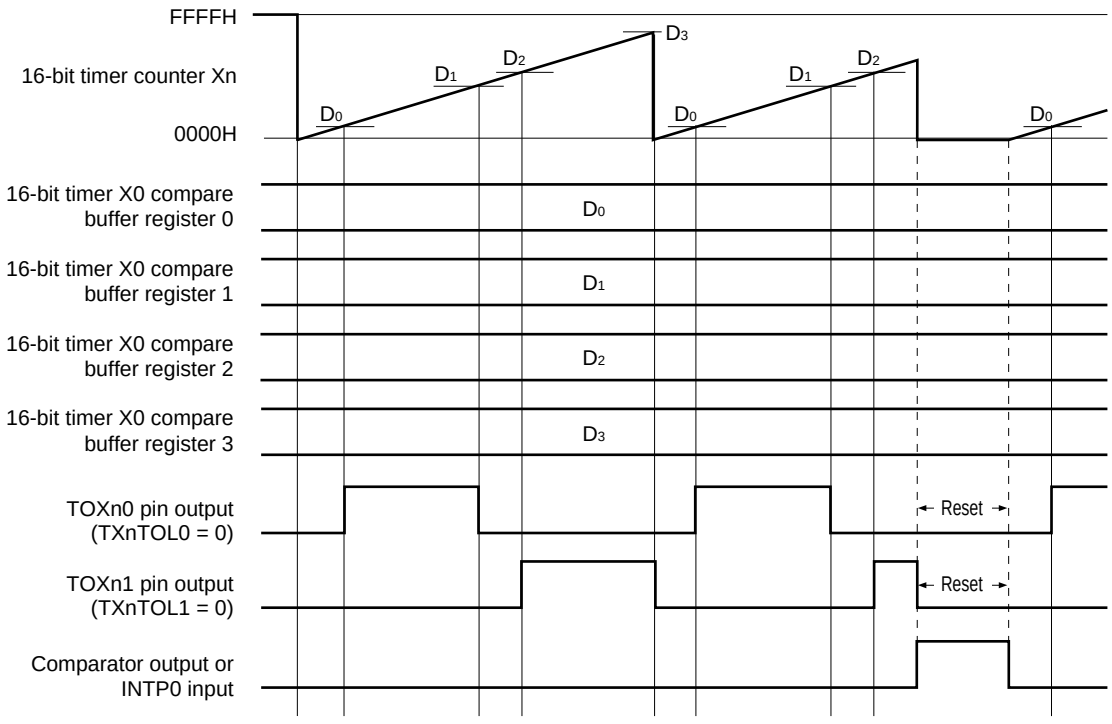
Remark When interlocking the timers with either CMP2 or INTP0, set all bits of CMP2 or INTP0, whichever is not used, to 0.

- Using CMP0 and CMP1 as triggers



Remark When interlocking the timers with either CMP0 or CMP1, set all bits of CMP0 or CMP1, whichever is not used, to 0.

Figure 6-47 Timing of Interlocking Mode 1 (Timer Reset Mode)



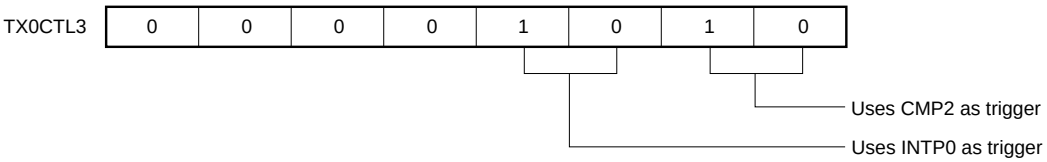
Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(2) Interlocking mode 2 (timer restart mode)

This mode restarts the corresponding timer when the rising edge of the comparators 0 to 2 outputs or the INTP0 input is detected.

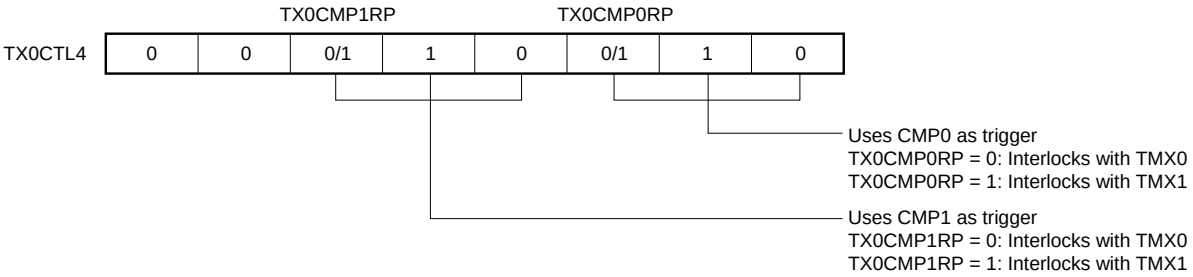
Figure 6-48 Example of Register Settings for Interlocking Mode 2 (Timer Restart Mode)

- Using CMP2 and INTP0 as triggers



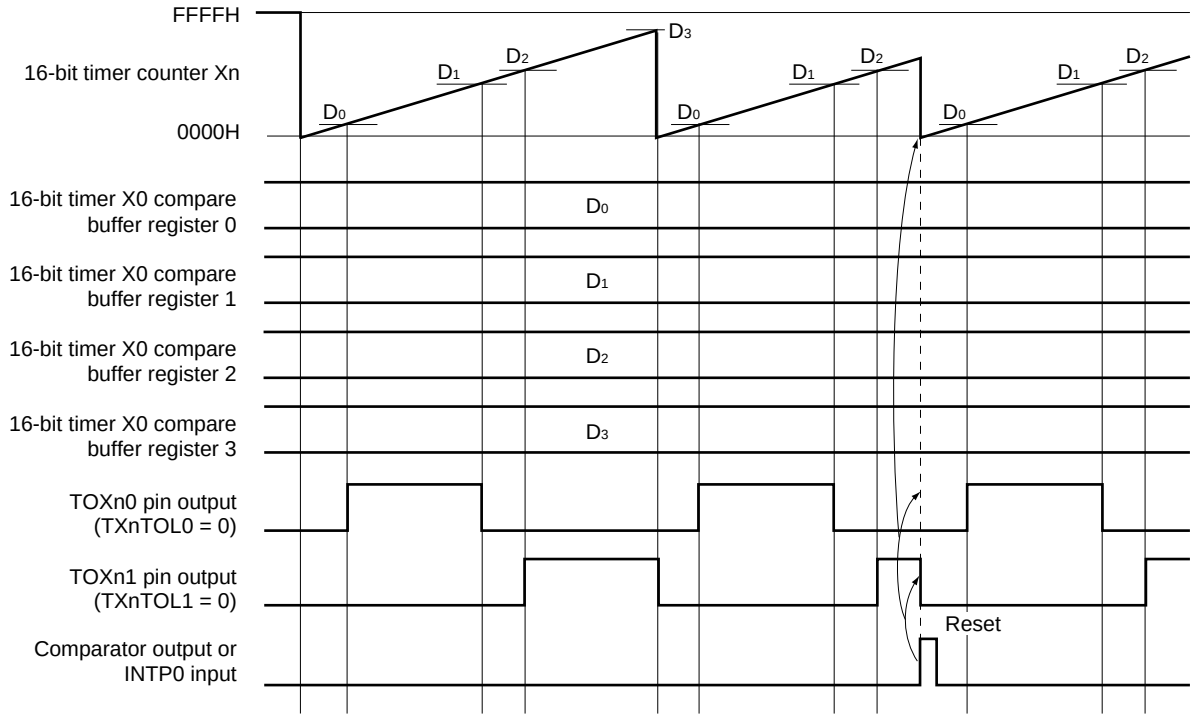
Remark When interlocking the timers with either CMP2 or INTP0, set all bits of CMP2 or INTP0, whichever is not used, to 0.

- Using CMP0 and CMP1 as triggers



Remark When interlocking the timers with either CMP0 or CMP1, set all bits of CMP0 or CMP1, whichever is not used, to 0.

Figure 6-49 Timing of Interlocking Mode 2 (Timer Restart Mode)



Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

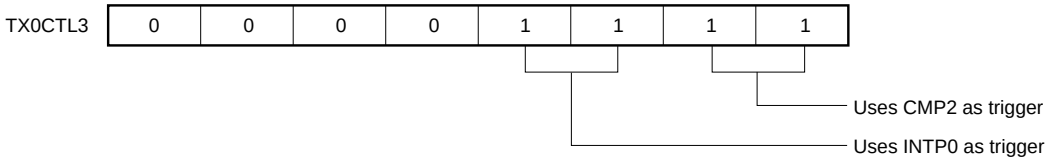
(3) Interlocking mode 3 (timer output reset mode)

This mode sets the output of the corresponding timer to the reset state from when the rising edge of the comparators 0 to 2 outputs or the INTP0 input is detected until the next interrupt is generated.

Caution Do not set to interlocking mode 3 when in TMX0 and TMX1 synchronous start/clear mode.

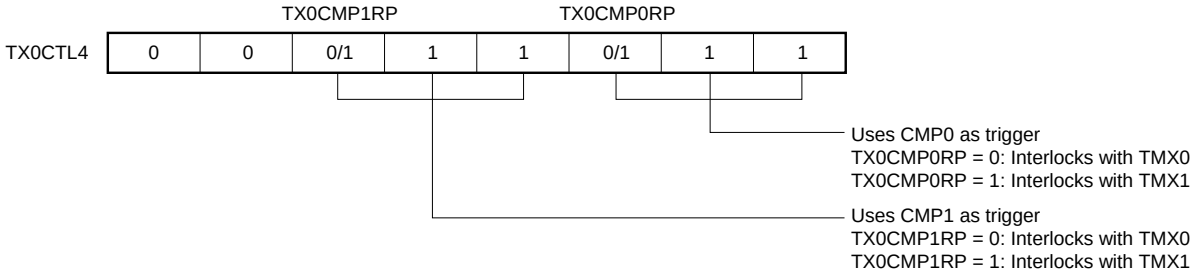
Figure 6-50 Example of Register Settings for Interlocking Mode 3 (Timer Output Reset Mode)

- Using CMP2 and INTP0 as triggers



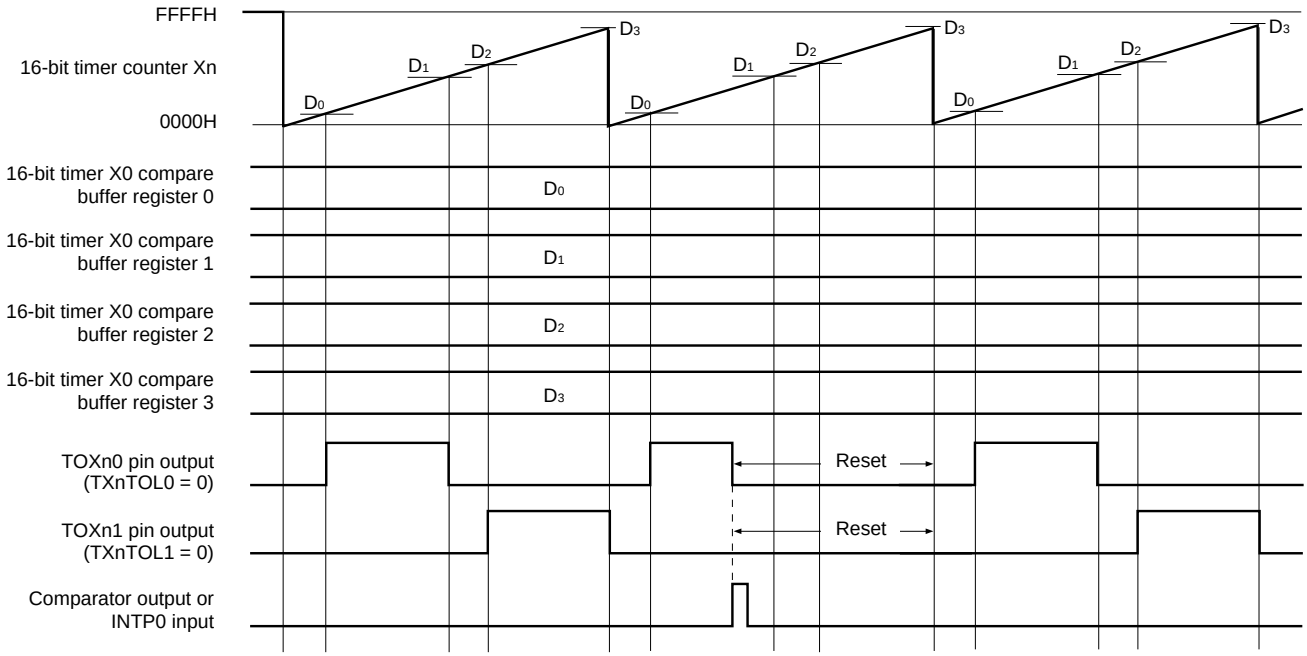
Remark When interlocking the timers with either CMP2 or INTP0, set all bits of CMP2 or INTP0, whichever is not used, to 0.

- Using CMP0 and CMP1 as triggers



Remark When interlocking the timers with either CMP0 or CMP1, set all bits of CMP0 or CMP1, whichever is not used, to 0.

Figure 6-51 Timing of Interlocking Mode 3 (Timer Output Reset Mode)



Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(4) Priority when multiple interlocking modes occur

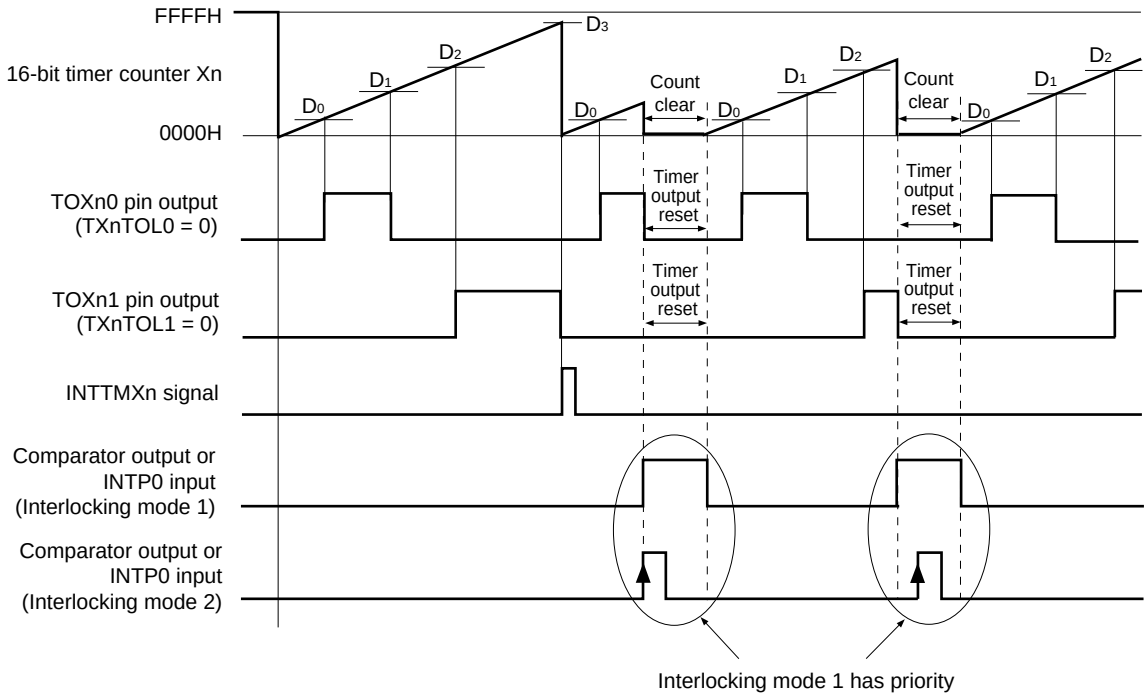
If multiple interlocking modes have been specified for TMX0 or TMX1, the priority order is as follows:

Interlocking mode 1 > Interlocking mode 2 > Interlocking mode 3

(a) Priority of interlocking modes 1 and 2

If interlocking modes 1 and 2 occur at the same time, or if interlocking mode 2 occur while the timer is being reset in interlocking mode 1, interlocking mode 1 has priority and interlocking mode 2 is invalid.

Figure 6-52. Priority When Interlocking Mode 1 Conflicts with Interlocking Mode 2

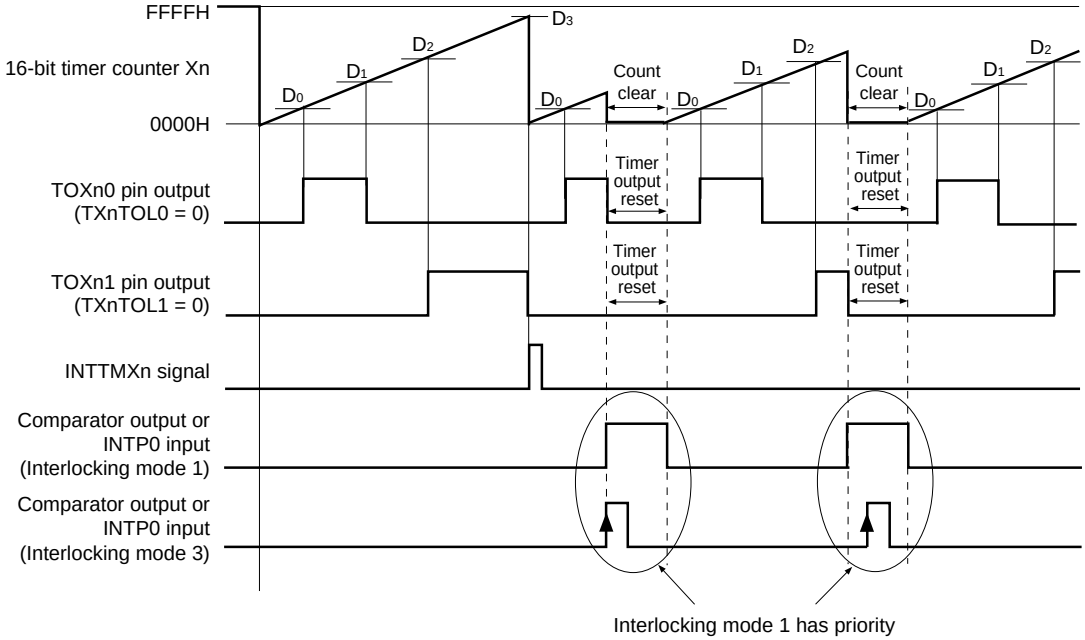


Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(b) Priority of interlocking modes 1 and 3

If interlocking modes 1 and 3 occur at the same time, or if interlocking mode 3 occur while the timer is being reset in interlocking mode 1, interlocking mode 1 has priority and interlocking mode 3 is invalid.

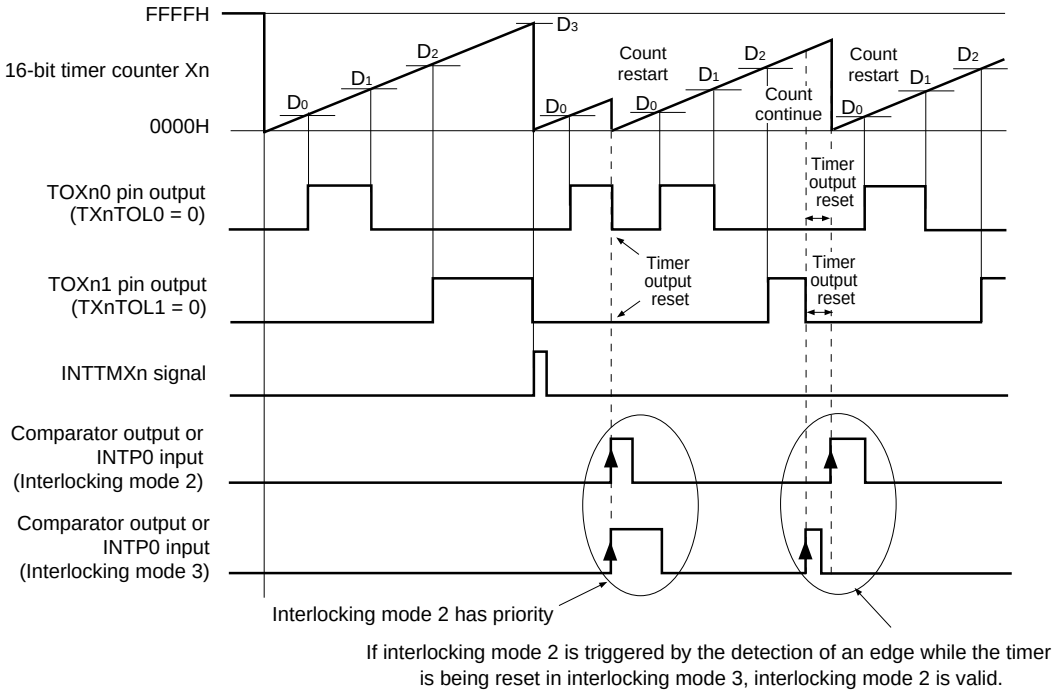
Figure 6-53. Priority When Interlocking Mode 1 Conflicts with Interlocking Mode 3



(c) Priority of interlocking modes 2 and 3

If interlocking modes 2 and 3 occur at the same time, interlocking mode 2 has priority and interlocking mode 3 is invalid.

Figure 6-54. Priority When Interlocking Mode 2 Conflicts with Interlocking Mode 3



Remark n = 0 : 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

6.7 High-Impedance Output Control Function

The high-impedance output control function changes the outputs of 16-bit timers X0 and X1 to a high-impedance state at the generation of an external interrupt input (INTP0) or comparator output (INTCMP0 to INTCMP2) and executes functions such as the PWM output emergency stop function.

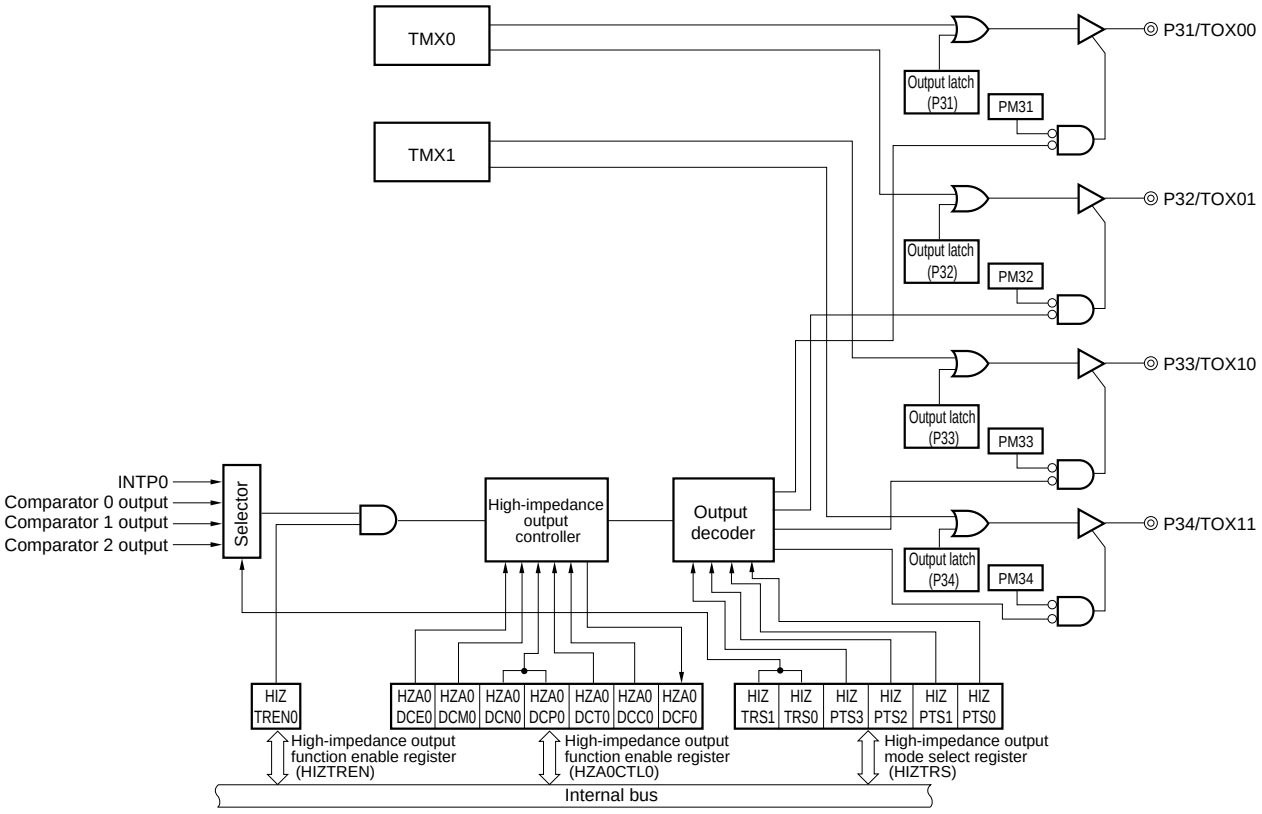
6.7.1 Configuration of high-impedance output controller

The high-impedance output control circuit includes the following hardware.

Table 6-4. Configuration of High-Impedance Output Controller

Item	Configuration
Control registers	High-impedance output function enable register (HIZTREN) High-impedance output mode select register (HIZTRS) High-impedance output function control register 0 (HZA0CTL0)

Figure 6-55 Block Diagram of High-Impedance Output Controller



6.7.2 Registers controlling high-impedance output controller

Registers used to control the high-impedance output controller are shown below.

- High-impedance output function enable register (HIZTREN)
- High-impedance output mode select register (HIZTRS)
- High-impedance output function control register 0 (HZA0CTL0)

(1) High-impedance output function enable register (HIZTREN)

HIZTREN is a register that enables/disables the input of the trigger signal used for controlling high-impedance output.

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

Reset signal generation clears HIZTREN to 00H.

Figure 6-56 Format of High-Impedance Output Function Enable Register (HIZTREN)

Address: FF6EH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
HIZTREN	HIZTREN0	0	0	0	0	0	0	0

HIZTREN0	Input control of trigger signal used for high-impedance output control
0	Disables input
1	Enables input

(2) High-impedance output mode select register (HIZTRS)

HIZTRS is a register that selects the signal to be used as the high-impedance control trigger and the pin to be set to the high-impedance output state.

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

Reset signal generation clears HIZTRS to 00H.

Figure 6-57 Format of High-Impedance Output Mode Select Register (HIZTRS)

Address: FF6FH After reset: 00H R/W

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
HIZTRS	HIZTRS1	HIZTRS0	0	0	HIZPTS3Note	HIZPTS2Note	HIZPTS1	HIZPTS0

HIZTRS1	HIZTRS0	Selection of signal to be used as trigger
0	0	INTP0
0	1	Comparator 0 output
1	0	Comparator 1 output
1	1	Comparator 2 output

HIZPTS3Note	P34/TOX11 pin control
0	Normal output
1	Can be used as high-impedance output

HIZPTS2Note	P33/TOX10 pin control
0	Normal output
1	Can be used as high-impedance output

HIZPTS1	P32/TOX01 pin control
0	Normal output
1	Can be used as high-impedance output

HIZPTS0	P31/TOX00 pin control
0	Normal output
1	Can be used as high-impedance output

Note 78K0/FB2-L only.

(3) High-impedance output function control register 0 (HZA0CTL0)

HZA0CTL0 is a register that controls the high-impedance state of the output buffers.

HZA0CTL0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears HZA0CTL0 to 00H.

Figure 6-58 Format of High-Impedance Output Function Control Register 0 (HZA0CTL0) (1/2)

Address: FF78H After reset: 00H R/W[Note 1]

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
HZA0CTL0	HZA0DCE0	HZA0DCM0	HZA0DCN0	HZA0DCP0	HZA0DCT0	HZA0DCC0	0	HZA0DCF0

HZA0DCE0	High-impedance output control
0	Disables high-impedance output control operation. Output of target pin can be performed.
1	Enables high-impedance output control operation

HZA0DCM0 Note 2	Condition of releasing high-impedance state by HZA0DCC0 bit
0	HZA0DCC0 bit setting is valid regardless of signal to be used.
1	HZA0DCC0 bit setting is invalid while signal to be used holds abnormal detection level (active level).

HZA0DCN0 Note 2	HZA0DCP0 Note 2	Valid-edge specification during high-impedance control[Notes 3, 4]
0	0	No valid edge (disables setting (1) HZA0DCF0 bit by INTP0 or comparator output).
0	1	Enables rising edge of signal to be used (abnormal detection at rising edge).
1	0	Enables falling edge of signal to be used (abnormal detection at falling edge).
1	1	Setting prohibited

Notes 1. Bit 0 is read only.

2. The HZA0DCM0, HZA0DCN0, and HZA0DCP0 bits should be rewritten when HZA0DCE0 = 0.

3. For the valid edge for INTP0, and INTCMP0 to INTCMP2, see **18.3 (5) External interrupt rising edge enable registers 0, 1 (EGPCTL0, EGPCTL1), external interrupt falling edge enable registers 0, 1 (EGNCTL0, EGNCTL1)**.

4. High-impedance output control will be performed if a valid edge is detected after the high-impedance output control operation is enabled (HZA0DCE0 = 1). Consequently, if the trigger signal to be used is at the active level when the operation is enabled, high-impedance output control will not be performed.

Figure 6-58 Format of High-Impedance Output Function Control Register 0 (HZA0CTL0) (2/2)

HZA0DCT0 **Notes 1 to 4**	High-impedance output trigger bit
0	Does not operate.
1	Sets target pin to high-impedance output state and sets (1) HZA0DCF0 bit.

HZA0DCC0 **Notes 2 to 6**	High-impedance output control clear bit
0	Does not operate.
1	Enables output of target pin and clears (0) HZA0DCF0 bit.

HZA0DCF0	High-impedance output state flag
0	Output of target pin is enabled. • Clears (0) when HZA0DCE0 = 0. • Clears (0) when HZA0DCC0 = 1.
1	Target pin is in high-impedance output state. • Sets (1) when HZA0DCT0 = 1. • Sets (1) when edge indicating abnormality is detected from trigger signal (when valid edge set by HZA0DCN0 and HZA0DCP0 bits is detected).

Notes 1. This is invalid even if the HZA0DCT0 bit is set (1), when an edge indicating an abnormality is detected from the trigger signal to be used (when the valid edge set by the HZA0DCN0 and HZA0DCP0 bits is detected).

2. This is invalid even if the HZA0DCT0 and HZA0DCC0 bits are set (1) when HZA0DCE0 = 0.

3. 0 is read from the HZA0DCT0 and HZA0DCC0 bits.

4. Do not simultaneously set the HZA0DCT0 and HZA0DCC0 bits to 1.

5. If the HZA0DCC0 bit is set (1) when HZA0DCM0 = 0, the high-impedance output state of the target pin will be released regardless of the signal to be used as the trigger.

6. This is invalid even if the HZA0DCC0 bit is set (1), when an edge indicating an abnormality is detected from the signal to be used as the trigger (when the valid edge set by the HZA0DCN0 and HZA0DCP0 bits is detected) when HZA0DCM0 = 1.

6.7.3 High-impedance output control circuit setting procedure

(1) Transitioning to the high-impedance output state by detecting the valid edge of the INPT0 input or the output of comparators 0 to 2

<1> Set the HIZTRS1, HSTRS0, and HIZPTS3 to HIZPTS0 bits (select the trigger source and the high-impedance target pin).
<2> Set the HZA0DCM0, HZA0DCN0, and HZA0DCP0 bits (select the high-impedance state release condition and valid edge).
<3> Set (1) the HIZTREN0 bit (enable the input of the trigger signal).
<4> Set (1) the HZA0DCE0 bit (enable the high-impedance output control operation).

(2) Changing the settings after enabling the high-impedance output control operation

<1> Clear (0) the HZA0DCE0 bit (disable the high-impedance output control operation).
<2> Clear (0) the HIZTREN0 bit (disable the input of the trigger signal).
<3> Change the HIZTRS1, HSTRS0, and HIZPTS3 to HIZPTS0 bits (change the trigger source and high-impedance target pin).
<4> Set the HZA0DCM0, HZA0DCN0, and HZA0DCP0 bits (select the high-impedance state release condition and valid edge).
<5> Set (1) the HIZTREN0 bit (re-enable the input of the trigger signal).
<6> Set (1) the HZA0DCE0 bit (re-enable the high-impedance output control operation).

(3) Restarting output while a pin is in the high-impedance output state

The impedance output state can be released only when the valid edge of the trigger to be used is detected when HZA0DCM0 = 1 and the HZA0DCC0 bit is set (1) after the trigger signal is set to the inactive level.
When HZA0DCM0 = 0, the impedance output state is released when the HZA0DCC0 bit is set (1), regardless of the level of the trigger signal.

(a) HZA0DCM0 = 1

<1> Set (1) the HZA0DCC0 bit (generate the instruction signal releasing the high-impedance state).
<2> Read the HZA0DCF0 bit and check the flag status.
<3> If HZA0DCF0 = 1, return to operation <1>. The level of the trigger signal must be checked.
If HZA0DCF0 = 0, pin output can be performed.

(b) HZA0DCM0 = 0

- Set (1) the HZA0DCC0 bit (the instruction signal releasing the high-impedance state is generated -> pin output can be performed).

Caution If the trigger signal is at the inactive level and the timing of setting (1) the HZA0DCC0 bit and the timing at which the valid edge of the trigger signal is detected match, setting (1) the HZA0DCC0 bit may be given precedence.

(4) Transitioning to the high-impedance output state by using the HZA0DCT0 bit

To set the pin to the high-impedance output state by using the HZA0DCT0 bit, the HZA0DCT0 bit must be set (1) while the trigger signal is in the inactive-level state. When not using the trigger signal (HZA0DCN0 = HZA0DCP0 = 0), however, the high-impedance output state can be entered by setting (1) the HZA0DCT0 bit.

(a) HZA0DCN0 = 0 and HZA0DCP0 = 1, or HZA0DCN0 = 1 and HZA0DCP0 = 0

<1> Set (1) the HZA0DCT0 bit (generate the high-impedance output instruction signal).

<2> Read the HZA0DCF0 bit and check the flag status.

<3> If HZA0DCF0 = 1, return to operation <1>. The level of the trigger signal must be checked.
If HZA0DCF0 = 0, the high-impedance output state is entered.

(b) HZA0DCN0 = HZA0DCP0 = 0

- Set (1) the HZA0DCT0 bit (the high-impedance output instruction signal is generated -> the high-impedance output state is entered).

CHAPTER 7 16-BIT TIMER/EVENT COUNTER 00

7.1 Functions of 16-bit Timer/Event Counter 00

16-bit timer/event counter 00 is mounted onto all 78K0/Fx2-L microcontroller products.

16-bit timer/event counter 00 has the following functions.

(1) Interval timer

16-bit timer/event counter 00 generates an interrupt request at the preset time interval.

(2) Square-wave output

16-bit timer/event counter 00 can output a square wave with any selected frequency.

(3) External event counter

16-bit timer/event counter 00 can measure the number of pulses of an externally input signal.

(4) One-shot pulse output

16-bit timer event counter 00 can output a one-shot pulse whose output pulse width can be set freely.

(5) PPG output

16-bit timer/event counter 00 can output a rectangular wave whose frequency and output pulse width can be set freely.

(6) Pulse width measurement

16-bit timer/event counter 00 can measure the pulse width of an externally input signal.

7.2 Configuration of 16-bit Timer/Event Counter 00

16-bit timer/event counter 00 includes the following hardware.

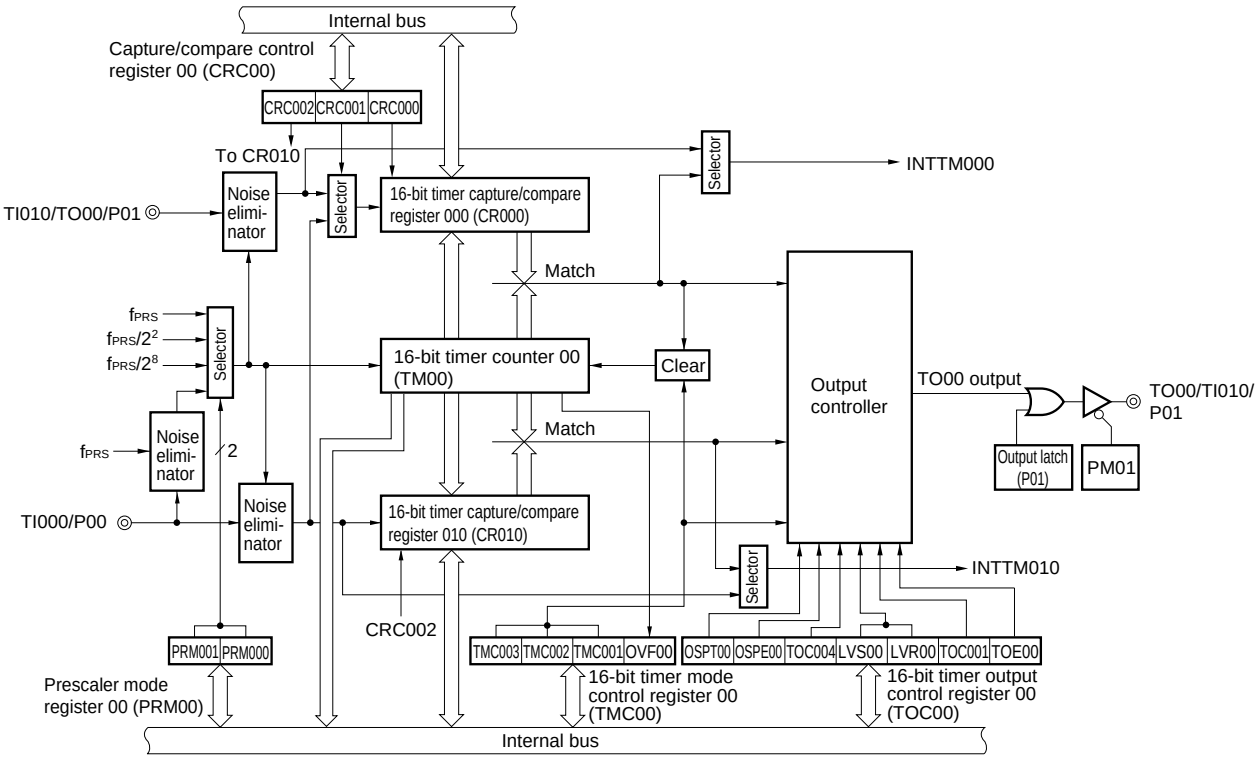
Table 7-1. Configuration of 16-bit Timer/Event Counter 00

Item	Configuration
Time/counter	16-bit timer counter 00 (TM00)
Register	16-bit timer capture/compare registers 000, 010 (CR000, CR010)
Timer input	TI000, TI010
Timer output	TO00, output controller
Control registers	16-bit timer mode control register 00 (TMC00) Capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Port alternate switch control register (MUXSEL)[Note] Port mode register 0 (PM0) Port register 0 (P0)

Note 78K0/FB2-L only.

Figure 7-1 shows a block diagram.

Figure 7-1. Block Diagram of 16-bit Timer/Event Counter 00



Remarks 1. 78K0/FY2-L, 78K0/FA2-L: TI000/INTP0/P00, TI010/TO00/P01
78K0/FB2-L: TI000/INTP0/P00, TI010/TO00/P01, <TI000>/P121/X1/TOOLC0/<INTP0>

2. Functions in angle brackets < > in **Remark 1** can be assigned by setting the input switch control register (MUXSEL).

Cautions 1. The valid edge of TI010 and timer output (TO00) cannot be used for the P01 pin at the same time. Select either of the functions.

2. If clearing of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) 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 TMC003 and TMC002 bits to 00, and then change the setting.

A value that has been once captured remains stored in CR000 unless the device is reset. If the mode has been changed to the comparison mode, be sure to set a comparison value.

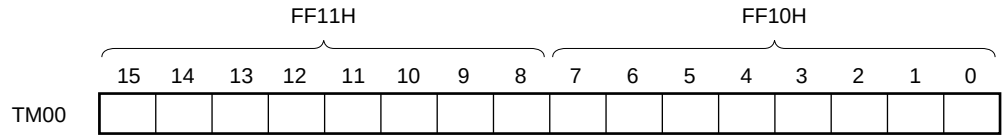
(1) 16-bit timer counter 00 (TM00)

TM00 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 7-2. Format of 16-bit Timer Counter 00 (TM00)

Address: FF10H, FF11H After reset: 0000H R



The count value of TM00 can be read by reading TM00 when the value of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) is other than 00. The value of TM00 is 0000H if it is read when TMC003 and TMC002 = 00.

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

- At reset signal generation
- If TMC003 and TMC002 are cleared to 00
- If the valid edge of the TI000 pin is input in the mode in which the clear & start occurs when inputting the valid edge to the TI000 pin
- If TM00 and CR000 match in the mode in which the clear & start occurs when TM00 and CR000 match
- OSPT00 is set to 1 in one-shot pulse output mode or the valid edge is input to the TI000 pin

Caution Even if TM00 is read, the value is not captured by CR010.

(2) 16-bit timer capture/compare register 000 (CR000), 16-bit timer capture/compare register 010 (CR010)

CR000 and CR010 are 16-bit registers that are used with a capture function or comparison function selected by using CRC00.

Change the value of CR000 while the timer is stopped (TMC003 and TMC002 = 00).

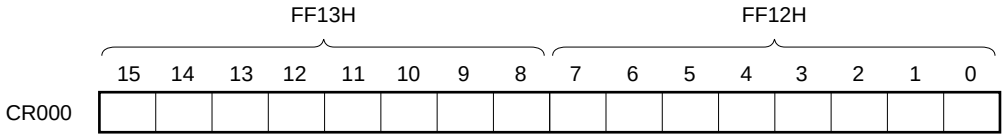
The value of CR010 can be changed during operation if the value has been set in a specific way. For details, refer to **7.5.1 Rewriting CR010 during TM00 operation**.

These registers can be read or written in 16-bit units.

Reset signal generation clears these registers to 0000H.

Figure 7-3. Format of 16-bit Timer Capture/Compare Register 000 (CR000)

Address: FF12H, FF13H After reset: 0000H R/W



(i) When CR000 is used as a compare register

The value set in CR000 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM000) is generated if they match. The value is held until CR000 is rewritten.

Caution CR000 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

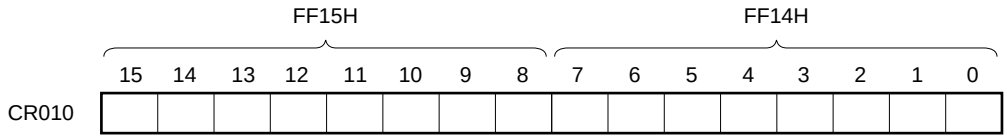
(ii) When CR000 is used as a capture register

The count value of TM00 is captured to CR000 when a capture trigger is input.

As the capture trigger, an edge of a phase reverse to that of the TI000 pin or the valid edge of the TI010 pin can be selected by using CRC00 or PRM00.

Figure 7-4. Format of 16-bit Timer Capture/Compare Register 010 (CR010)

Address: FF14H, FF15H After reset: 0000H R/W



(i) When CR010 is used as a compare register

The value set in CR010 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM010) is generated if they match.

Caution CR010 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 CR010 is used as a capture register

The count value of TM00 is captured to CR010 when a capture trigger is input.

It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 pin valid edge is set by PRM00.

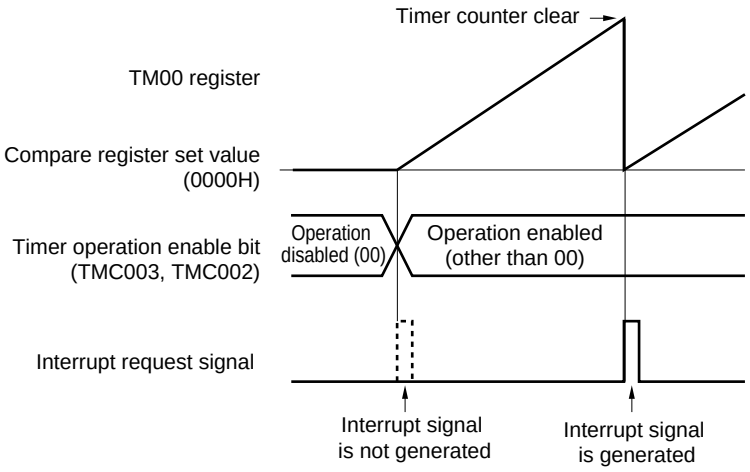
(iii) Setting range when CR000 or CR010 is used as a compare register

When CR000 or CR010 is used as a compare register, set it as shown below.

Operation	CR000 Register Setting Range	CR010 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 (INTTM010).
Operation as square-wave output		
Operation as external event counter		
Operation in the clear & start mode entered by TI000 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 (TM00 register) is changed from 0000H to 0001H.

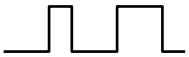
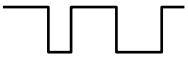
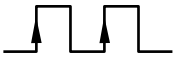
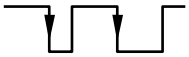
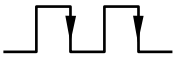
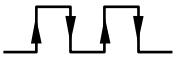
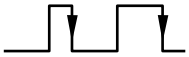
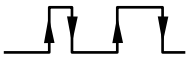
- When the timer counter is cleared due to overflow
- When the timer counter is cleared due to TI000 pin valid edge (when clear & start mode is entered by TI000 pin valid edge input)
- When the timer counter is cleared due to compare match (when clear & start mode is entered by match between TM00 and CR000 (CR000 = other than 0000H, CR010 = 0000H))



Remarks 1. N: CR000 register set value, M: CR010 register set value

2. For details of the operation enable bits (bits 3 and 2 (TMC003 and TMC002)), refer to **7.3 (1) 16-bit timer mode control register 00 (TMC00)**.

Table 7-2. Capture Operation of CR000 and CR010

<table>
<tr><th>External Input Signal / Capture Operation</th><th colspan="2">TI000 Pin Input  </th><th colspan="2">TI010 Pin Input  </th></tr>
<tr><td rowspan="5">Capture operation of CR000</td><td rowspan="4">CRC001 = 1
TI000 pin input
(reverse phase)
  </td><td>Set values of ES010 and ES000
Position of edge to be captured</td><td rowspan="4">CRC001 bit = 0
TI010 pin input
  </td><td>Set values of ES110 and ES100
Position of edge to be captured</td></tr>
<tr><td>01: Rising  </td><td>01: Rising  </td></tr>
<tr><td>00: Falling  </td><td>00: Falling  </td></tr>
<tr><td>11: Both edges
(cannot be captured)</td><td>11: Both edges  </td></tr>
<tr><td>Interrupt signal</td><td>INTTM000 signal is not generated even if value is captured.</td><td>Interrupt signal</td><td>INTTM000 signal is generated each time value is captured.</td></tr>
<tr><td rowspan="5">Capture operation of CR010</td><td rowspan="4">TI000 pin input^{Note}
  </td><td>Set values of ES010 and ES000
Position of edge to be captured</td><td colspan="2" rowspan="5"></td></tr>
<tr><td>01: Rising  </td></tr>
<tr><td>00: Falling  </td></tr>
<tr><td>11: Both edges  </td></tr>
<tr><td>Interrupt signal</td><td>INTTM010 signal is generated each time value is captured.</td></tr>
</table>

Note The capture operation of CR010 is not affected by the setting of the CRC001 bit.

Caution To capture the count value of the TM00 register to the CR000 register by using the phase reverse to that input to the TI000 pin, the interrupt request signal (INTTM000) is not generated after the value has been captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. To not use the external interrupt, mask the INTTM000 signal.

Remark CRC001: Refer to **7.3 (2) Capture/compare control register 00 (CRC00)**.
ES101 ES100, ES001 ES000: Refer to **7.3 (4) Prescaler mode register 00 (PRM00)**.

7.3 Registers Controlling 16-bit Timer/Event Counter 00

Registers used to control 16-bit timer/event counter 00 are shown below.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Port alternate switch control register (MUXSEL)[Note]
- Port mode register 0 (PM0)
- Port register 0 (P0)

Note 78K0/FB2-L only.

(1) 16-bit timer mode control register 00 (TMC00)

TMC00 is an 8-bit register that sets the 16-bit timer/event counter 00 operation mode, TM00 clear mode, and output timing, and detects an overflow.

Rewriting TMC00 is prohibited during operation (when TMC003 and TMC002 = other than 00). However, it can be changed when TMC003 and TMC002 are cleared to 00 (stopping operation) and when OVF00 is cleared to 0.

TMC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TMC00 to 00H.

Caution 16-bit timer/event counter 00 starts operation at the moment TMC003 and TMC002 are set to values other than 00 (operation stop mode), respectively. Set TMC003 and TMC002 to 00 to stop the operation.

Figure 7-5. 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 (ES010, ES000) of prescaler mode register 00 (PRM00).

(2) Capture/compare control register 00 (CRC00)

CRC00 is the register that controls the operation of CR000 and CR010.

Changing the value of CRC00 is prohibited during operation (when TMC003 and TMC002 = other than 00).

CRC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CRC00 to 00H.

Figure 7-6. 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

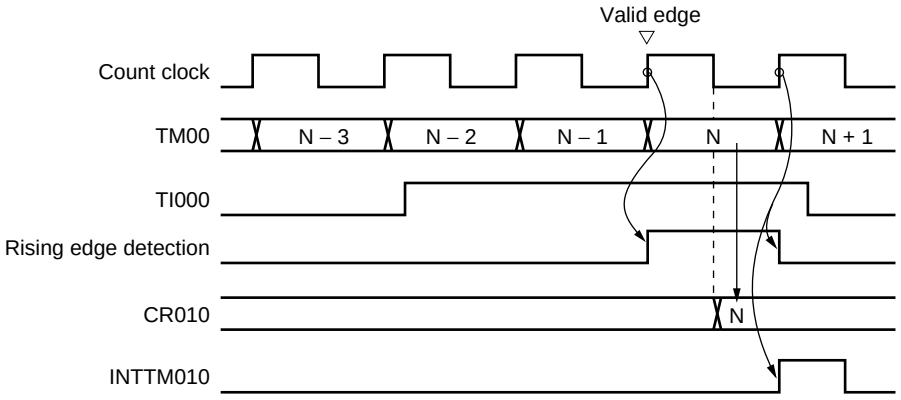
CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010 pin
1	Captures on valid edge of TI000 pin by reverse phase[Note]
The valid edge of the TI010 and TI000 pin is set by PRM00. If ES010 and ES000 are set to 11 (both edges) when CRC001 is 1, the valid edge of the TI000 pin cannot be 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 7-7. Example of CR010 Capture Operation (When Rising Edge Is Specified)



(3) 16-bit timer output control register 00 (TOC00)

TOC00 is an 8-bit register that controls the TO00 output.

TOC00 can be rewritten while only OSPT00 is operating (when TMC003 and TMC002 = other than 00). Rewriting the other bits is prohibited during operation.

However, TOC004 can be rewritten during timer operation as a means to rewrite CR010 (refer to **7.5.1 Rewriting CR010 during TM00 operation**).

TOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TOC00 to 00H.

Caution Be sure to set TOC00 using the following procedure.

<1> Set TOC004 and TOC001 to 1.

<2> Set only TOE00 to 1.

<3> Set either of LVS00 or LVR00 to 1.

Figure 7-8. 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, refer to **7.5.2 Setting LVS00 and LVR00**.
- 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

(4) Prescaler mode register 00 (PRM00)

PRM00 is the register that sets the TM00 count clock and TI000 and TI010 pin input valid edges.

Rewriting PRM00 is prohibited during operation (when TMC003 and TMC002 = other than 00).

PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears PRM00 to 00H.

Cautions 1. Do not apply the following setting when setting the PRM001 and PRM000 bits to 11 (to specify the valid edge of the TI000 pin as a count clock).

- **Clear & start mode entered by the TI000 pin valid edge**
- **Setting the TI000 pin as a capture trigger**

2. If the operation of the 16-bit timer/event counter 00 is enabled when the TI000 or TI010 pin is at high level and when the valid edge of the TI000 or TI010 pin is specified to be the rising edge or both edges, the high level of the TI000 or TI010 pin is detected as a rising edge. Note this when the TI000 or TI010 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. Select either of the functions.

Figure 7-9. Format of Prescaler Mode Register 00 (PRM00)

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000

ES110	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

ES010	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[Note 1]				
			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	78.13 kHz
1	1	TI000 valid edge[Notes 2, 3]				

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.
- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. The external clock from the TI000 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

3. Do not start timer operation with the external clock from the TI000 pin when in the STOP mode.

Remark f_{PRS}: Peripheral hardware clock frequency

(5) Port alternate switch control register (MUXSEL)[Note]

This register assigns the pin function.

The timer input (TI000) function can be assigned to the P121 pin of the 78K0/FB2-L.

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

Reset signal generation clears MUXSEL to 00H.

Note 78K0/FB2-L only

Figure 7-10. Format of Port Alternate Switch Control Register (MUXSEL)

Address: FF39H After reset: 00H R/W

Symbol	7	<6>	5	<4>	3	2	1	0
MUXSEL	0	INTP0SEL0	0	TM00SEL0	0	0	0	0

TM00SEL0	16-bit timer/event counter 00 input (TI000) pin assignment
0	(default)
1	P121/TI000

(6) Port mode register 0 (PM0)

This register sets port 0 input/output in 1-bit units.

When using the P01/TO00/TI010 pin for timer output, set PM01 and the output latches of P01 to 0.

When using the P00/TI000/INTP0 and P01/TI010/TO00 pins for timer input, set PM00 and PM01 to 1. At this time, the output latches of P00 and P01 may be 0 or 1.

PM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM0 to FFH.

Remark When the timer input (TI000) function is assigned to the P121 pin by setting the input switch register (MUXSEL), the port mode register and port register are not required to be set.

Figure 7-11. Format of Port Mode Register 0 (PM0)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	PM02	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode register 0 of the 78K0/FB2-L. For the format of port mode register 0 of other products, refer to **(1) Port mode registers (PMxx)** in **4.3 Registers Controlling Port Function**.

7.4 Operation of 16-bit Timer/Event Counter 00

7.4.1 Interval timer operation

If bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register (TMC00) are set to 11 (clear & start mode entered upon a match between TM00 and CR000), the count operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H and a match interrupt signal (INTTM000) is generated. This INTTM000 signal enables TM00 to operate as an interval timer.

Remarks 1. For the setting of I/O pins, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 7-12. Block Diagram of Interval Timer Operation

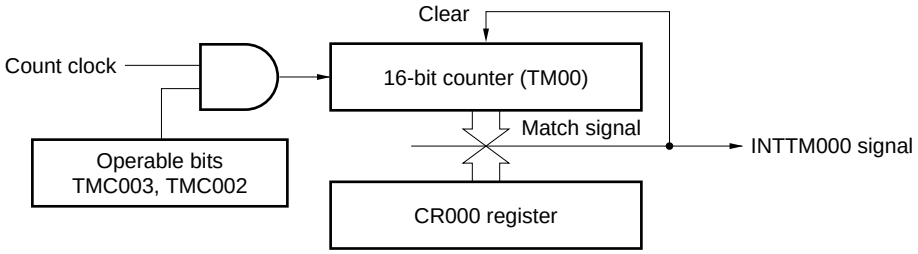


Figure 7-13. Basic Timing Example of Interval Timer Operation

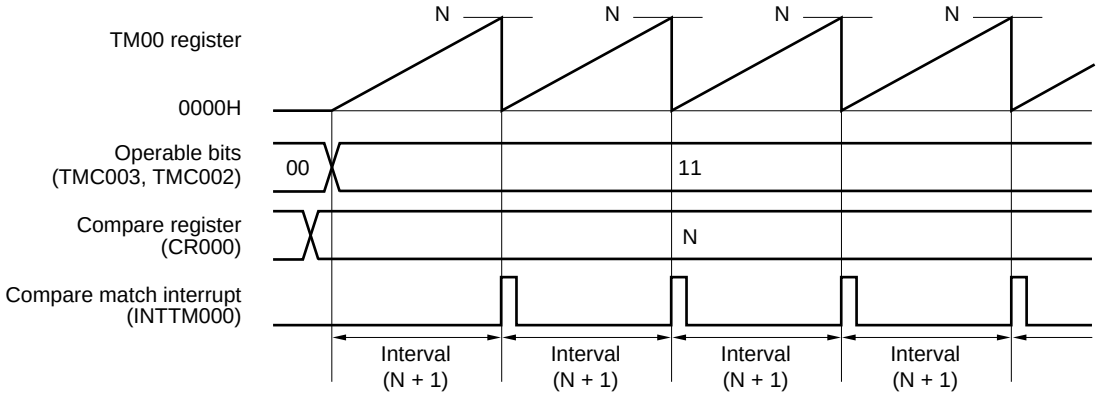
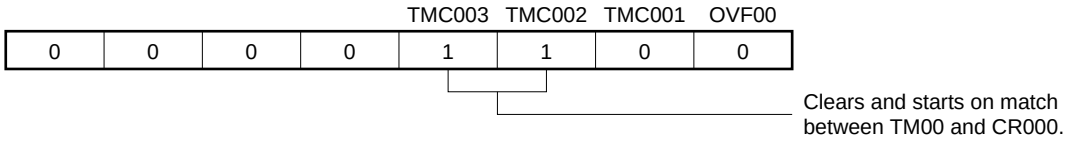
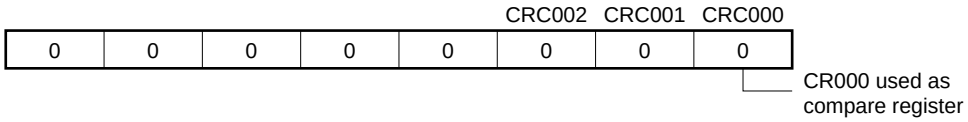


Figure 7-14. Example of Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 00 (TMC00)



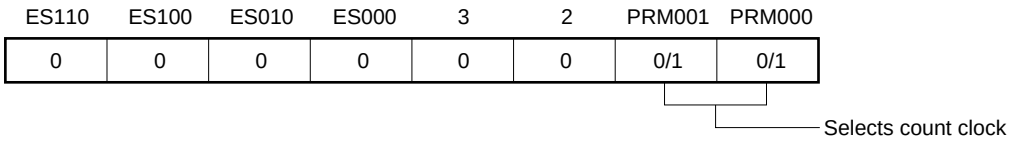
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

- Interval time = (M + 1) × Count clock cycle

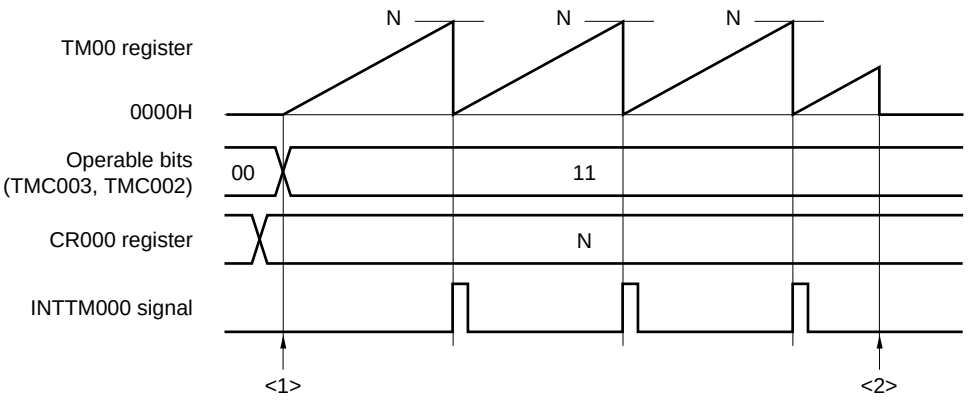
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

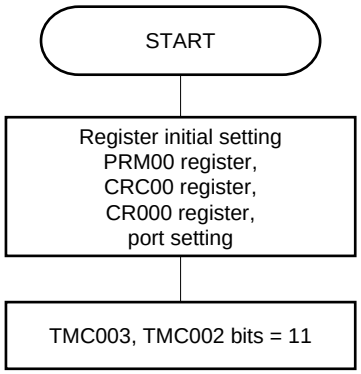
Usually, CR010 is not used for the interval timer function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 7-15. Example of Software Processing for Interval Timer Function



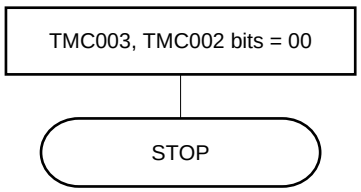
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

7.4.2 Square-wave output operation

When 16-bit timer/event counter 00 operates as an interval timer (refer to **7.4.1**), a square wave can be output from the TO00 pin by setting the 16-bit timer output control register 00 (TOC00) to 03H.

When TMC003 and TMC002 are set to 11 (count clear & start mode entered upon a match between TM00 and CR000), the counting operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H, an interrupt signal (INTTM000) is generated, and TO00 output is inverted. This TO00 output that is inverted at fixed intervals enables TO0n to output a square wave.

Remarks 1. For the setting of I/O pins, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 7-16. Block Diagram of Square-Wave Output Operation

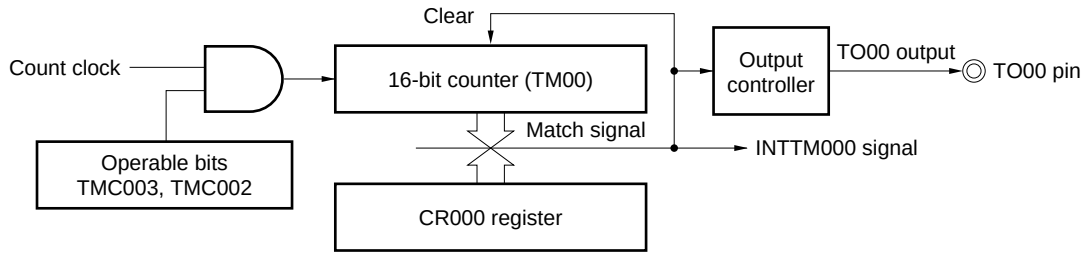


Figure 7-17. Basic Timing Example of Square-Wave Output Operation

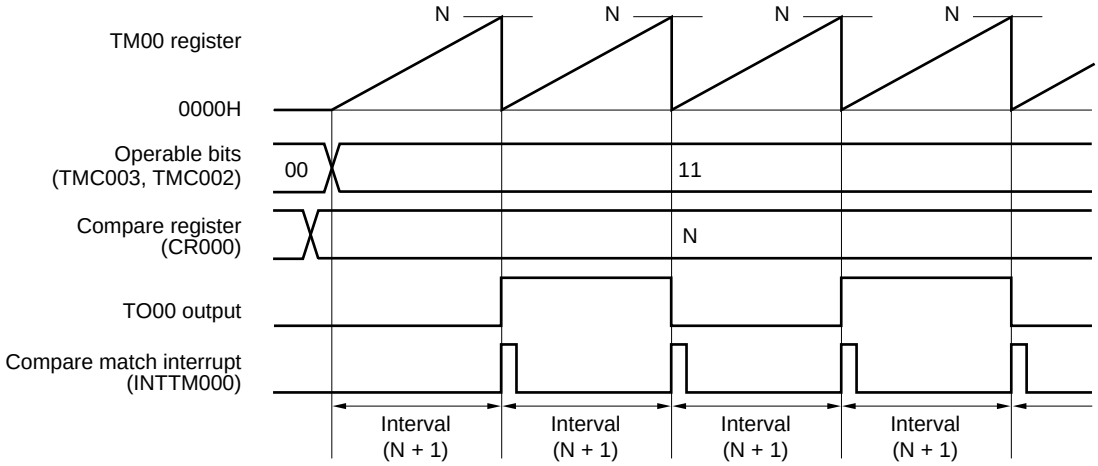
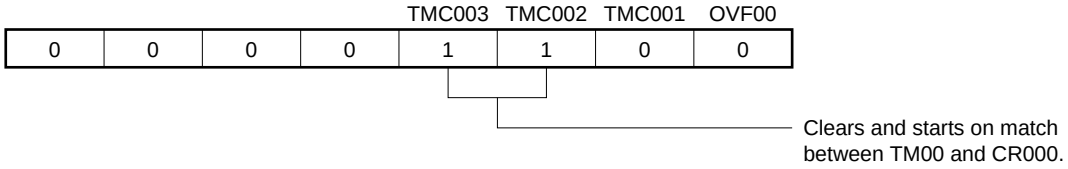
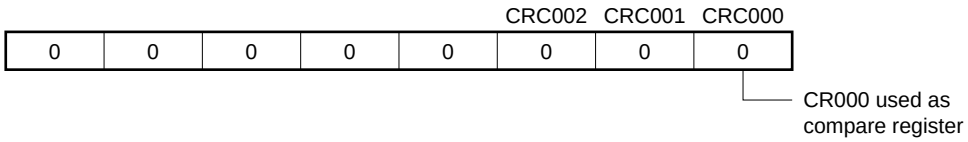


Figure 7-18. Example of Register Settings for Square-Wave Output Operation

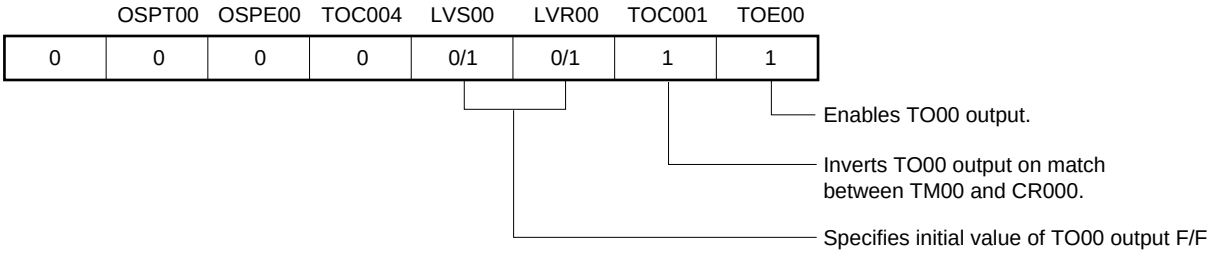
(a) 16-bit timer mode control register 00 (TMC00)



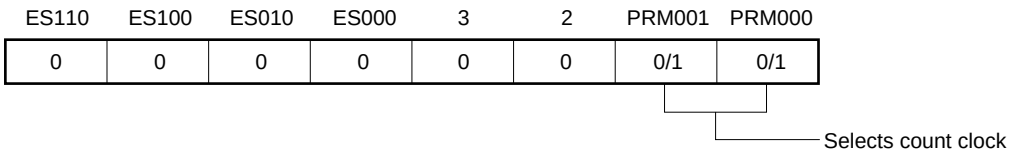
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

- Square wave frequency = 1 / [2 × (M + 1) × Count clock cycle]

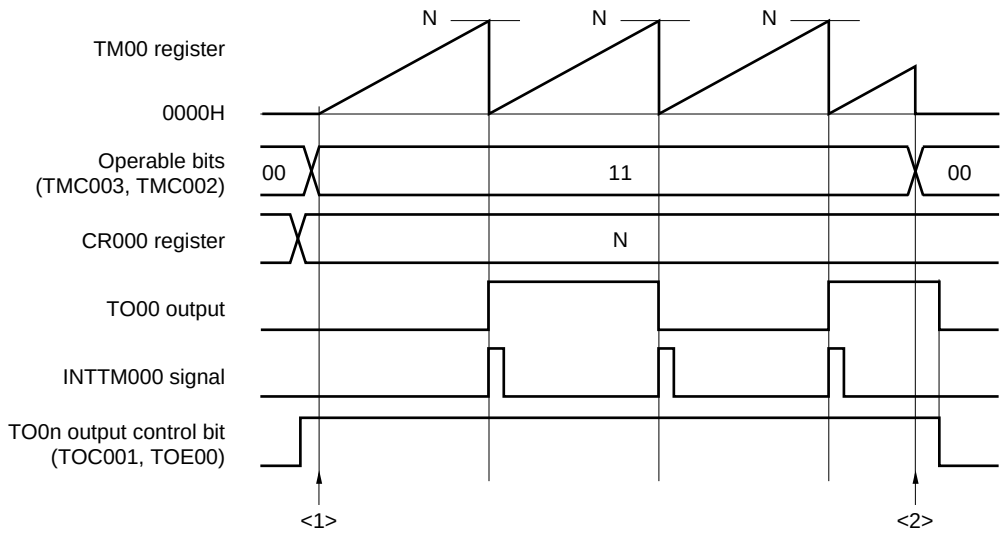
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

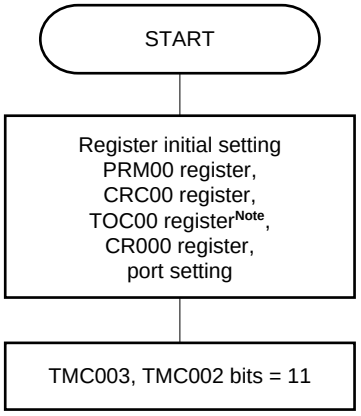
Usually, CR010 is not used for the square-wave output function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 7-19. Example of Software Processing for Square-Wave Output Function



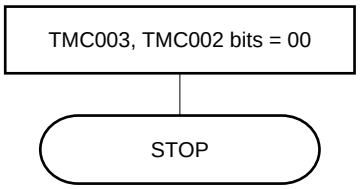
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Note Care must be exercised when setting TOC00. For details, refer to **7.3 (3) 16-bit timer output control register 00 (TOC00)**.

7.4.3 External event counter operation

When bits 1 and 0 (PRM001 and PRM000) of the prescaler mode register 00 (PRM00) are set to 11 (for counting up with the valid edge of the TI000 pin) and bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11, the valid edge of an external event input is counted, and a match interrupt signal indicating matching between TM00 and CR000 (INTTM000) is generated.

To input the external event, the TI000 pin is used. Therefore, the timer/event counter cannot be used as an external event counter in the clear & start mode entered by the TI000 pin valid edge input (when TMC003 and TMC002 = 10).

The INTTM000 signal is generated with the following timing.

- Timing of generation of INTTM000 signal (second time or later)
  = Number of times of detection of valid edge of external event × (Set value of CR000 + 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 INTTM000 signal (first time only)
  = Number of times of detection of valid edge of external event input × (Set value of CR000 + 2)

To detect the valid edge, the signal input to the TI000 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, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 7-20. Block Diagram of External Event Counter Operation

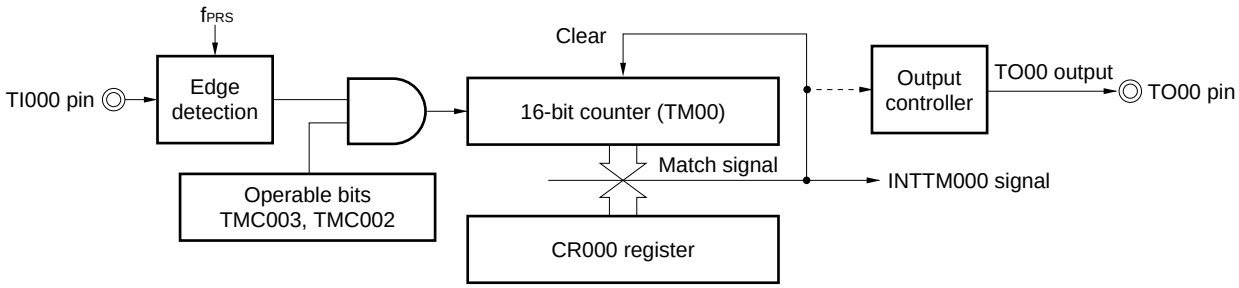
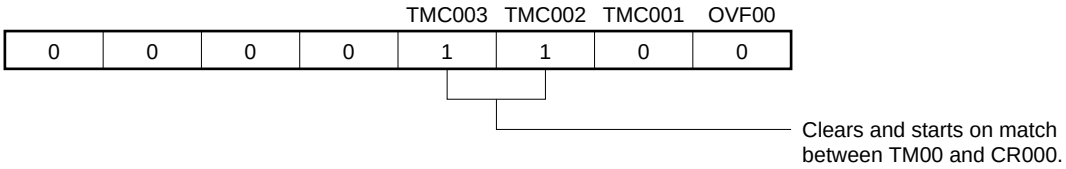
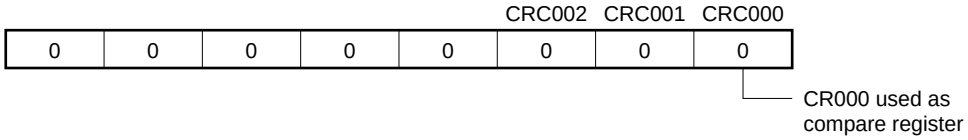


Figure 7-21. Example of Register Settings in External Event Counter Mode (1/2)

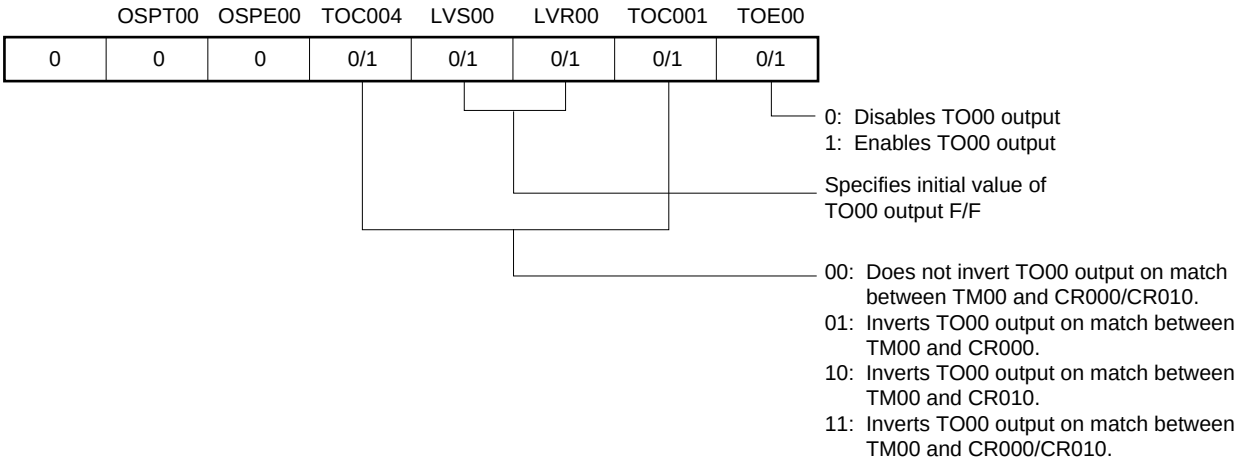
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

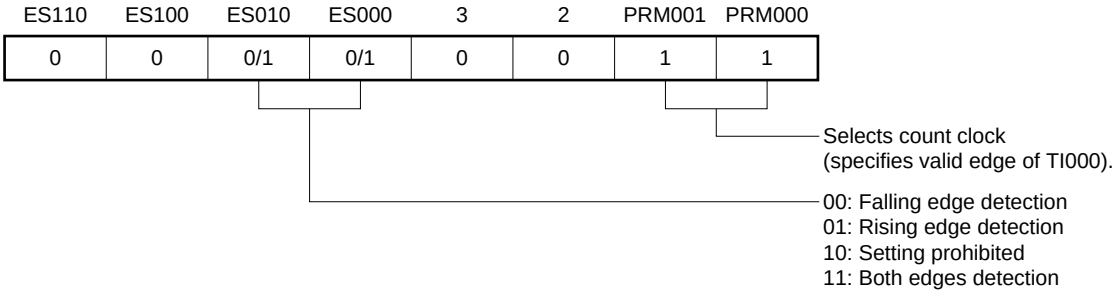


Figure 7-21. Example of Register Settings in External Event Counter Mode (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interrupt signal (INTTM000) is generated when the number of external events reaches (M + 1).

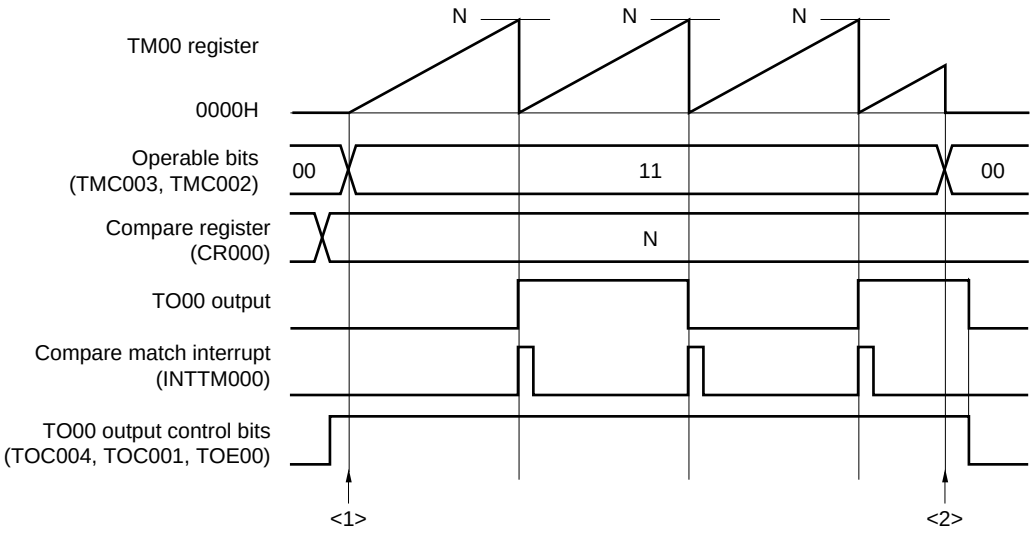
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

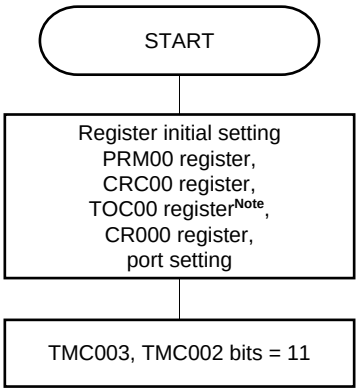
Usually, CR010 is not used in the external event counter mode. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 7-22. Example of Software Processing in External Event Counter Mode



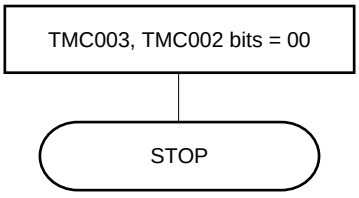
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Note Care must be exercised when setting TOC00. For details, refer to **7.3 (3) 16-bit timer output control register 00 (TOC00)**.

7.4.4 Operation in clear & start mode entered by TI000 pin valid edge input

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 10 (clear & start mode entered by the TI000 pin valid edge input) and the count clock (set by PRM00) is supplied to the timer/event counter, TM00 starts counting up. When the valid edge of the TI000 pin is detected during the counting operation, TM00 is cleared to 0000H and starts counting up again. If the valid edge of the TI000 pin is not detected, TM00 overflows and continues counting.

The valid edge of the TI000 pin is a cause to clear TM00. Starting the counter is not controlled immediately after the start of the operation.

CR000 and CR010 are used as compare registers and capture registers.

(a) When CR000 and CR010 are used as compare registers

Signals INTTM000 and INTTM010 are generated when the value of TM00 matches the value of CR000 and CR010.

(b) When CR000 and CR010 are used as capture registers

The count value of TM00 is captured to CR000 and the INTTM000 signal is generated when the valid edge is input to the TI010 pin (or when the phase reverse to that of the valid edge is input to the TI000 pin).

When the valid edge is input to the TI000 pin, the count value of TM00 is captured to CR010 and the INTTM010 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 TI000 pin (PRM001 and PRM000 = 11). When PRM001 and PRM000 = 11, TM00 is cleared.

Remarks 1. For the setting of the I/O pins, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

(1) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: compare register, CR010: compare register)

Figure 7-23. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

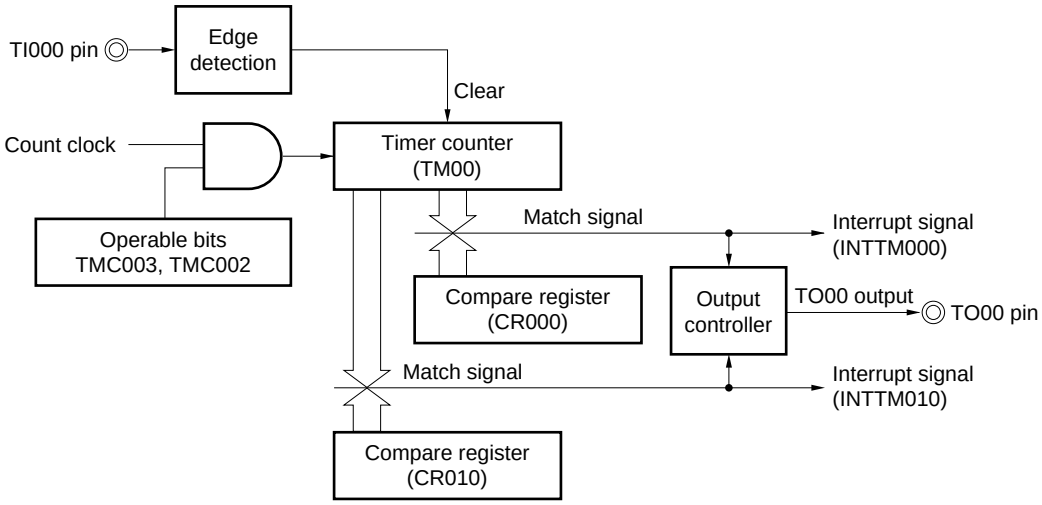
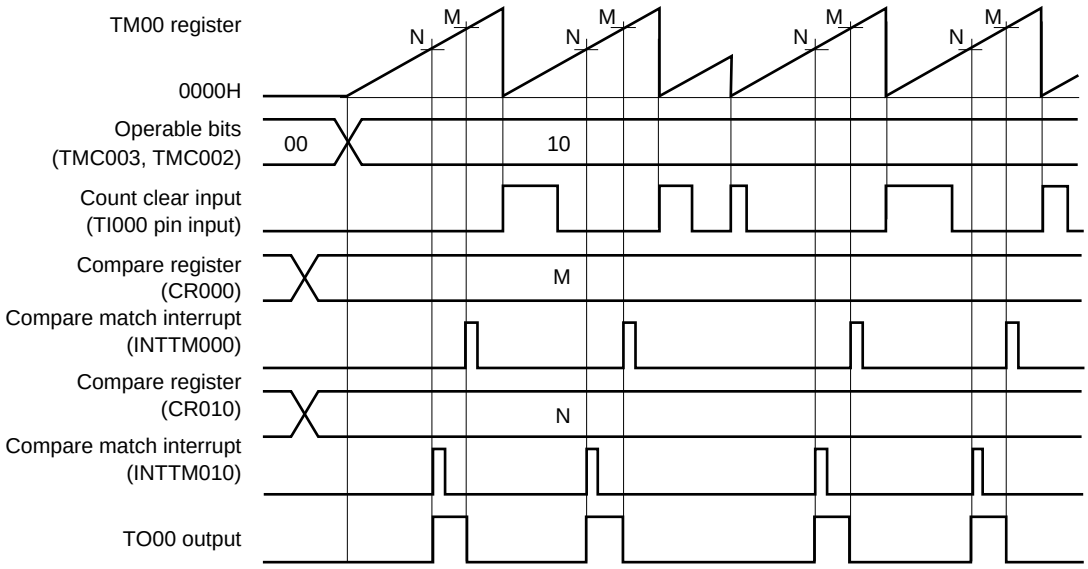
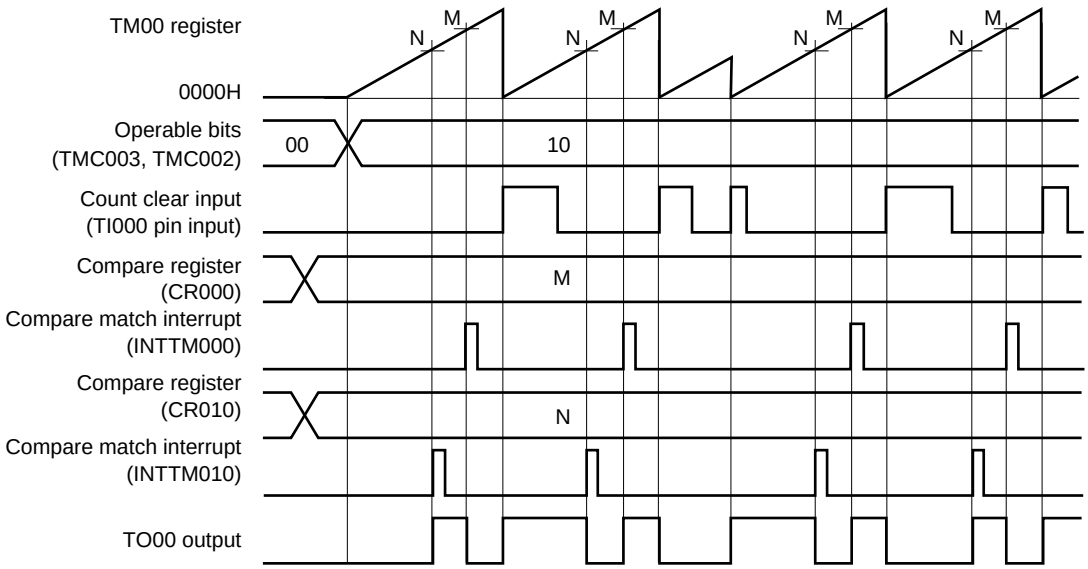


Figure 7-24. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

(a) TOC00 = 13H, PRM00 = 10H, CRC00 = 00H, TMC00 = 08H



(b) TOC00 = 13H, PRM00 = 10H, CRC00 = 00H, TMC00 = 0AH



(a) and (b) differ as follows depending on the setting of bit 1 (TMC001) of the 16-bit timer mode control register 00 (TMC00).

(a) The TO00 output level is inverted when TM00 matches a compare register.

(b) The TO00 output level is inverted when TM00 matches a compare register or when the valid edge of the TI000 pin is detected.

(2) Operation in clear & start mode entered by TI000 pin valid edge input
(CR000: compare register, CR010: capture register)

Figure 7-25. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register)

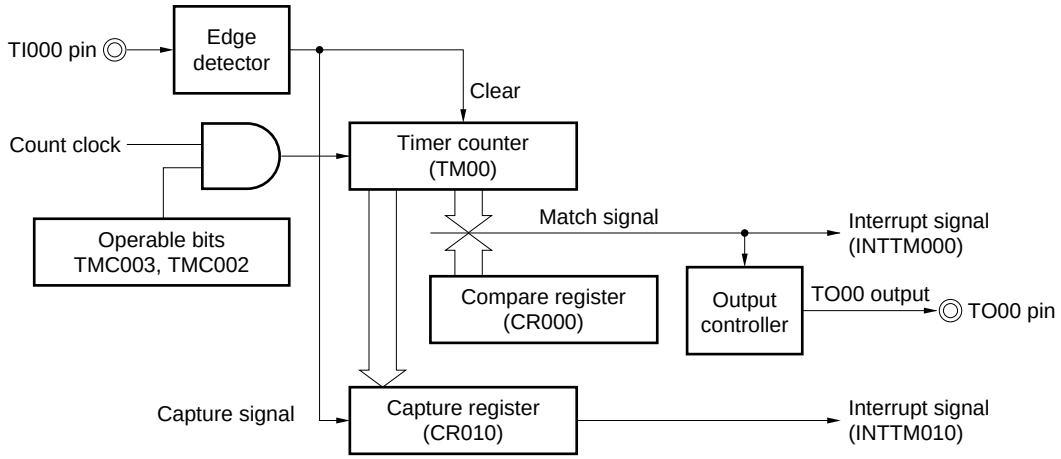
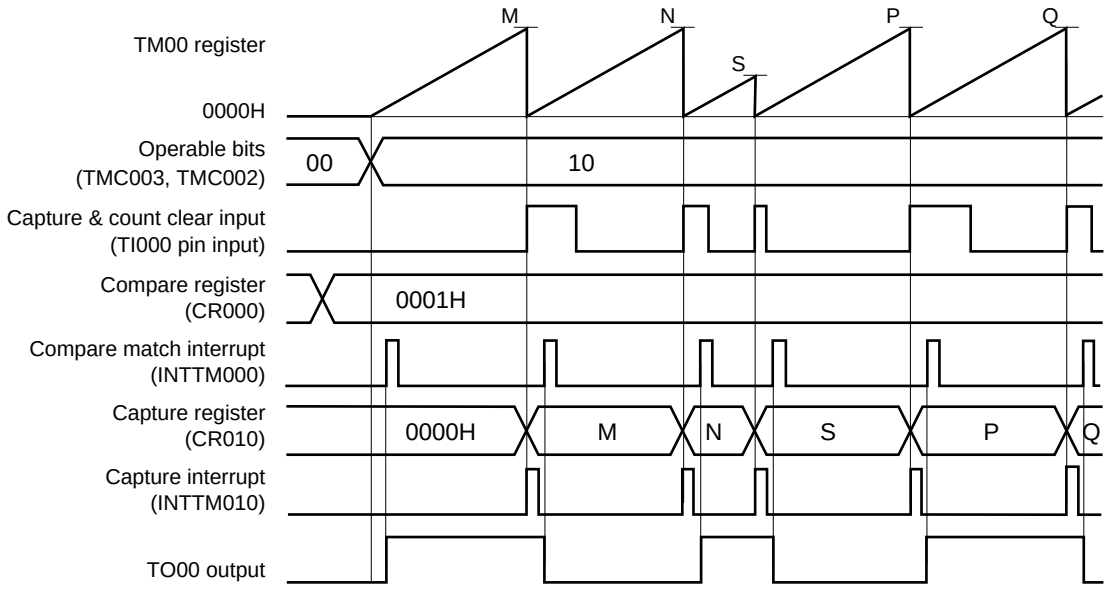


Figure 7-26. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 08H, CR000 = 0001H

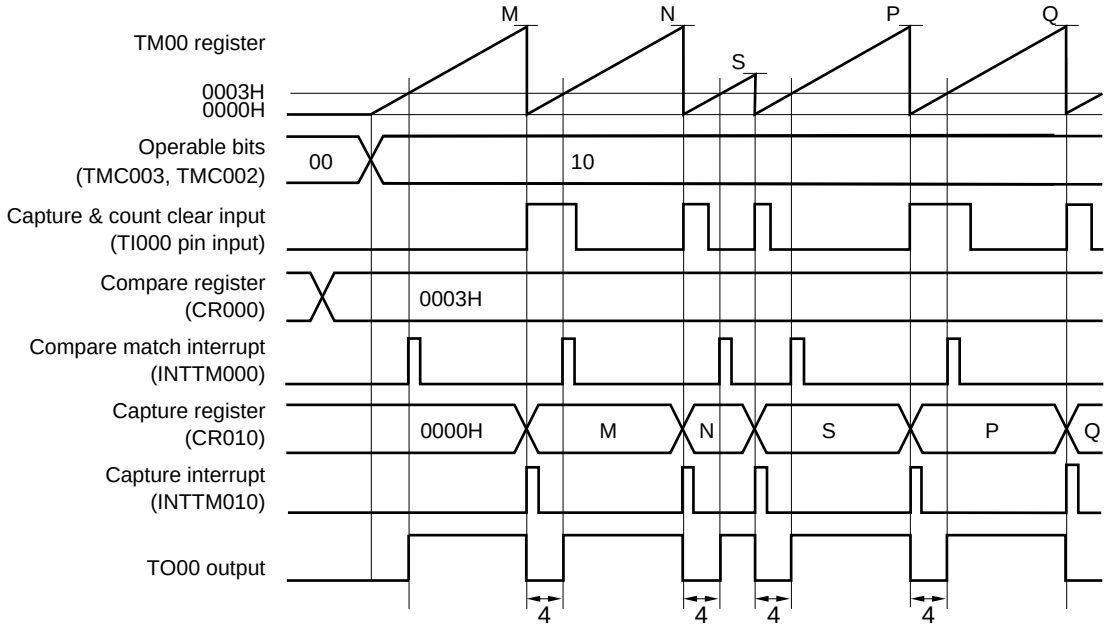


This is an application example where the TO00 output level is inverted when the count value has been captured & cleared.

The count value is captured to CR010 and TM00 is cleared (to 0000H) when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0001H, a compare match interrupt signal (INTTM000) is generated, and the TO00 output level is inverted.

Figure 7-26. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 0AH, CR000 = 0003H



This is an application example where the width set to CR000 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

The count value is captured to CR010, a capture interrupt signal (INTTM010) is generated, TM00 is cleared (to 0000H), and the TO00 output level is inverted when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0003H (four clocks have been counted), a compare match interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(3) Operation in clear & start mode by entered TI000 pin valid edge input
(CR000: capture register, CR010: compare register)

Figure 7-27. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register)

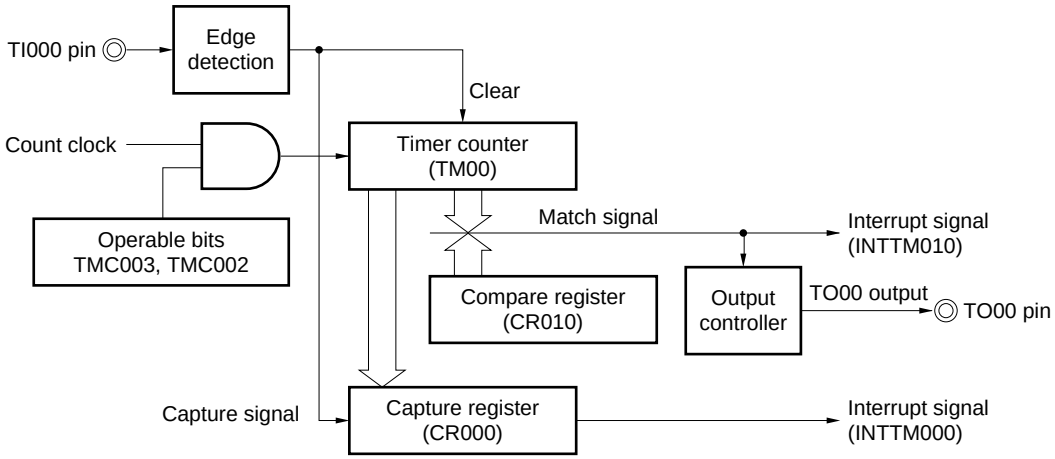
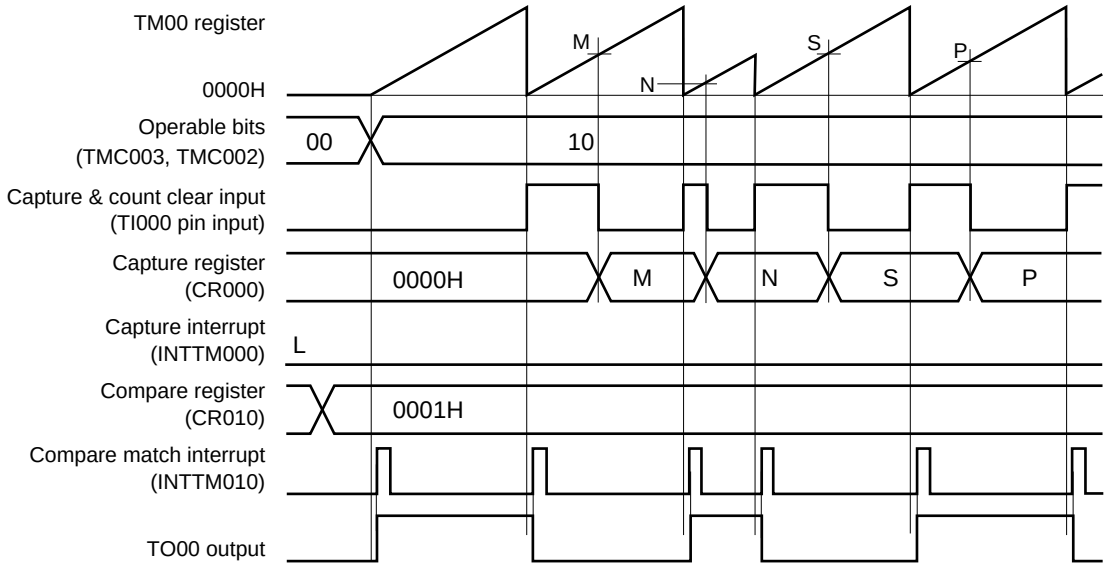


Figure 7-28. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00 = 03H, TMC00 = 08H, CR010 = 0001H



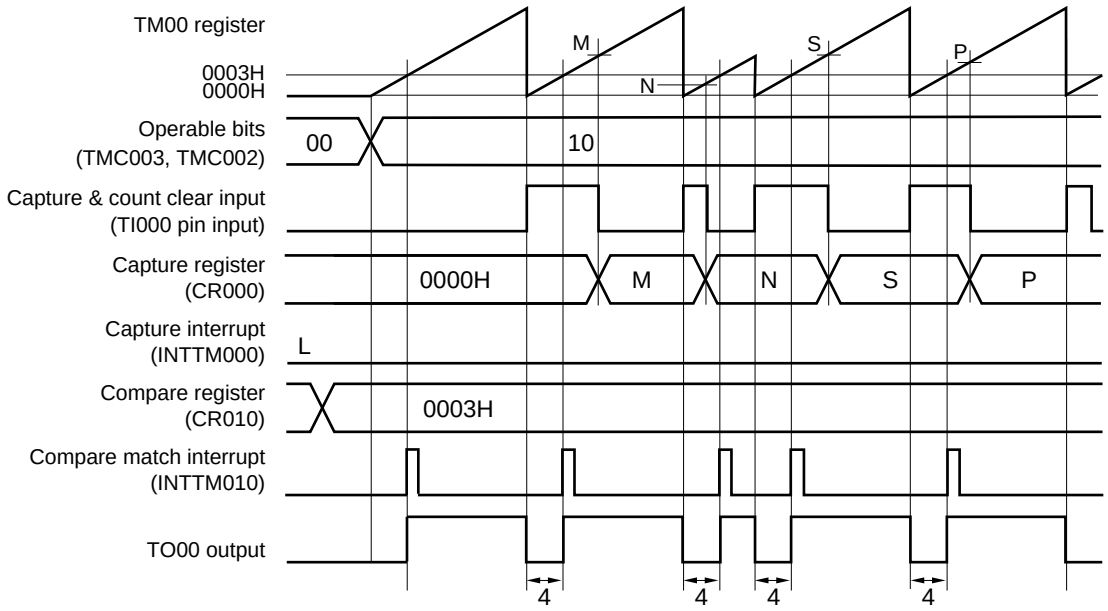
This is an application example where the TO00 output level is to be inverted when the count value has been captured & cleared.

TM00 is cleared at the rising edge detection of the TI000 pin and it is captured to CR000 at the falling edge detection of the TI000 pin.

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is set to 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the signal input to the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 signal is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

Figure 7-28. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00 = 03H, TMC00 = 0AH, CR010 = 0003H



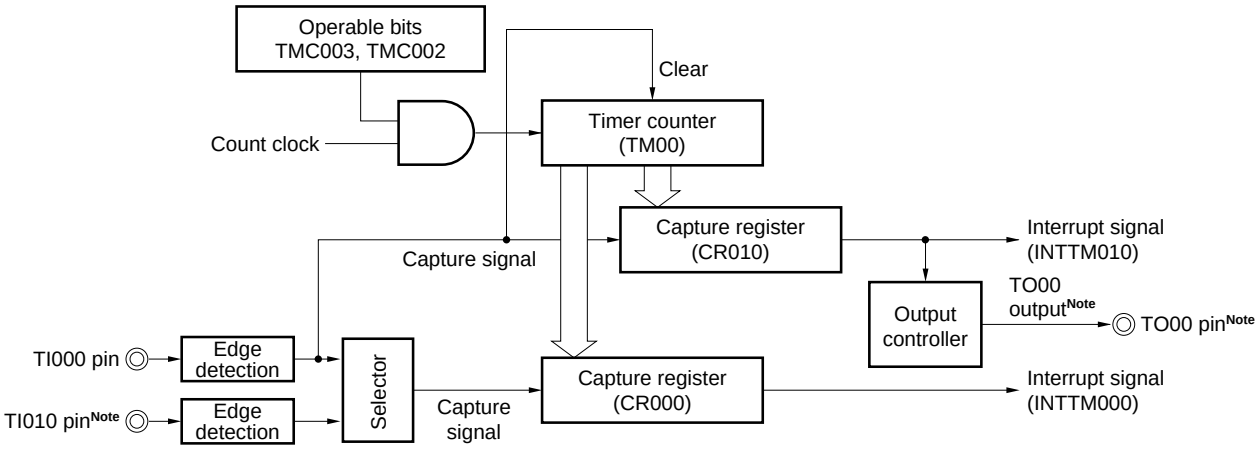
This is an application example where the width set to CR010 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

TM00 is cleared (to 0000H) at the rising edge detection of the TI000 pin and captured to CR000 at the falling edge detection of the TI000 pin. The TO00 output level is inverted when TM00 is cleared (to 0000H) because the rising edge of the TI000 pin has been detected or when the value of TM00 matches that of a compare register (CR010).

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the input signal of the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 interrupt is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

(4) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: capture register, CR010: capture register)

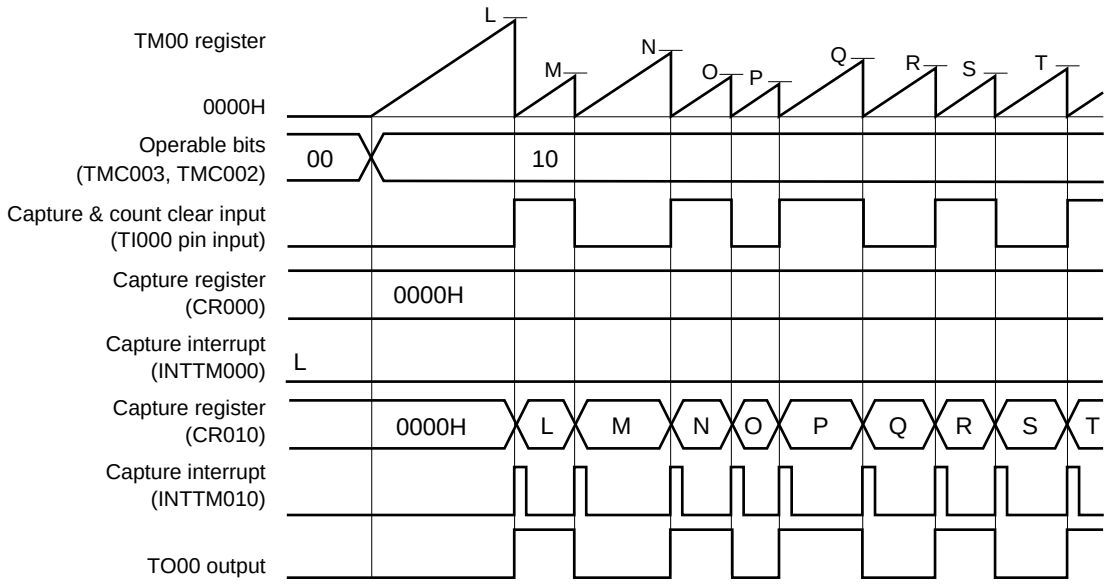
Figure 7-29. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register)



Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 7-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (1/3)

(a) TOC00 = 13H, PRM00 = 30H, CRC00 = 05H, TMC00 = 0AH

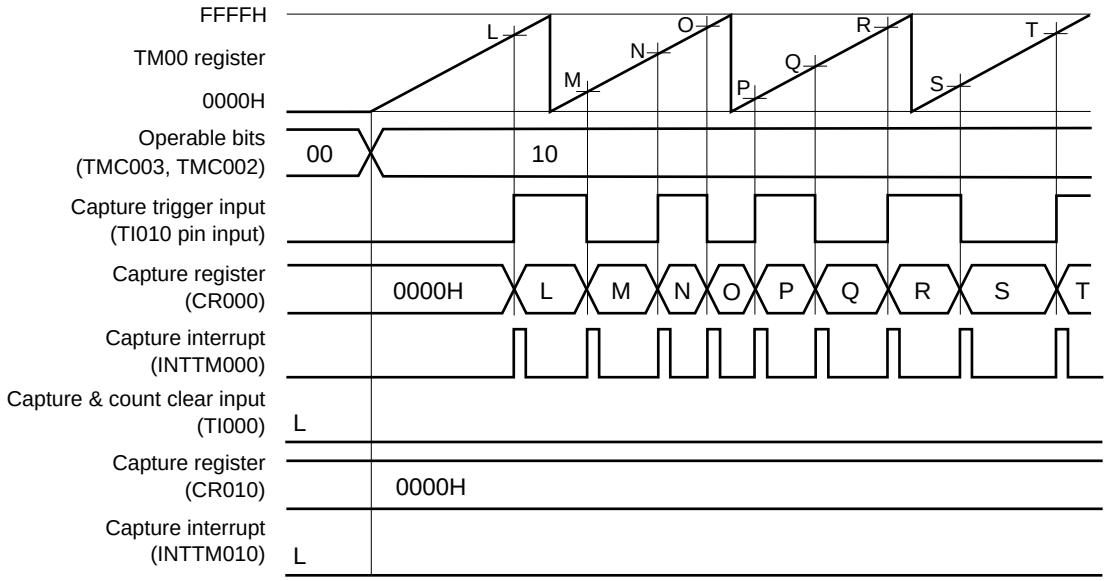


This is an application example where the count value is captured to CR010, TM00 is cleared, and the TO00 output is inverted when the rising or falling edge of the TI000 pin is detected.

When the edge of the TI010 pin is detected, an interrupt signal (INTTM000) is generated. Mask the INTTM000 signal when it is not used.

Figure 7-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (2/3)

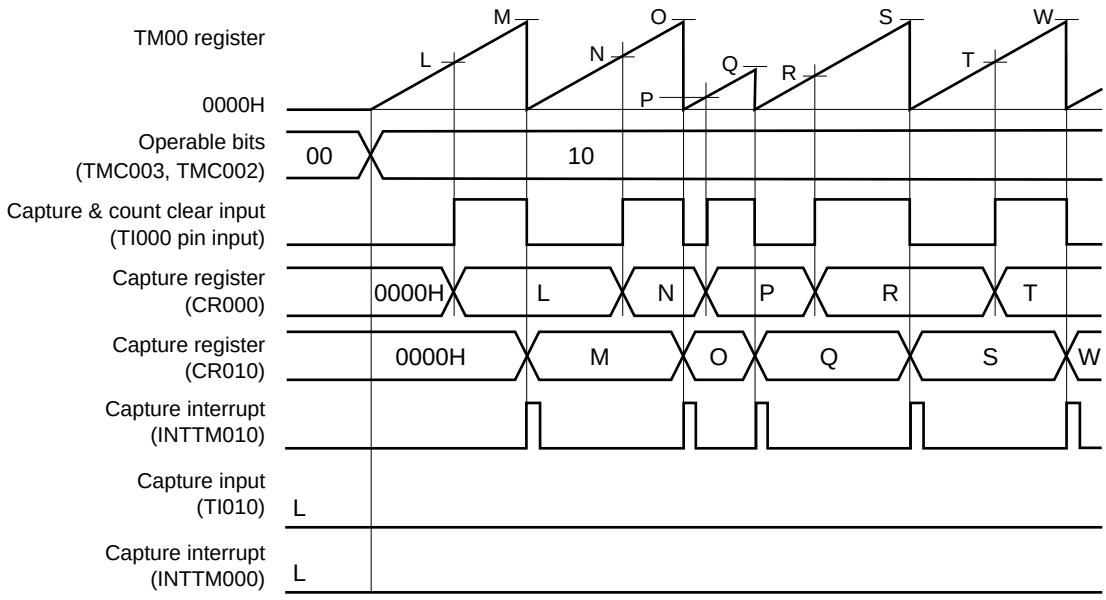
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH



This is a timing example where an edge is not input to the TI000 pin, in an application where the count value is captured to CR000 when the rising or falling edge of the TI010 pin is detected.

Figure 7-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (3/3)

(c) TOC00 = 13H, PRM00 = 00H, CRC00 = 07H, TMC00 = 0AH



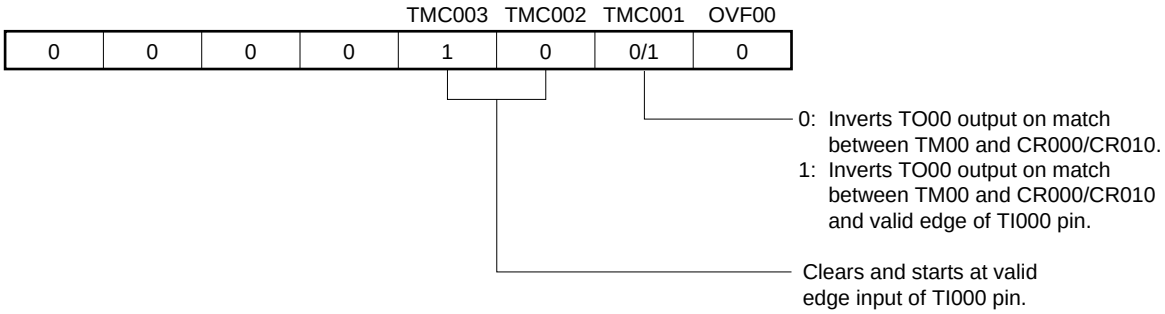
This is an application example where the pulse width of the signal input to the TI000 pin is measured.
By setting CRC00, the count value can be captured to CR000 in the phase reverse to the falling edge of the TI000 pin (i.e., rising edge) and to CR010 at the falling edge of the TI000 pin.
The high- and low-level widths of the input pulse can be calculated by the following expressions.

- High-level width = [CR010 value] – [CR000 value] × [Count clock cycle]
- Low-level width = [CR000 value] × [Count clock cycle]

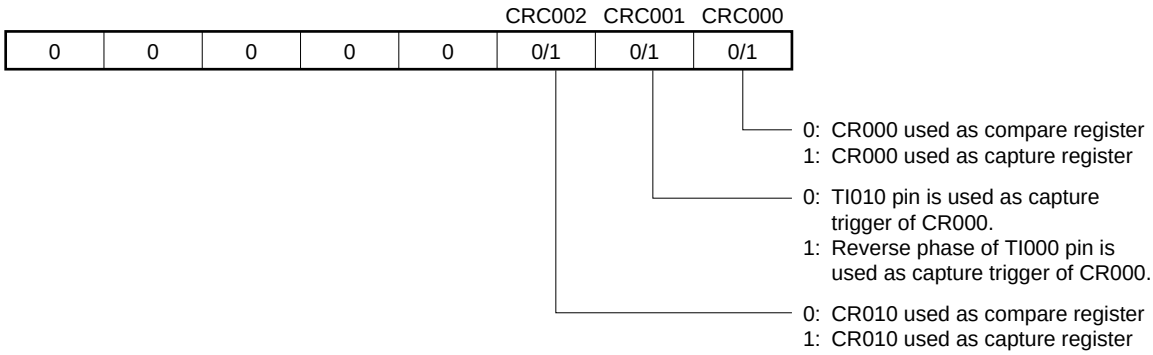
If the reverse phase of the TI000 pin is selected as a trigger to capture the count value to CR000, the INTTM000 signal is not generated. Read the values of CR000 and CR010 to measure the pulse width immediately after the INTTM010 signal is generated.
However, if the valid edge specified by bits 6 and 5 (ES110 and ES100) of prescaler mode register 00 (PRM00) is input to the TI010 pin, the count value is not captured but the INTTM000 signal is generated. To measure the pulse width of the TI000 pin, mask the INTTM000 signal when it is not used.

Figure 7-31. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (1/2)

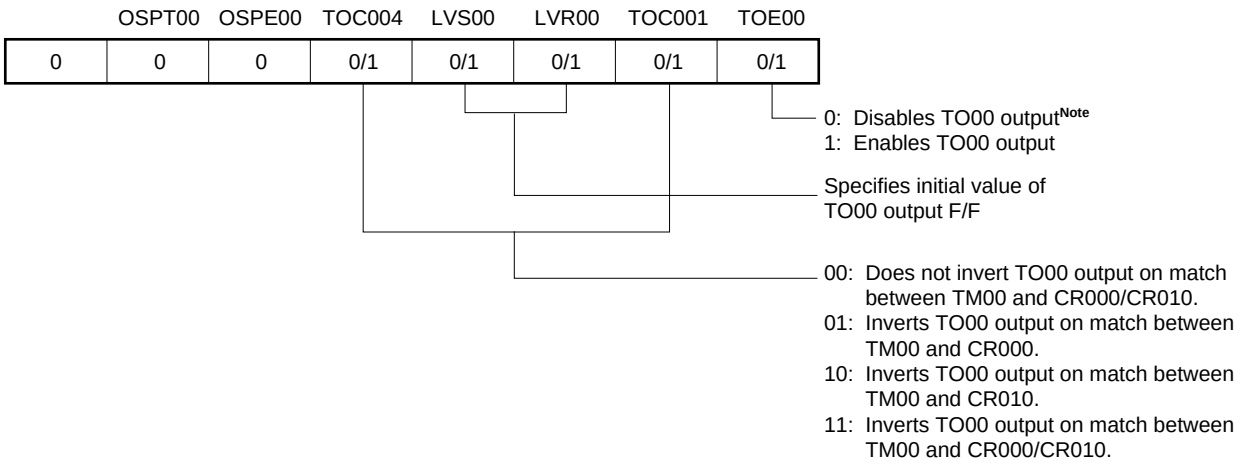
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



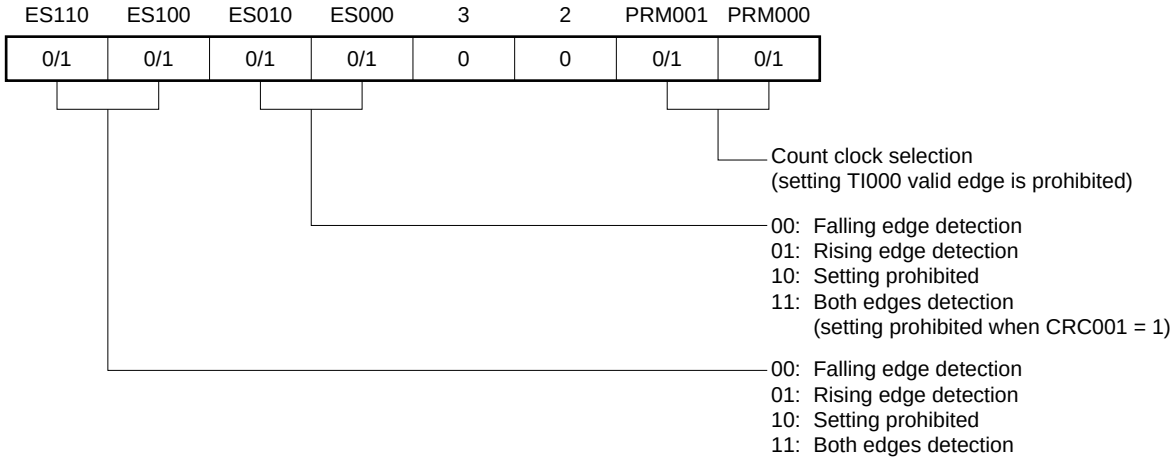
(c) 16-bit timer output control register 00 (TOC00)



Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 7-31. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

To use this register as a capture register, select either the TI000 or TI010 pin[Note] input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

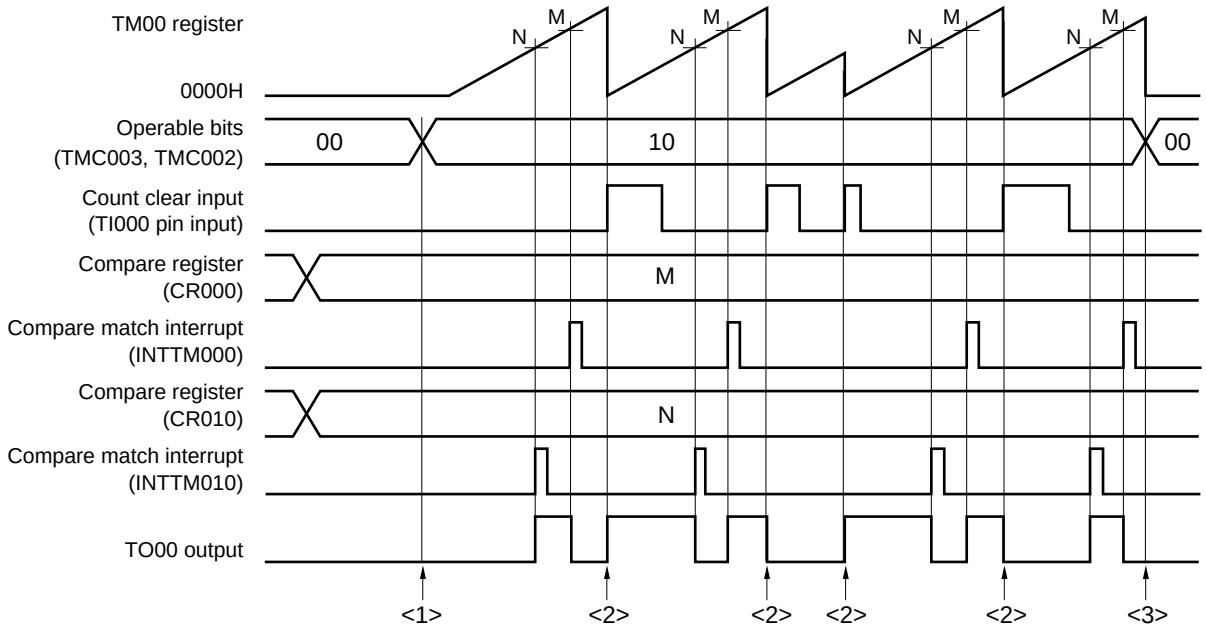
Note The timer output (TO00) cannot be used when detection of the valid edge of the TI010 pin is used.

(g) 16-bit capture/compare register 010 (CR010)

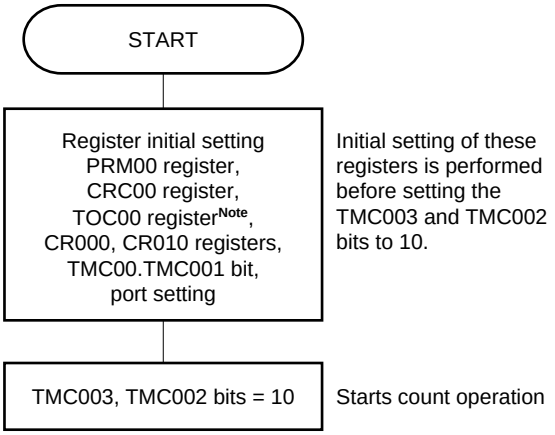
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

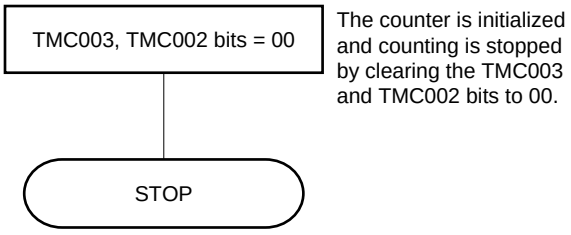
Figure 7-32. Example of Software Processing in Clear & Start Mode Entered by TI000 Pin Valid Edge Input



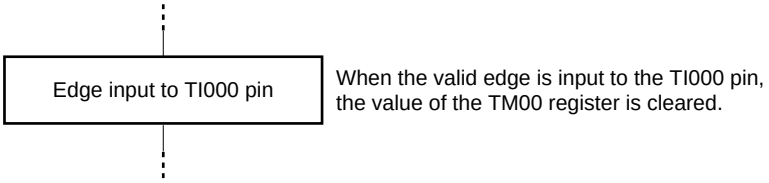
<1> Count operation start flow



<3> Count operation stop flow



<2> TM00 register clear & start flow



Note Care must be exercised when setting TOC00. For details, refer to **7.3 (3) 16-bit timer output control register 00 (TOC00)**.

7.4.5 Free-running timer operation

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 01 (free-running timer mode), 16-bit timer/event counter 00 continues counting up in synchronization with the count clock. When it has counted up to FFFFH, the overflow flag (OVF00) is set to 1 at the next clock, and TM00 is cleared (to 0000H) and continues counting. Clear OVF00 to 0 by executing the CLR instruction via software.

The following three types of free-running timer operations are available.

- Both CR000 and CR010 are used as compare registers.
- One of CR000 or CR010 is used as a compare register and the other is used as a capture register.
- Both CR000 and CR010 are used as capture registers.

Remarks 1. For the setting of the I/O pins, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

(1) Free-running timer mode operation

(CR000: compare register, CR010: compare register)

Figure 7-33. Block Diagram of Free-Running Timer Mode
(CR000: Compare Register, CR010: Compare Register)

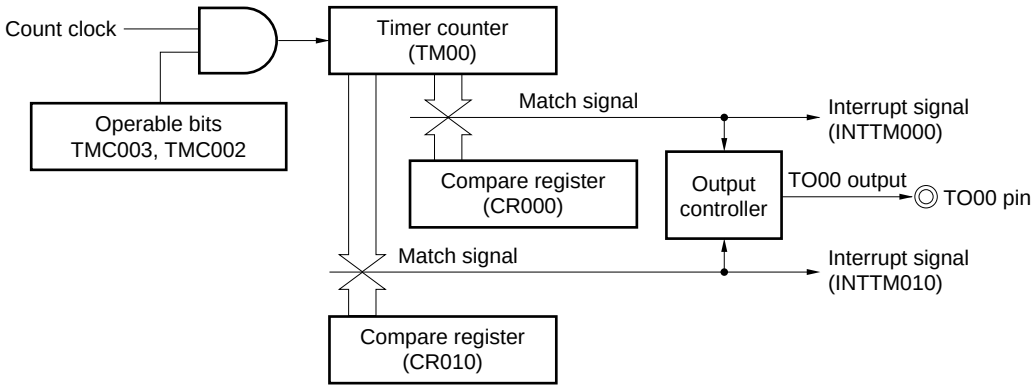
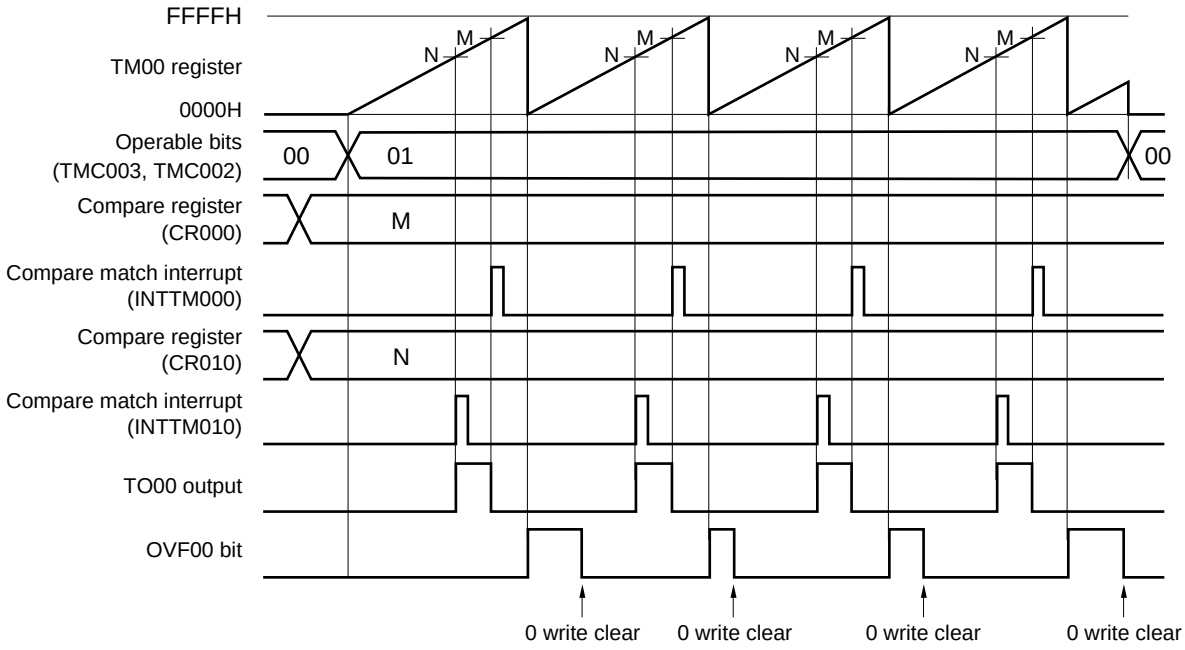


Figure 7-34. Timing Example of Free-Running Timer Mode
(CR000: Compare Register, CR010: Compare Register)

• TOC00 = 13H, PRM00 = 00H, CRC00 = 00H, TMC00 = 04H



This is an application example where two compare registers are used in the free-running timer mode.

The TO00 output level is reversed each time the count value of TM00 matches the set value of CR000 or CR010. When the count value matches the register value, the INTTM000 or INTTM010 signal is generated.

(2) Free-running timer mode operation
(CR000: compare register, CR010: capture register)

Figure 7-35. Block Diagram of Free-Running Timer Mode
(CR000: Compare Register, CR010: Capture Register)

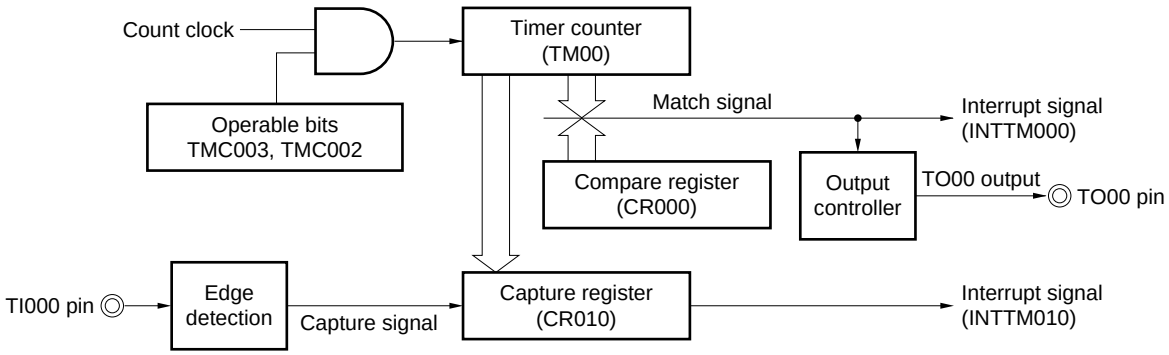
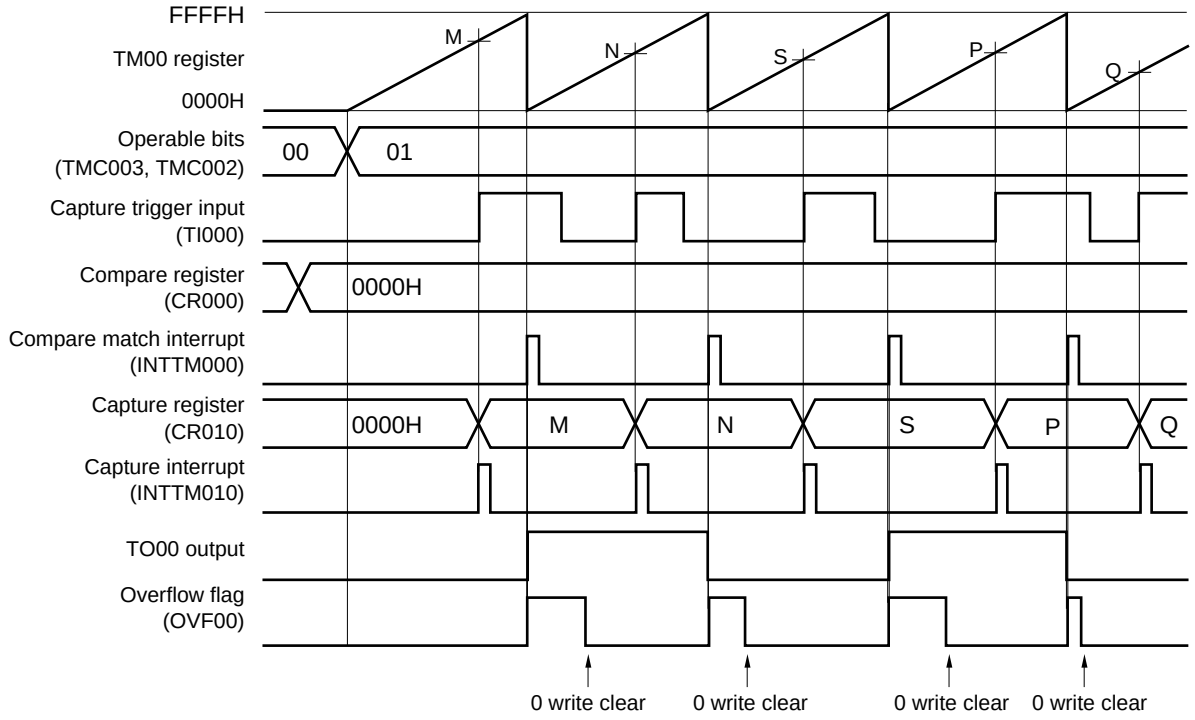


Figure 7-36. Timing Example of Free-Running Timer Mode
(CR000: Compare Register, CR010: Capture Register)

• TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 04H



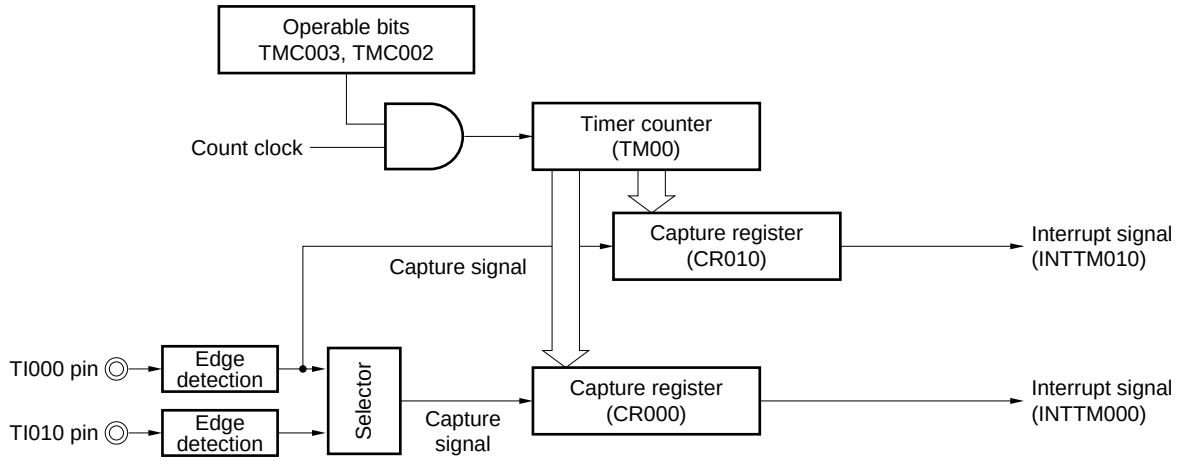
This is an application example where a compare register and a capture register are used at the same time in the free-running timer mode.

In this example, the INTTM000 signal is generated and the TO00 output level is reversed each time the count value of TM00 matches the set value of CR000 (compare register). In addition, the INTTM010 signal is generated and the count value of TM00 is captured to CR010 each time the valid edge of the TI000 pin is detected.

(3) Free-running timer mode operation

(CR000: capture register, CR010: capture register)

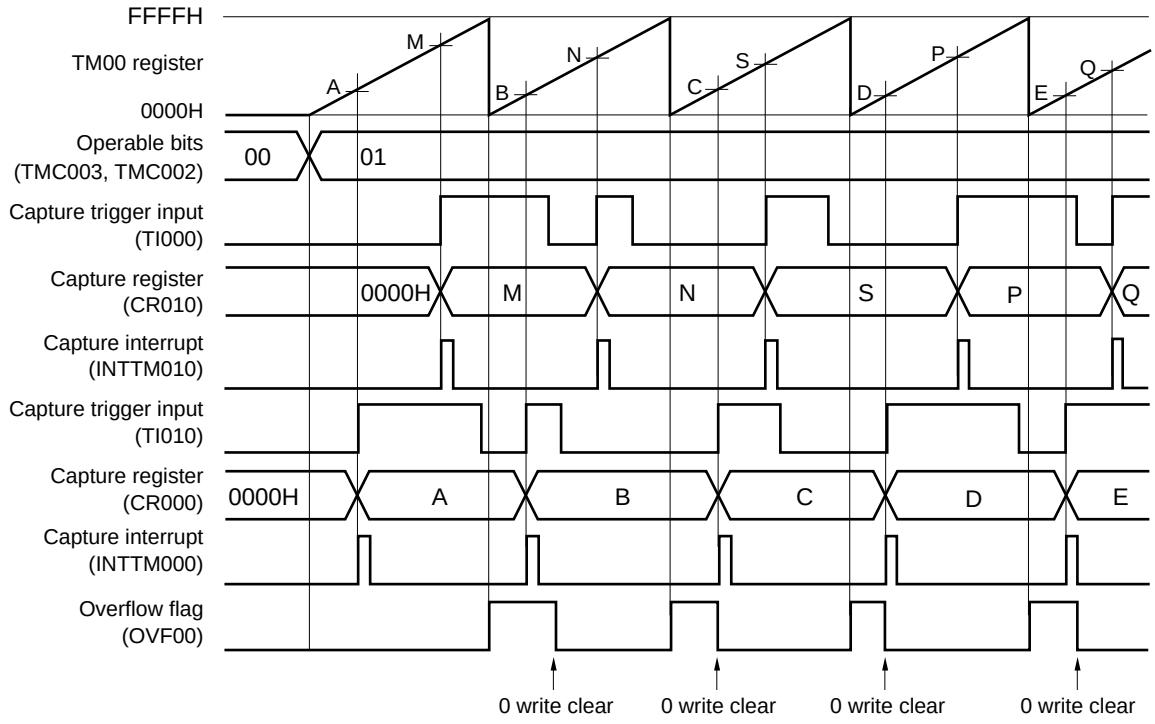
Figure 7-37. Block Diagram of Free-Running Timer Mode
(CR000: Capture Register, CR010: Capture Register)



Remark If both CR000 and CR010 are used as capture registers in the free-running timer mode, the TO00 output level is not inverted.
However, it can be inverted each time the valid edge of the TI000 pin is detected if bit 1 (TMC001) of 16-bit timer mode control register 00 (TMC00) is set to 1.

Figure 7-38. Timing Example of Free-Running Timer Mode (CR000: Capture Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 50H, CRC00 = 05H, TMC00 = 04H

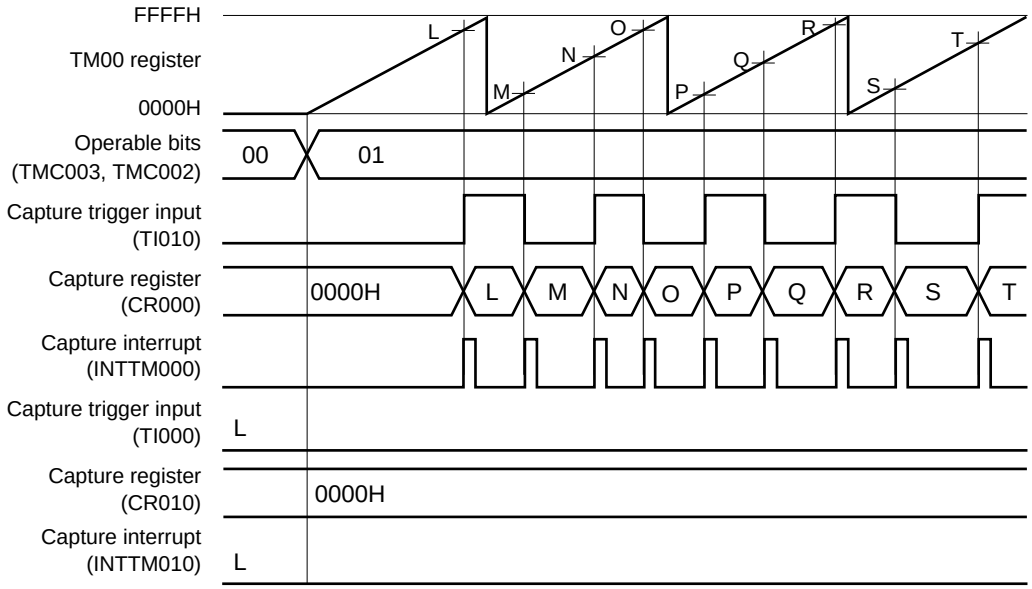


This is an application example where the count values that have been captured at the valid edges of separate capture trigger signals are stored in separate capture registers in the free-running timer mode.

The count value is captured to CR010 when the valid edge of the TI000 pin input is detected and to CR000 when the valid edge of the TI010 pin input is detected.

Figure 7-38. Timing Example of Free-Running Timer Mode (CR000: Capture Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H

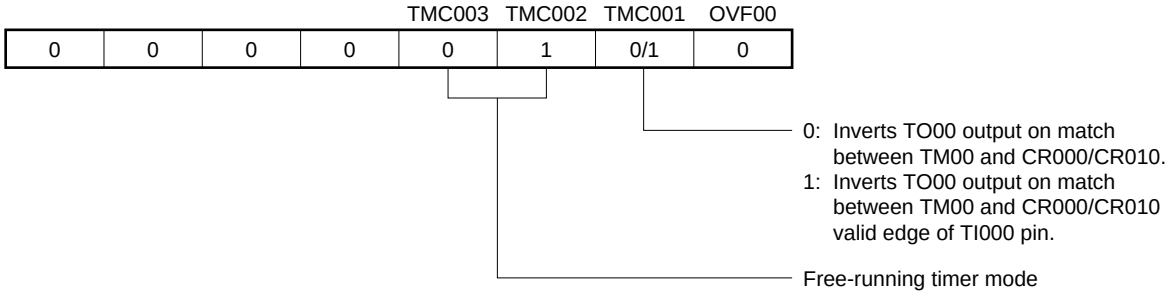


This is an application example where both the edges of the TI010 pin are detected and the count value is captured to CR000 in the free-running timer mode.

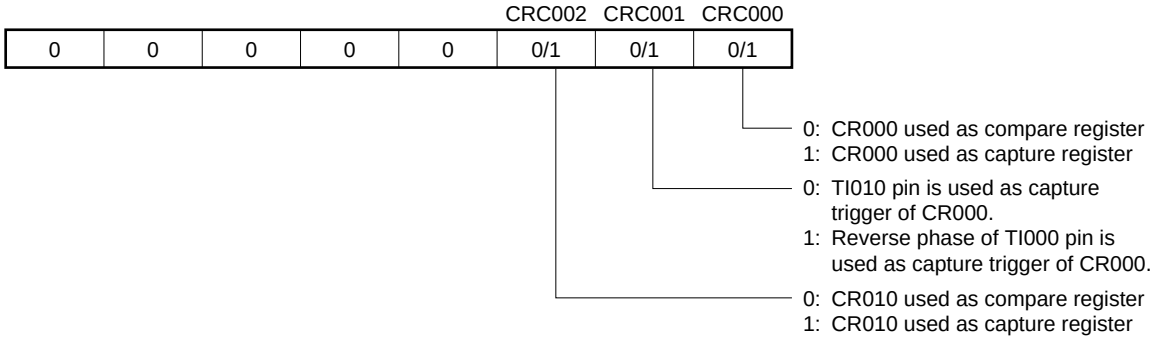
When both CR000 and CR010 are used as capture registers and when the valid edge of only the TI010 pin is to be detected, the count value cannot be captured to CR010.

Figure 7-39. Example of Register Settings in Free-Running Timer Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

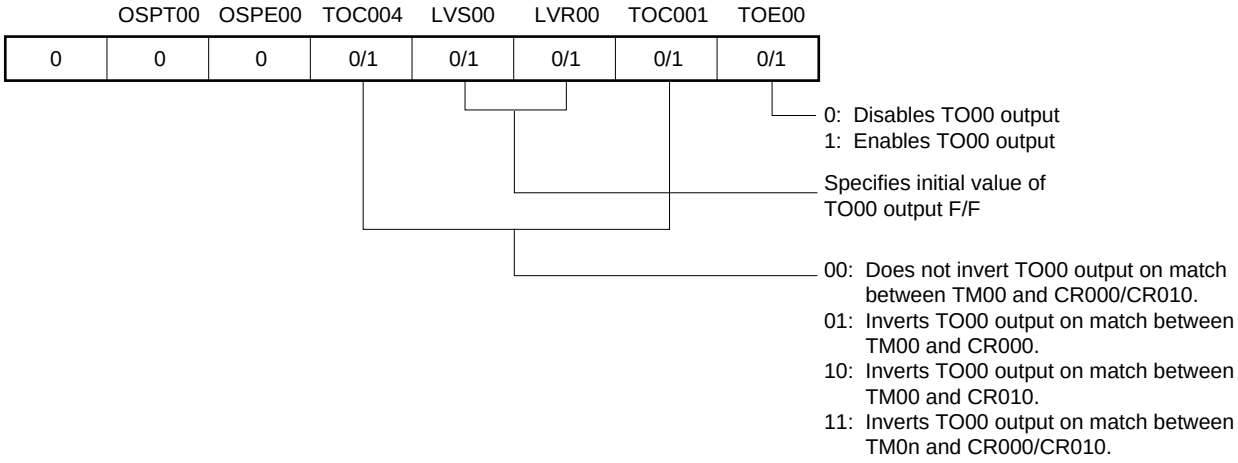
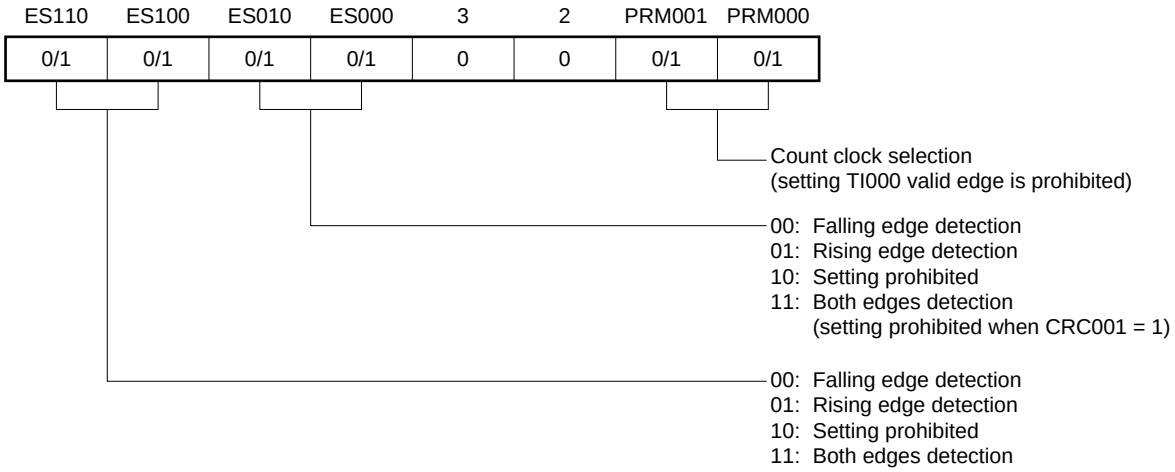


Figure 7-39. Example of Register Settings in Free-Running Timer Mode (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

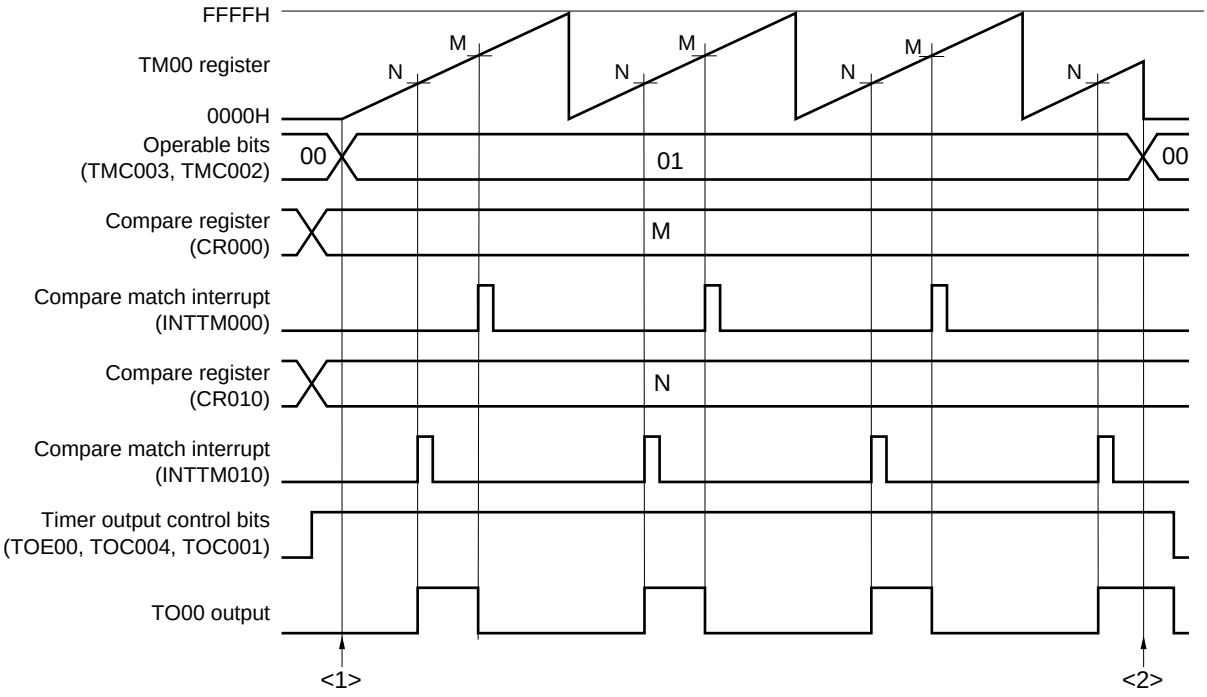
To use this register as a capture register, select either the TI000 or TI010 pin input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

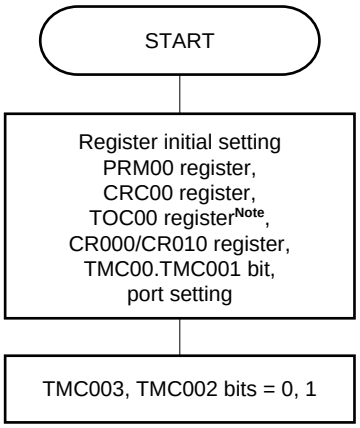
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

Figure 7-40. Example of Software Processing in Free-Running Timer Mode



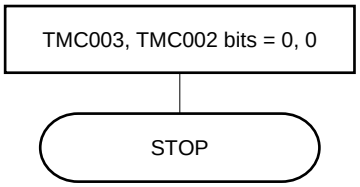
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 01.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Note Care must be exercised when setting TOC00. For details, refer to **7.3 (3) 16-bit timer output control register 00 (TOC00)**.

7.4.6 PPG output operation

A square wave having a pulse width set in advance by CR010 is output from the TO00 pin as a PPG (Programmable Pulse Generator) signal during a cycle set by CR000 when bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11 (clear & start upon a match between TM00 and CR000).

The pulse cycle and duty factor of the pulse generated as the PPG output are as follows.

- Pulse cycle = (Set value of CR000 + 1) × Count clock cycle
- Duty = (Set value of CR010 + 1) / (Set value of CR000 + 1)

Caution To change the duty factor (value of CR010) during operation, refer to 7.5.1 Rewriting CR010 during TM00 operation.

Remarks 1. For the setting of I/O pins, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 7-41. Block Diagram of PPG Output Operation

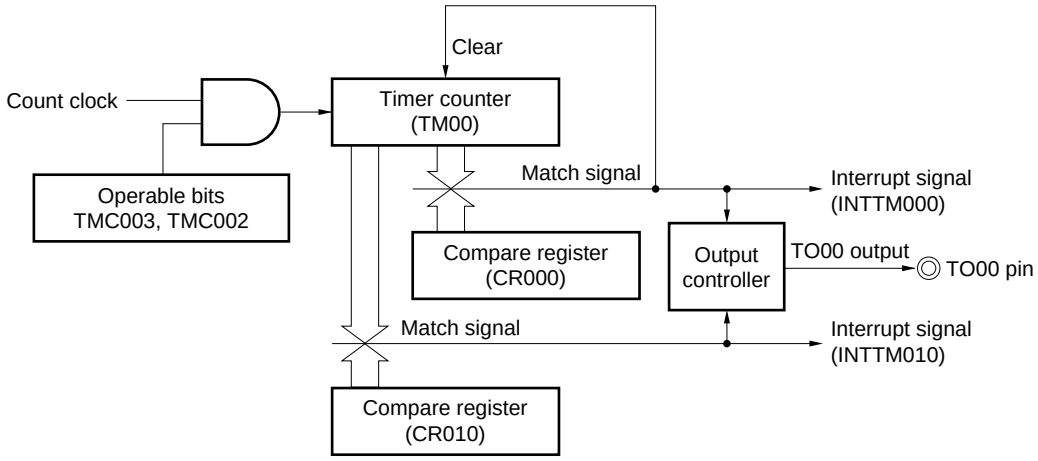
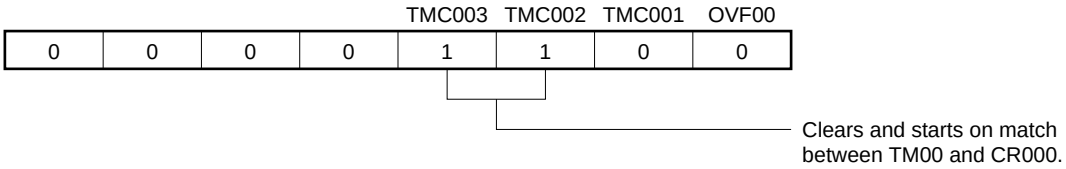
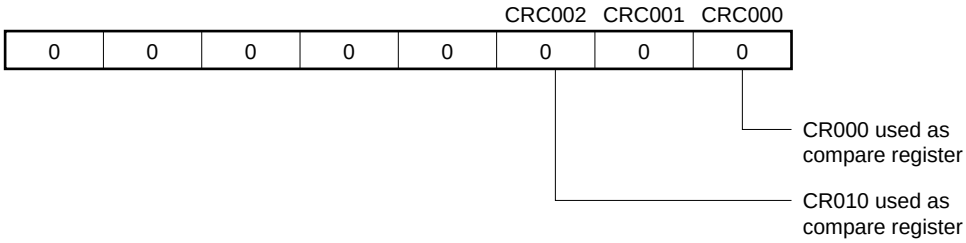


Figure 7-42. Example of Register Settings for PPG Output Operation (1/2)

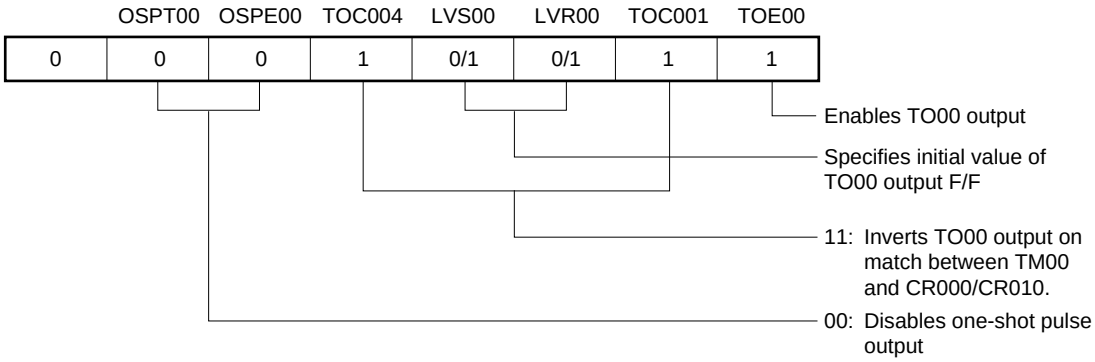
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

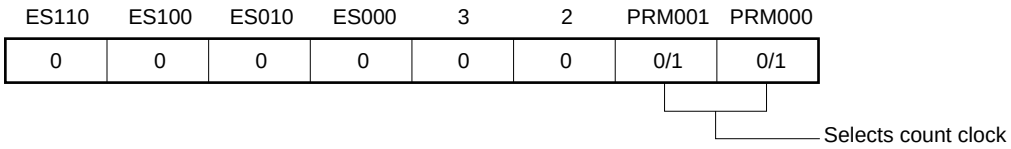


Figure 7-42. Example of Register Settings for PPG Output Operation (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

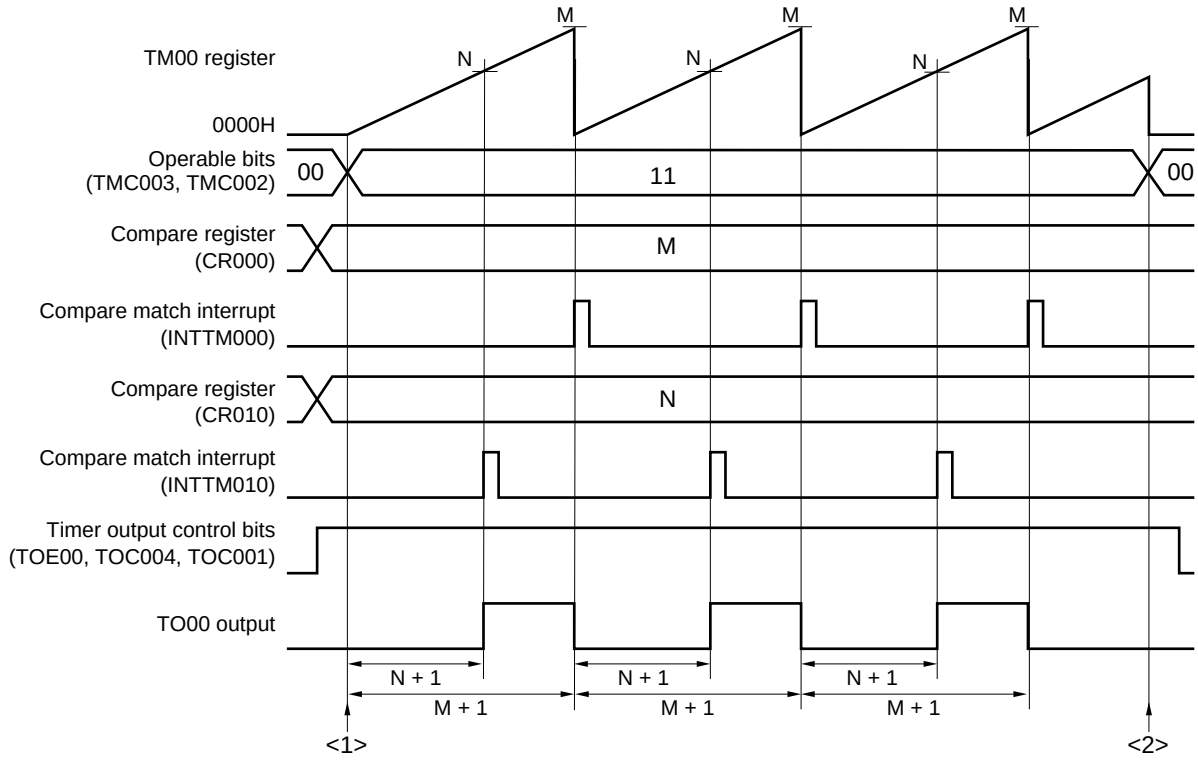
An interrupt signal (INTTM000) is generated when the value of this register matches the count value of TM00. The count value of TM00 is cleared.

(g) 16-bit capture/compare register 010 (CR010)

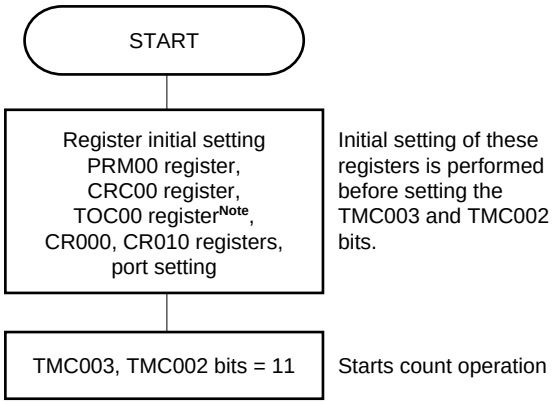
An interrupt signal (INTTM010) is generated when the value of this register matches the count value of TM00. The count value of TM00 is not cleared.

Caution Set values to CR000 and CR010 such that the condition 0000H ≤ CR010 < CR000 ≤ FFFFH is satisfied.

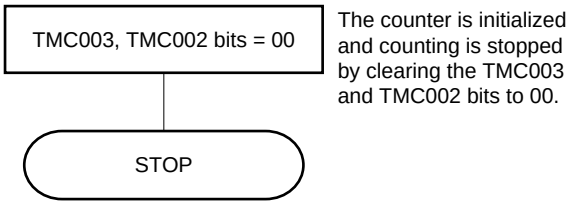
Figure 7-43. Example of Software Processing for PPG Output Operation



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, refer to **7.3 (3) 16-bit timer output control register 00 (TOC00)**.

Remarks PPG pulse cycle = (M + 1) × Count clock cycle
PPG duty = (N + 1)/(M + 1)

7.4.7 One-shot pulse output operation

A one-shot pulse can be output by setting bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register 00 (TMC00) to 01 (free-running timer mode) or to 10 (clear & start mode entered by the TI000 pin valid edge) and setting bit 5 (OSPE00) of 16-bit timer output control register 00 (TOC00) to 1.

When bit 6 (OSPT00) of TOC00 is set to 1 or when the valid edge is input to the TI000 pin during timer operation, clearing & starting of TM00 is triggered, and a pulse of the difference between the values of CR000 and CR010 is output only once from the TO00 pin.

Cautions 1. Do not input the trigger again (setting OSPT00 to 1 or detecting the valid edge of the TI000 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 OSPT00 to 1 as the trigger of one-shot pulse output, do not change the level of the TI000 pin or its alternate function port pin. Otherwise, the pulse will be unexpectedly output.

Remarks 1. For the setting of the I/O pins, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 7-44. Block Diagram of One-Shot Pulse Output Operation

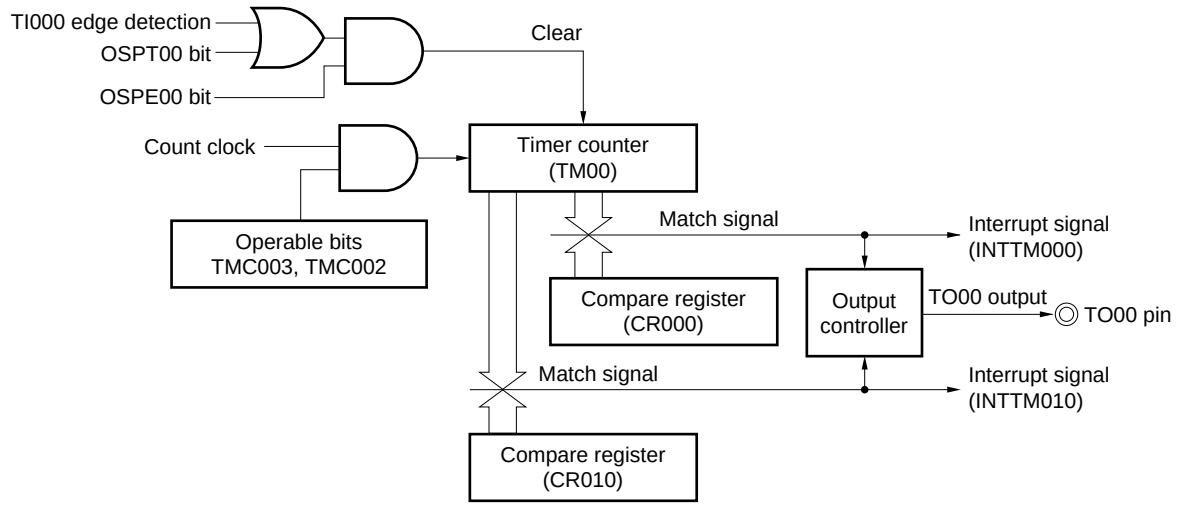
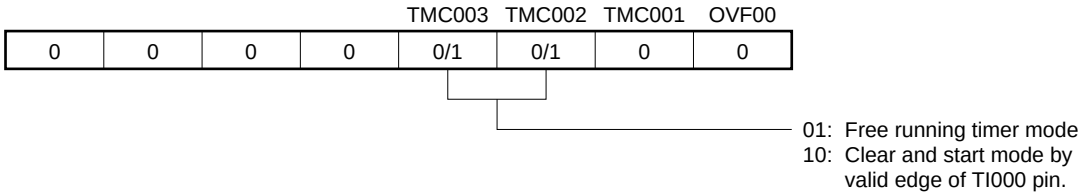
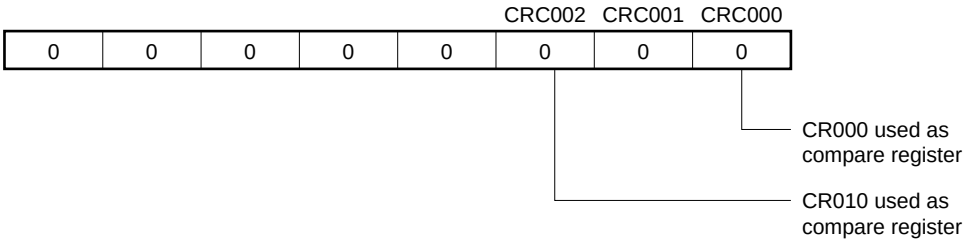


Figure 7-45. Example of Register Settings for One-Shot Pulse Output Operation (1/2)

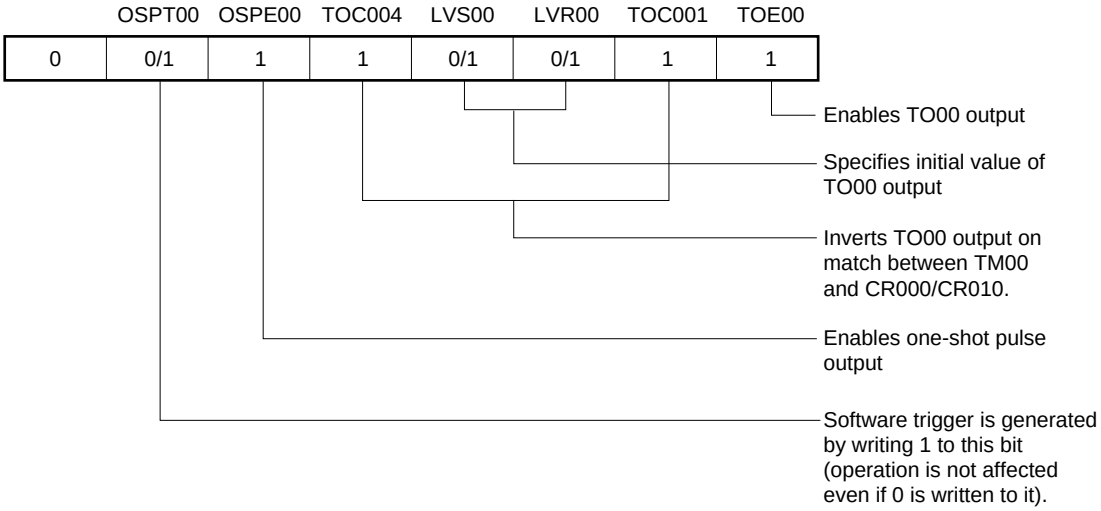
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

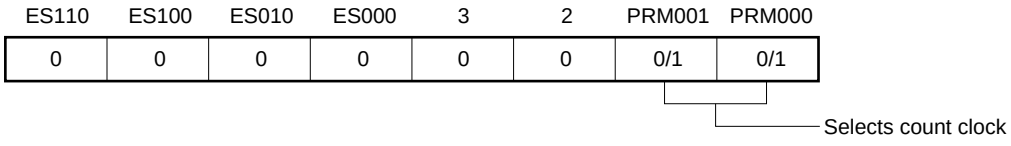


Figure 7-45. Example of Register Settings for One-Shot Pulse Output Operation (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

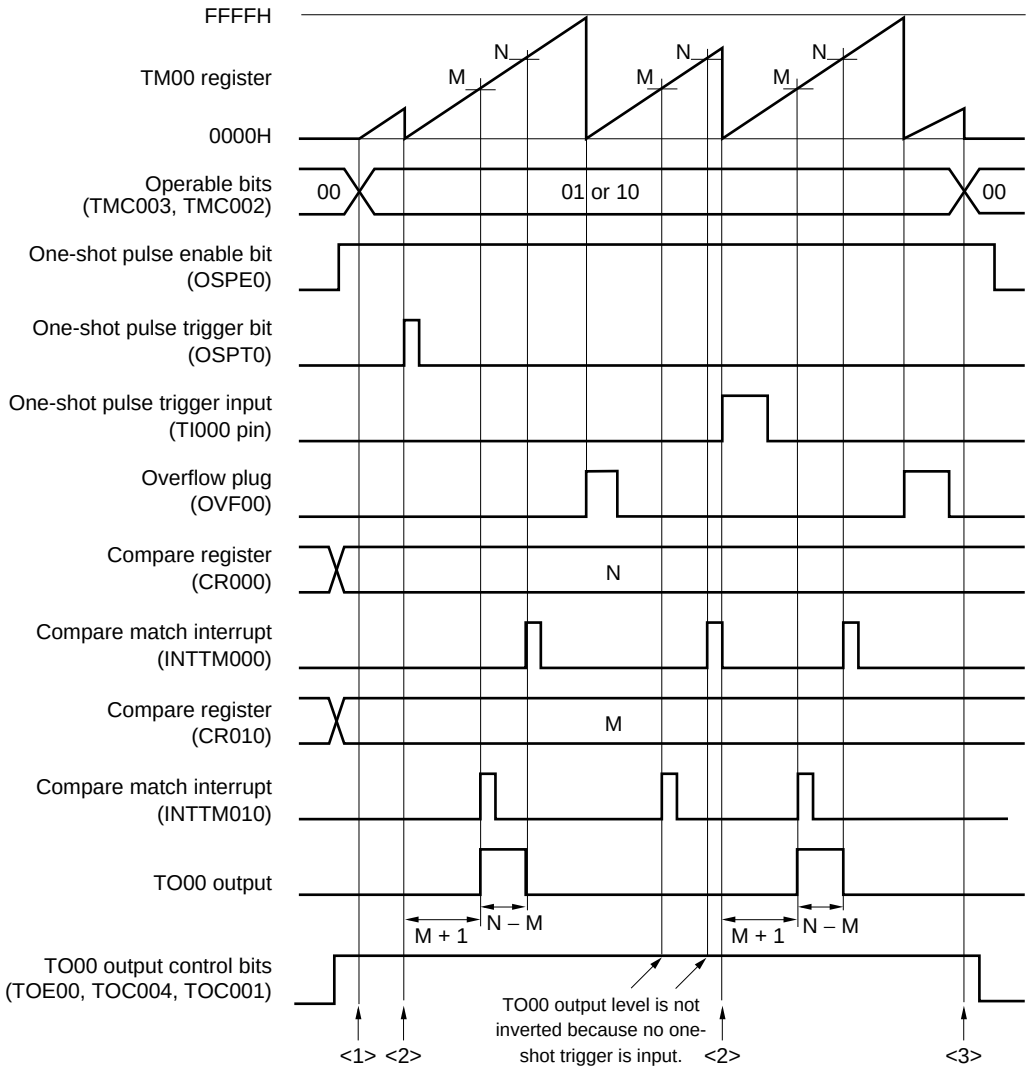
This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR000, an interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR010, an interrupt signal (INTTM010) is generated and the TO00 output level is inverted.

Caution Do not set the same value to CR000 and CR010.

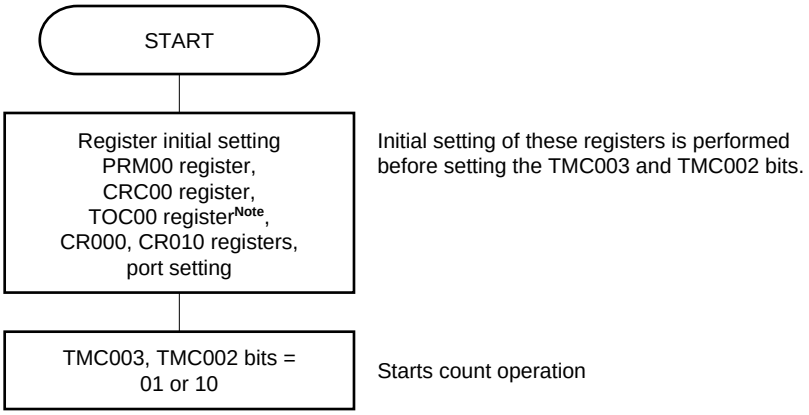
Figure 7-46. Example of Software Processing for One-Shot Pulse Output Operation (1/2)



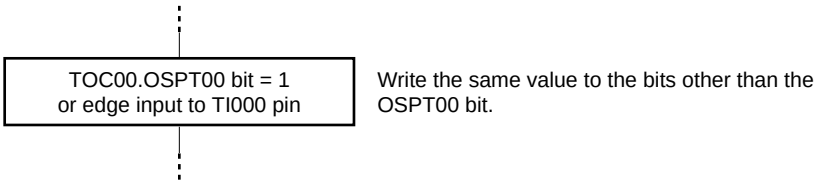
- Time from when the one-shot pulse trigger is input until the one-shot pulse is output
  = (M + 1) × Count clock cycle
- One-shot pulse output active level width
  = (N – M) × Count clock cycle

Figure 7-46. Example of Software Processing for One-Shot Pulse Output Operation (2/2)

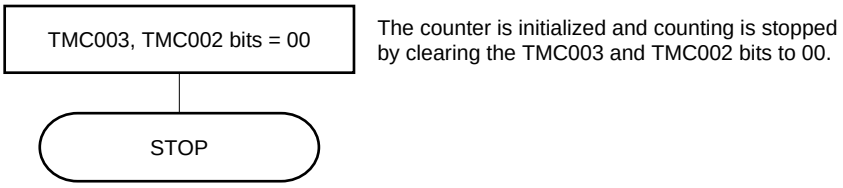
<1> Count operation start flow



<2> One-shot trigger input flow



<3> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, refer to **7.3 (3) 16-bit timer output control register 00 (TOC00)**.

7.4.8 Pulse width measurement operation

TM00 can be used to measure the pulse width of the signal input to the TI000 and TI010 pins.

Measurement can be accomplished by operating the 16-bit timer/event counter 00 in the free-running timer mode or by restarting the timer in synchronization with the signal input to the TI000 pin.

When an interrupt is generated, read the value of the valid capture register and measure the pulse width. Check bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00). If it is set (to 1), clear it to 0 by software.

Figure 7-47. Block Diagram of Pulse Width Measurement (Free-Running Timer Mode)

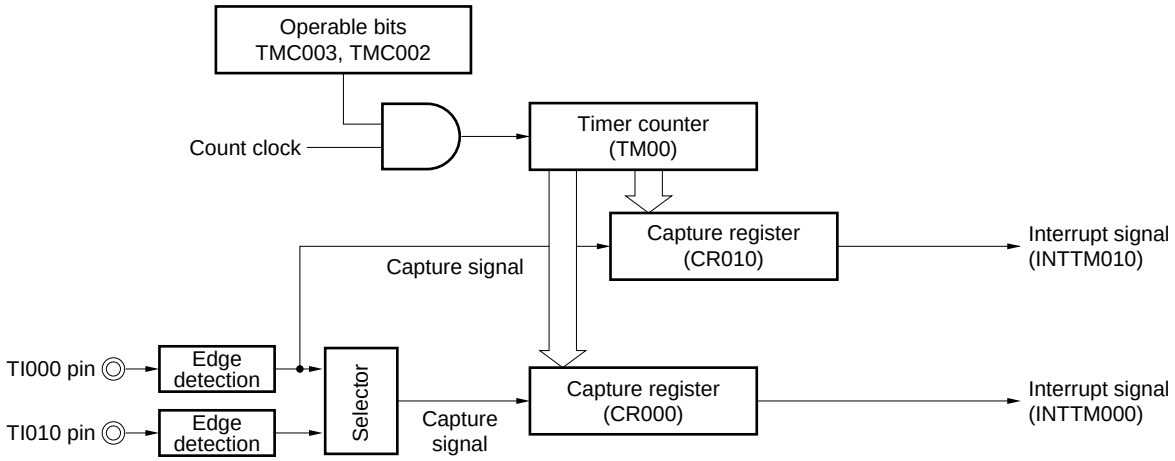
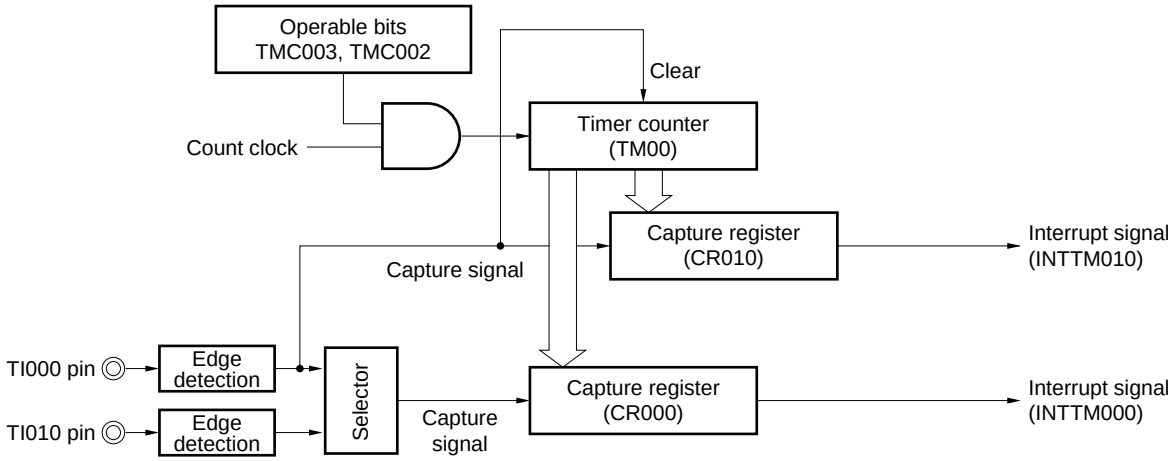


Figure 7-48. Block Diagram of Pulse Width Measurement
(Clear & Start Mode Entered by TI000 Pin Valid Edge Input)



A pulse width can be measured in the following three ways.

- Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Remarks 1. For the setting of the I/O pins, refer to **7.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

(1) Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)

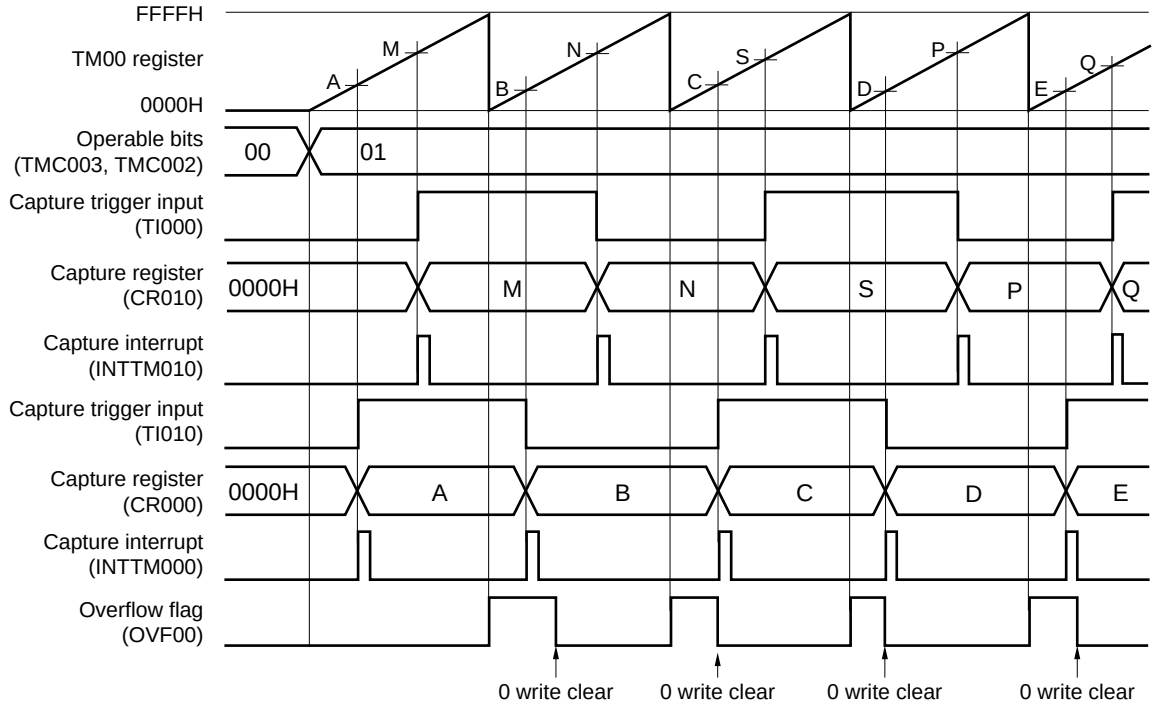
Set the free-running timer mode (TMC003 and TMC002 = 01). When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010. When the valid edge of the TI010 pin is detected, the count value of TM00 is captured to CR000. Specify detection of both the edges of the TI000 and TI010 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 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 7-49. Timing Example of Pulse Width Measurement (1)

• TMC00 = 04H, PRM00 = F0H, CRC00 = 05H



(2) Measuring the pulse width by using one input signal of the TI000 pin (free-running timer mode)

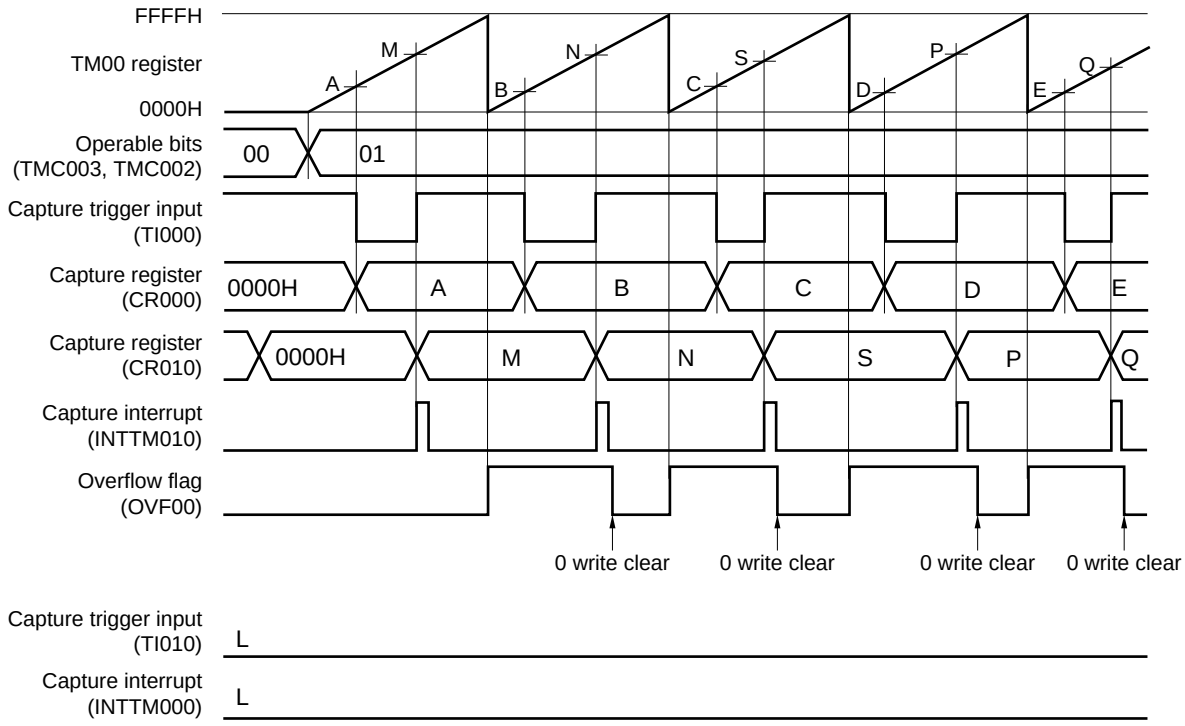
Set the free-running timer mode (TMC003 and TMC002 = 01). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge detected on the TI000 pin. When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010.

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 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 7-50. Timing Example of Pulse Width Measurement (2)

• TMC00 = 04H, PRM00 = 10H, CRC00 = 07H



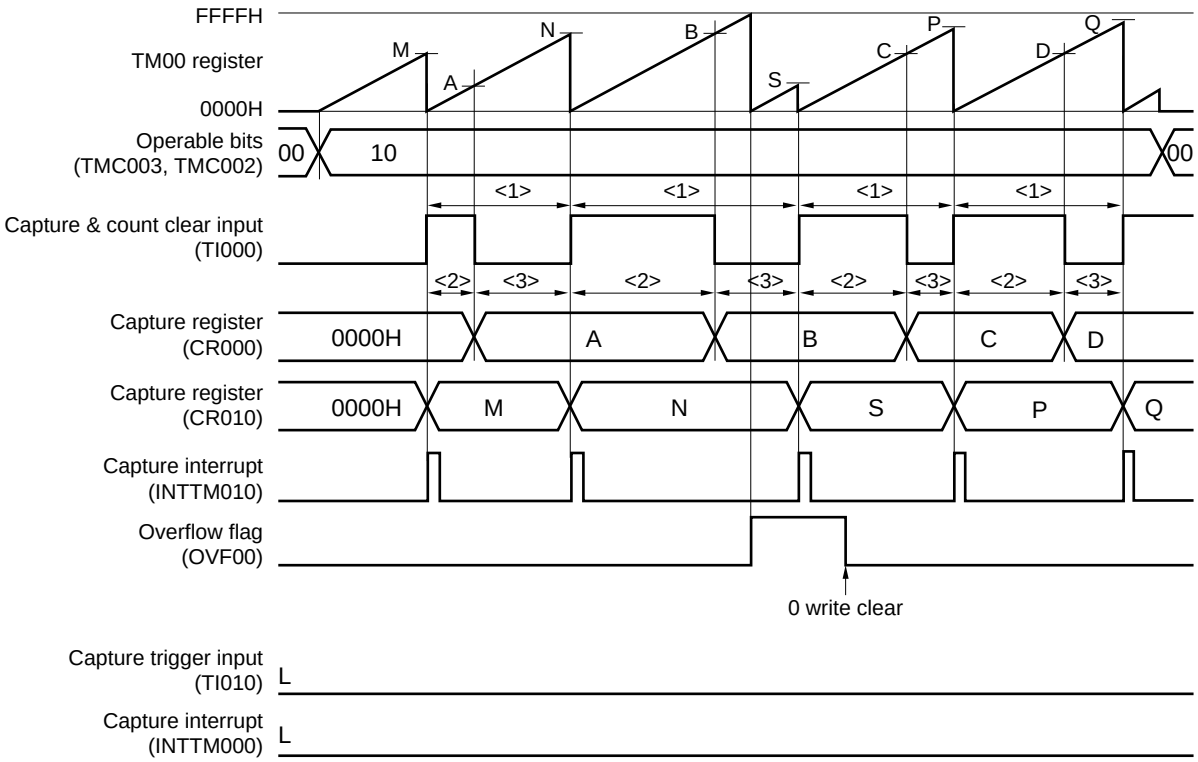
(3) Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Set the clear & start mode entered by the TI000 pin valid edge (TMC003 and TMC002 = 10). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge of the TI000 pin, and the count value of TM00 is captured to CR010 and TM00 is cleared (0000H) when the valid edge of the TI000 pin is detected. Therefore, a cycle is stored in CR010 if TM00 does not overflow.

If an overflow occurs, take the value that results from adding 10000H to the value stored in CR010 as a cycle. Clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 7-51. Timing Example of Pulse Width Measurement (3)

• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



<1> Pulse cycle = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR010) × Count clock cycle

<2> High-level pulse width = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR000) × Count clock cycle

<3> Low-level pulse width = (Pulse cycle – High-level pulse width)

Figure 7-52. Example of Register Settings for Pulse Width Measurement (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	0/1	0/1	0	0

01: Free running timer mode
10: Clear and start mode entered by valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	1	0/1	1

1: CR000 used as capture register

0: TI010 pin is used as capture trigger of CR000.
1: Reverse phase of TI000 pin is used as capture trigger of CR000.

1: CR010 used as capture register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)

ES110	ES100	ES010	ES000	3	2	PRM001	PRM000
0/1	0/1	0/1	0/1	0	0	0/1	0/1

Selects count clock
(setting valid edge of TI000 is prohibited)

00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection
(setting when CRC001 = 1 is prohibited)

00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection

Figure 7-52. Example of Register Settings for Pulse Width Measurement (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

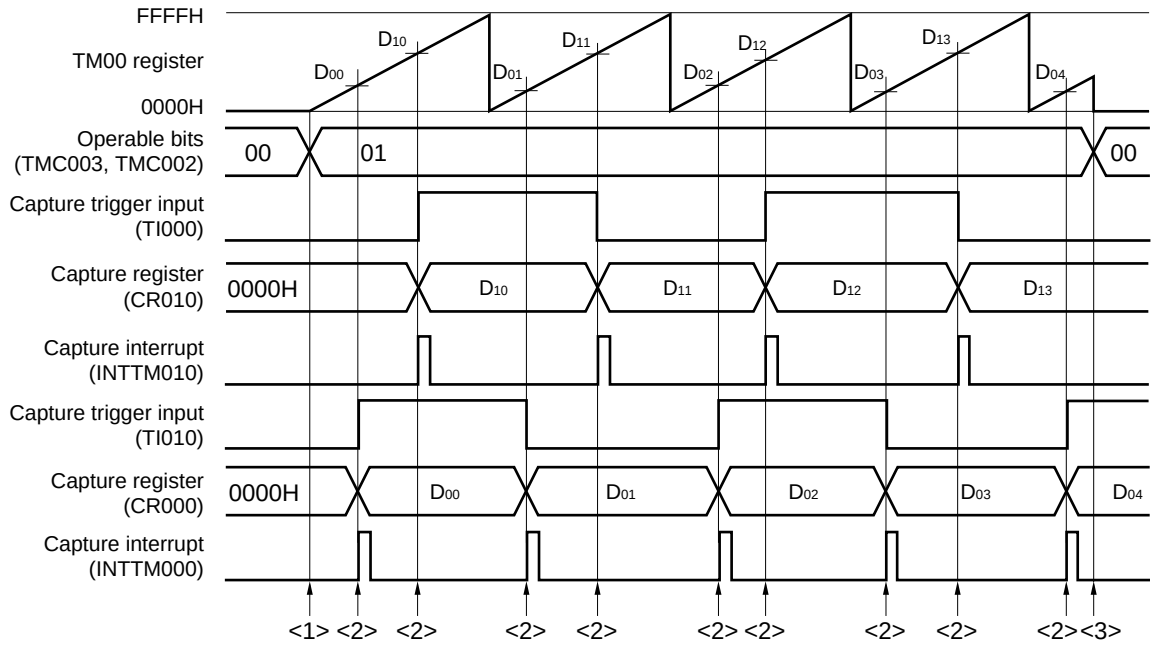
This register is used as a capture register. Either the TI000 or TI010 pin is selected as a capture trigger. When a specified edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a capture register. The signal input to the TI000 pin is used as a capture trigger. When the capture trigger is detected, the count value of TM00 is stored in CR010.

Figure 7-53. 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 TI000 pin valid edge

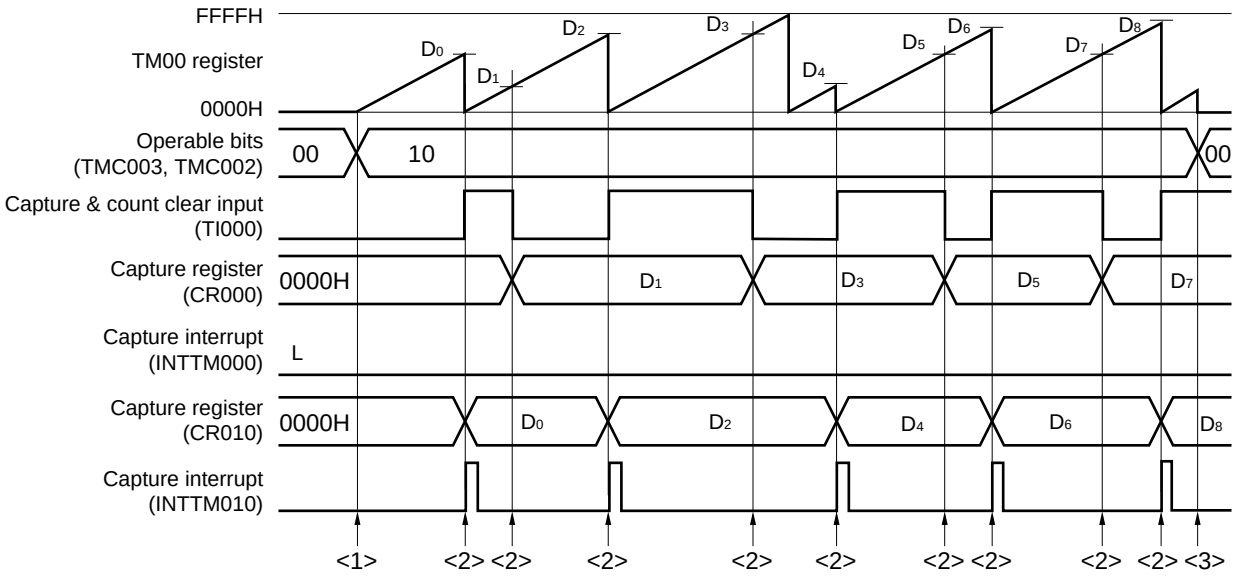
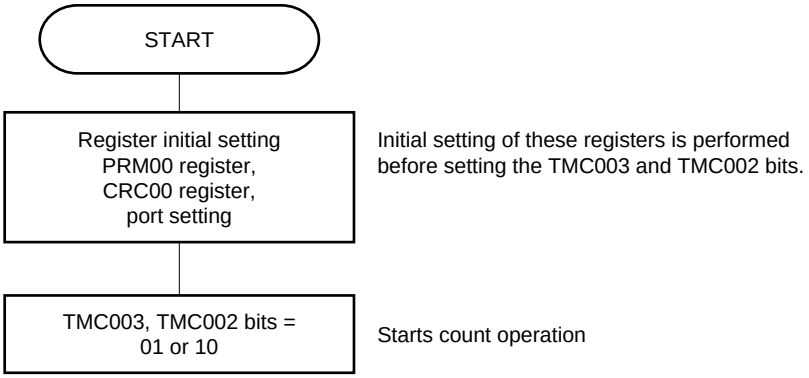
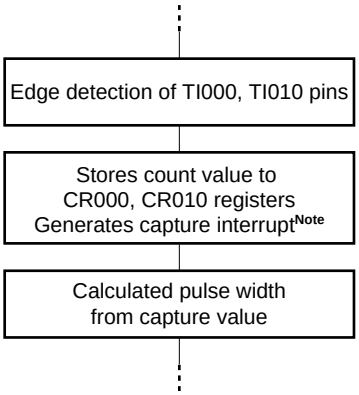


Figure 7-53. Example of Software Processing for Pulse Width Measurement (2/2)

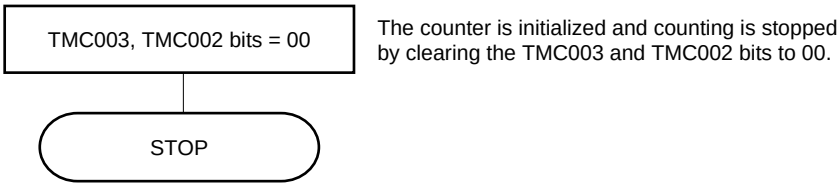
<1> Count operation start flow



<2> Capture trigger input flow



<3> Count operation stop flow



Note The capture interrupt signal (INTTM000) is not generated when the reverse-phase edge of the TI000 pin input is selected to the valid edge of CR000.

7.5 Special Use of TM00

7.5.1 Rewriting CR010 during TM00 operation

In principle, rewriting CR000 and CR010 of the 78K0/Fx2-L microcontrollers when they are used as compare registers is prohibited while TM00 is operating (TMC003 and TMC002 = other than 00).

However, the value of CR010 can be changed, even while TM00 is operating, using the following procedure if CR010 is used for PPG output and the duty factor is changed. (When changing the value of CR010 to a smaller value than the current one, rewrite it immediately after its value matches the value of TM00. When changing the value of CR010 to a larger value than the current one, rewrite it immediately after the values of CR000 and TM00 match. If the value of CR010 is rewritten immediately before a match between CR010 and TM00, or between CR000 and TM00, an unexpected operation may be performed.).

Procedure for changing value of CR010

<1> Disable interrupt INTTM010 (TMMK010 = 1).
<2> Disable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 0).
<3> Change the value of CR010.
<4> Wait for one cycle of the count clock of TM00.
<5> Enable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 1).
<6> Clear the interrupt flag of INTTM010 (TMIF010 = 0) to 0.
<7> Enable interrupt INTTM010 (TMMK010 = 0).

Remark For TMIF010 and TMMK010, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

7.5.2 Setting LVS00 and LVR00

(1) Usage of LVS00 and LVR00

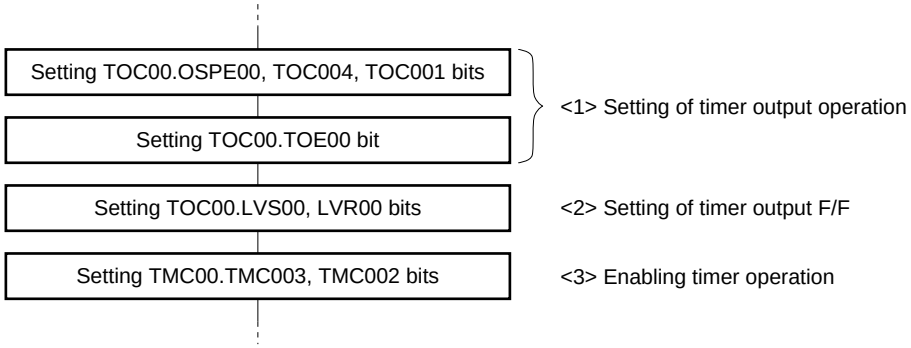
LVS00 and LVR00 are used to set the default value of the TO00 output and to invert the timer output without enabling the timer operation (TMC003 and TMC002 = 00). Clear LVS00 and LVR00 to 00 (default value: low-level output) when software control is unnecessary.

LVS00	LVR00	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

(2) Setting LVS00 and LVR00

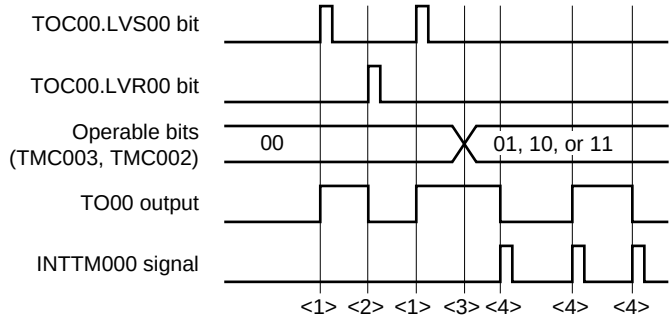
Set LVS00 and LVR00 using the following procedure.

Figure 7-54. Example of Flow for Setting LVS00 and LVR00 Bits



Caution Be sure to set LVS00 and LVR00 following steps <1>, <2>, and <3> above.
Step <2> can be performed after <1> and before <3>.

Figure 7-55. Timing Example of LVR00 and LVS00



<1> The TO00 output goes high when LVS00 and LVR00 = 10.

<2> The TO00 output goes low when LVS00 and LVR00 = 01 (the pin output remains unchanged from the high level even if LVS00 and LVR00 are cleared to 00).

<3> The timer starts operating when TMC003 and TMC002 are set to 01, 10, or 11. Because LVS00 and LVR00 were set to 10 before the operation was started, the TO00 output starts from the high level. After the timer starts operating, setting LVS00 and LVR00 is prohibited until TMC003 and TMC002 = 00 (disabling the timer operation).

<4> The TO00 output level is inverted each time an interrupt signal (INTTM000) is generated.

7.6 Cautions for 16-bit Timer/Event Counter 00

(1) Restrictions for each channel of 16-bit timer/event counter 00

Table 7-3 shows the restrictions for each channel.

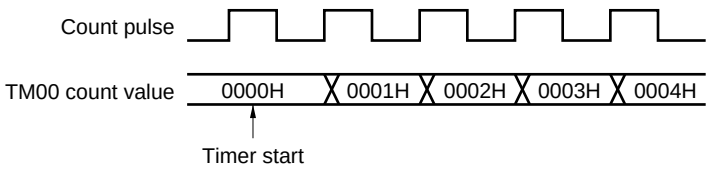
Table 7-3. Restrictions for Each Channel of 16-bit Timer/Event Counter 00

Operation	Restriction
As interval timer	–
As square-wave output	
As external event counter	
As clear & start mode entered by TI000 pin valid edge input	Using timer output (TO00) is prohibited when detection of the valid edge of the TI010 pin is used. (TOC00 = 00H)
As free-running timer	–
As PPG output	0000H ≤ CP010 < CR000 ≤ FFFFH
As one-shot pulse output	Setting the same value to CR000 and CP010 is prohibited.
As pulse width measurement	Using timer output (TO00) is prohibited (TOC00 = 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 TM00 is started asynchronously to the count pulse.

Figure 7-56. Start Timing of TM00 Count



(3) Setting of CR000 and CR010 (clear & start mode entered upon a match between TM00 and CR000)

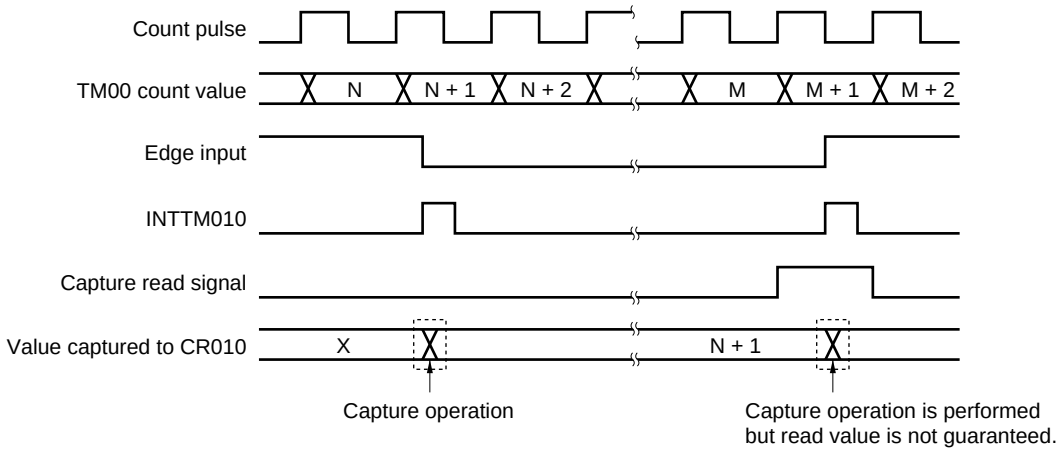
Set a value other than 0000H to CR000 and CR010 (TM00 cannot count one pulse when it is used as an external event counter).

(4) Timing of holding data by capture register

(a) When the valid edge is input to the TI000/TI010 pin and the reverse phase of the TI000 pin is detected while CR000/CR010 is read, CR010 performs a capture operation but the read value of CR000/CR010 is not guaranteed. At this time, an interrupt signal (INTTM000/INTTM010) is generated when the valid edge of the TI000/TI010 pin is detected (the interrupt signal is not generated when the reverse-phase edge of the TI000 pin is detected).

When the count value is captured because the valid edge of the TI000/TI010 pin was detected, read the value of CR000/CR010 after INTTM000/INTTM010 is generated.

Figure 7-57. Timing of Holding Data by Capture Register



(b) The values of CR000 and CR010 are not guaranteed after 16-bit timer/event counter 00 stops.

(5) Setting valid edge

Set the valid edge of the TI000 pin while the timer operation is stopped (TMC003 and TMC002 = 00). Set the valid edge by using ES000 and ES010.

(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.

(7) Operation of OVF00 flag

(a) Setting OVF00 flag (1)

The OVF00 flag is set to 1 in the following case, as well as when TM00 overflows.

Select the clear & start mode entered upon a match between TM00 and CR000.

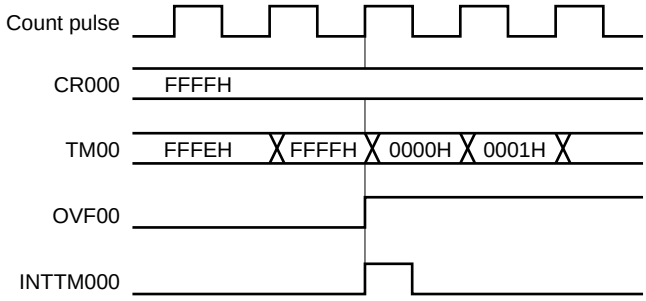
↓

Set CR000 to FFFFH.

↓

When TM00 matches CR000 and TM00 is cleared from FFFFH to 0000H

Figure 7-58. Operation Timing of OVF00 Flag



(b) Clearing OVF00 flag

Even if the OVF00 flag is cleared to 0 after TM00 overflows and before the next count clock is counted (before the value of TM00 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 TI000 pin valid edge. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.

(9) Capture operation

(a) When valid edge of TI000 is specified as count clock

When the valid edge of TI000 is specified as the count clock, the capture register for which TI000 is specified as a trigger does not operate correctly.

(b) Pulse width to accurately capture value by signals input to TI010 and TI000 pins

To accurately capture the count value, the pulse input to the TI000 and TI010 pins as a capture trigger must be wider than two count clocks selected by PRM00 (refer to **Figure 7-7**).

(c) Generation of interrupt signal

The capture operation is performed at the falling edge of the count clock but the interrupt signals (INTTM000 and INTTM010) are generated at the rising edge of the next count clock (refer to **Figure 7-7**).

(d) Note when CRC001 (bit 1 of capture/compare control register 00 (CRC00)) is set to 1

When the count value of the TM00 register is captured to the CR000 register in the phase reverse to the signal input to the TI000 pin, the interrupt signal (INTTM000) is not generated after the count value is captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. Mask the INTTM000 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 00 is enabled after reset and while the TI000 or TI010 pin is at high level and when the rising edge or both the edges are specified as the valid edge of the TI000 or TI010 pin, then the high level of the TI000 or TI010 pin is detected as the rising edge. Note this when the TI000 or TI010 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 TI000 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 PRM00 is used for sampling.

When the signal input to the TI000 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 (refer to **Figure 7-7**).

(11) Timer operation

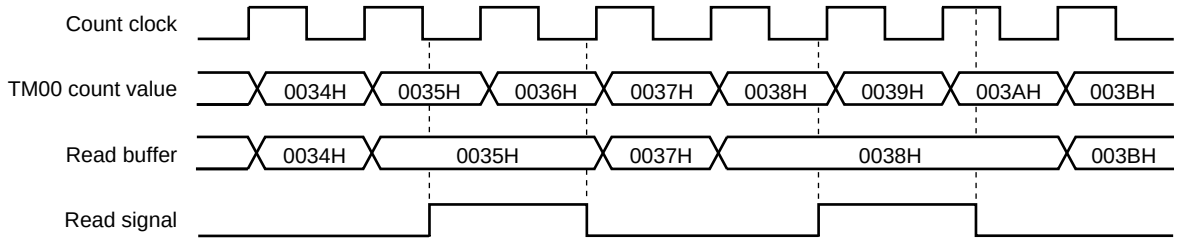
The signal input to the TI000/TI010 pin is not acknowledged while the timer is stopped, regardless of the operation mode of the CPU.

Remark f_{PRS}: Peripheral hardware clock frequency

(12) Reading of 16-bit timer counter 00 (TM00)

TM00 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 7-59. 16-bit Timer Counter 00 (TM00) Read Timing



CHAPTER 8 8-BIT TIMER/EVENT COUNTER 51

8.1 Functions of 8-bit Timer/Event Counter 51

8-bit timer/event counter 51 is mounted onto all 78K0/Fx2-L microcontroller products.
8-bit timer/event counter 51 has the following functions.

- Interval timer
- External event counter

8.2 Configuration of 8-bit Timer/Event Counter 51

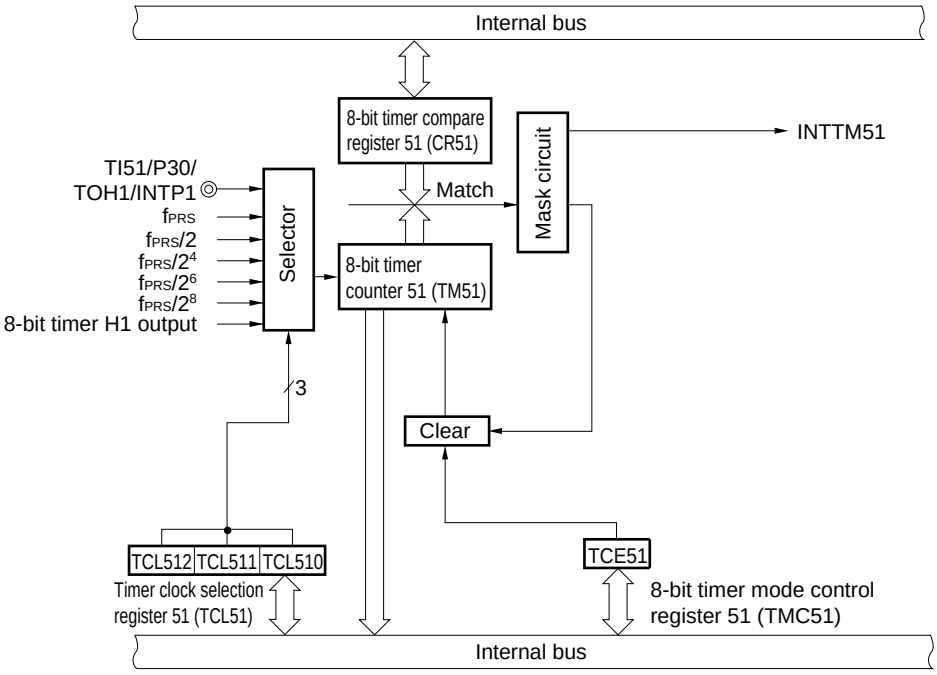
8-bit timer/event counter 51 includes the following hardware.

Table 8-1. Configuration of 8-bit Timer/Event Counter 51

Item	Configuration
Timer register	8-bit timer counter 51 (TM51)
Timer input	TI51
Register	8-bit timer compare register 51 (CR51)
Control registers	Timer clock selection register 51 (TCL51) 8-bit timer mode control register 51 (TMC51) Port mode register 3 (PM3)

Figure 8-1 shows the block diagram of 8-bit timer/event counter 51.

Figure 8-1. Block Diagram of 8-bit Timer/Event Counter 51



(1) 8-bit timer counter 51 (TM51)

TM51 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 8-2. Format of 8-bit Timer Counter 51 (TM51)

Address: FF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM51								

In the following situations, the count value is cleared to 00H.

<1> Reset signal generation

<2> When TCE51 is cleared

<3> When TM51 and CR51 match

(2) 8-bit timer compare register 51 (CR51)

CR51 can be read and written by an 8-bit memory manipulation instruction.

The value set in CR51 is constantly compared with the 8-bit timer counter 51 (TM51) count value, and an interrupt request (INTTM51) is generated if they match.

The value of CR51 can be set within 00H to FFH.

Reset signal generation clears CR51 to 00H.

Figure 8-3. Format of 8-bit Timer Compare Register 51 (CR51)

Address: FF41H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR51								

Caution Do not write other values to CR51 during operation.

8.3 Registers Controlling 8-bit Timer/Event Counter 51

The following three registers are used to control 8-bit timer/event counter 51.

- Timer clock selection register 51 (TCL51)
- 8-bit timer mode control register 51 (TMC51)
- Port mode register 3 (PM3)

(1) Timer clock selection register 51 (TCL51)

This register sets the count clock of 8-bit timer/event counter 51 and the valid edge of the TI51 pin input.

TCL51 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TCL51 to 00H.

Figure 8-4. 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 (when using PLL)
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	TMH1 output				

Note Do not start timer operation with the external clock from the TI51 pin 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 51 (TMC51)

TMC51 is a register that controls the 8-bit timer counter 51 (TM51) count operation.

TMC51 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-5. Format of 8-bit Timer Mode Control Register 51 (TMC51)

Address: FF43H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TMC51	TCE51	0	0	0	0	0	0	0

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

(3) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the TI51/P30/TOH1/INTP1 pin for timer input, set PM30 to 1. The output latch of P30 at this time may be 0 or 1.

PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 8-6. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode register 3 of the 78K0/FB2-L. For the format of port mode register 3 of other products, refer to **(1) Port mode registers (PMxx)** in **4.3 Registers Controlling Port Function**.

8.4 Operations of 8-bit Timer/Event Counter 51

8.4.1 Operation as interval timer

8-bit timer/event counter 51 operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 51 (CR51).

When the count value of 8-bit timer counter 51 (TM51) matches the value set to CR51, counting continues with the TM51 value cleared to 0 and an interrupt request signal (INTTM51) is generated.

The count clock of TM51 can be selected with bits 0 to 2 (TCL510 to TCL512) of timer clock selection register 51 (TCL51).

Setting

<1> Set the registers.
- TCL51: Select the count clock.
- CR51: Compare value
- TMC51: Stop the count operation.
  (TMC51 = 00000000B)

<2> After TCE51 = 1 is set, the count operation starts.

<3> If the values of TM51 and CR51 match, INTTM51 is generated (TM51 is cleared to 00H).

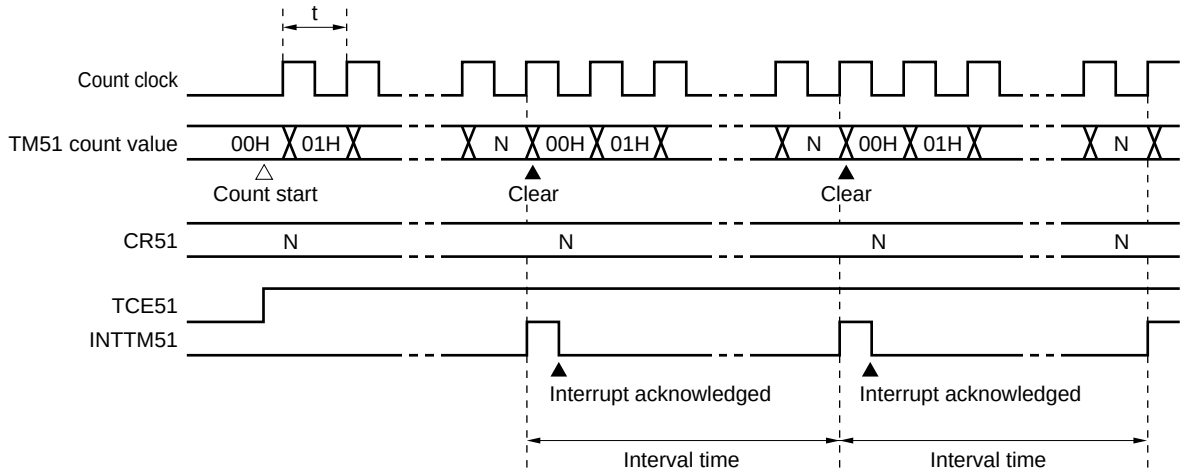
<4> INTTM51 is generated repeatedly at the same interval.
Set TCE51 to 0 to stop the count operation.

Caution Do not write other values to CR51 during operation.

Remark For how to enable the INTTM51 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

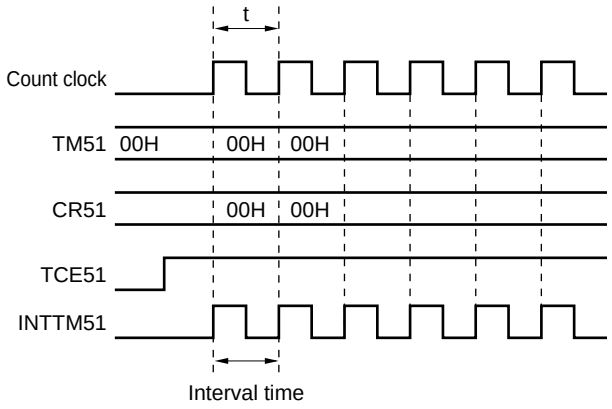
Figure 8-7. Interval Timer Operation Timing

(a) Basic operation

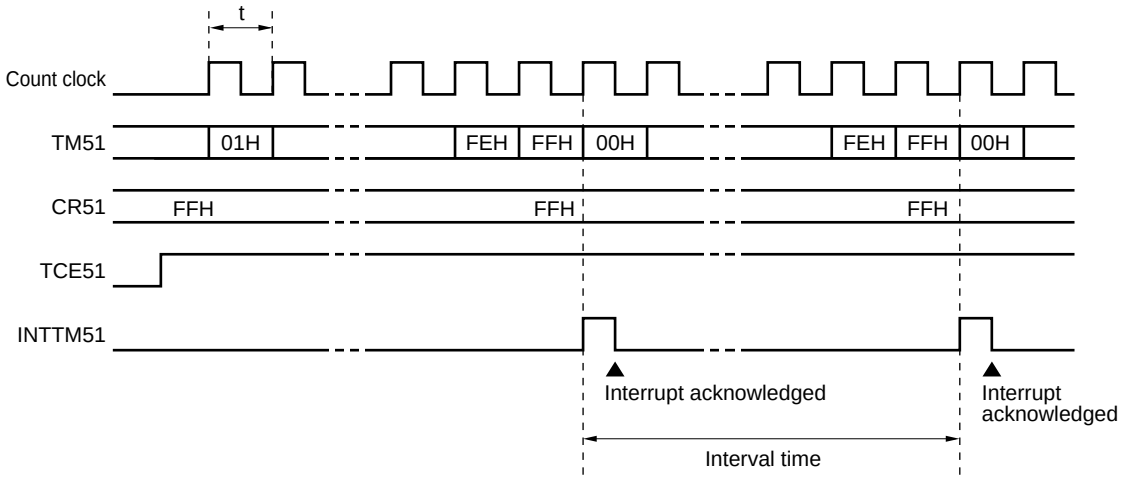


Remark Interval time = (N + 1) × t, N = 01H to FFH

(b) When CR51 = 00H



(c) When CR51 = FFH



8.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TI51 pin by 8-bit timer counter 51 (TM51).

TM51 is incremented each time the valid edge specified by timer clock selection register 51 (TCL51) is input. Either the rising or falling edge can be selected.

When the TM51 count value matches the value of 8-bit timer compare register 51 (CR51), TM51 is cleared to 0 and an interrupt request signal (INTTM51) is generated.

Whenever the TM51 value matches the value of CR51, INTTM51 is generated.

Setting

<1> Set each register.

- Set the port mode register (PM30) to 1.
- TCL51: Select TI51 pin input edge.
  TI51 pin falling edge → TCL51 = 00H
  TI51 pin rising edge → TCL51 = 01H
- CR51: Compare value
- TMC51: Stop the count operation. (TMC51 = 00000000B)

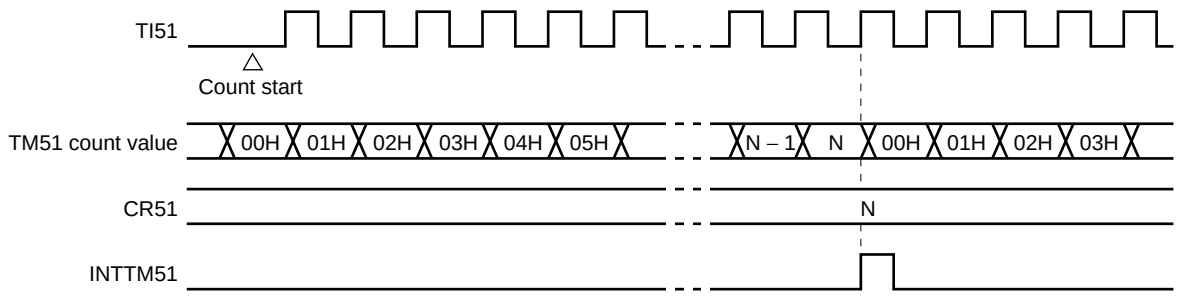
<2> When TCE51 = 1 is set, the number of pulses input from the TI51 pin is counted.

<3> When the values of TM51 and CR51 match, INTTM51 is generated (TM51 is cleared to 00H).

<4> After these settings, INTTM51 is generated each time the values of TM51 and CR51 match.

Remark For how to enable the INTTM51 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 8-8. External Event Counter Operation Timing (with Rising Edge Specified)



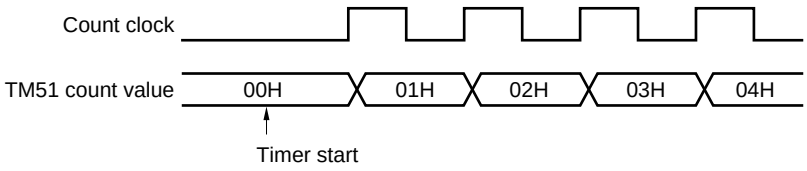
Remark N = 00H to FFH

8.5 Cautions for 8-bit Timer/Event Counter 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 counter 51 (TM51) are started asynchronously to the count clock.

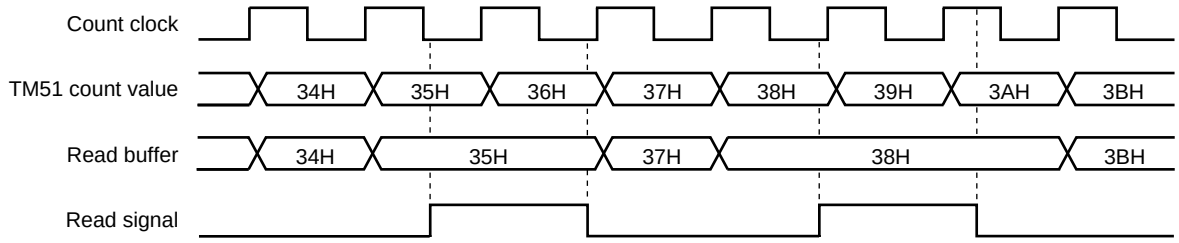
Figure 8-9. 8-bit Timer Counter 51 (TM51) Start Timing



(2) Reading of 8-bit timer counter 51 (TM51)

TM51 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 8-10. 8-bit Timer Counter 51 (TM51) Read Timing



CHAPTER 9 8-BIT TIMER H1

9.1 Functions of 8-bit Timer H1

8-bit timer H1 is mounted onto all 78K0/Fx2-L microcontroller products.
8-bit timer H1 has the following functions.

- Interval timer
- Square-wave output
- PWM output
- Carrier generator

9.2 Configuration of 8-bit Timer H1

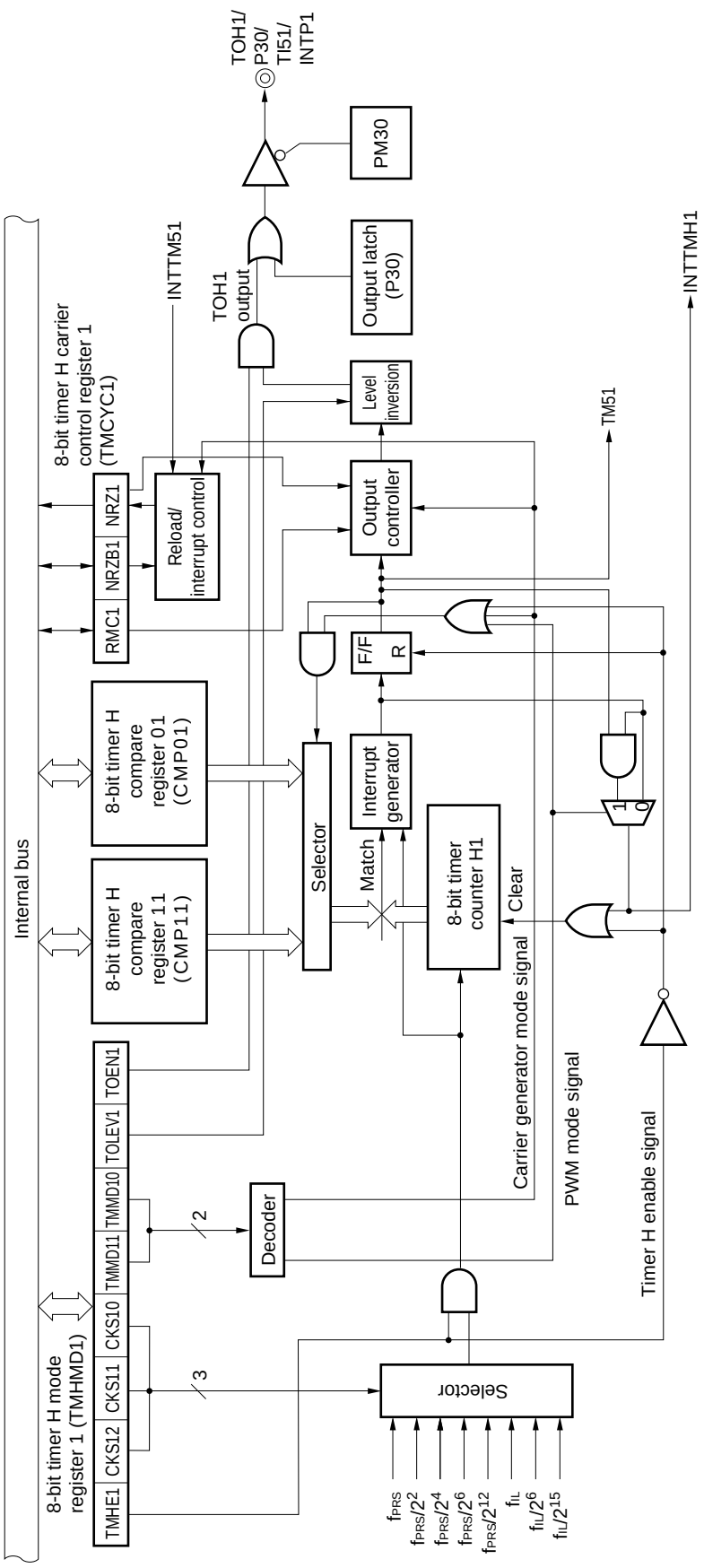
8-bit timer H1 includes the following hardware.

Table 9-1. Configuration of 8-bit Timer H1

Item	Configuration
Timer register	8-bit timer counter H1
Registers	8-bit timer H compare register 01 (CMP01) 8-bit timer H compare register 11 (CMP11)
Timer output	TOH1, output controller
Control registers	8-bit timer H mode register 1 (TMHMD1) 8-bit timer H carrier control register 1 (TMCYC1) Port mode register 3 (PM3) Port register 3 (P3)

Figure 9-1 shows the block diagram.

Figure 9-1. Block Diagram of 8-bit Timer H1



(1) 8-bit timer H compare register 01 (CMP01)

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 CMP01 with the count value of the 8-bit timer counter H1 and, when the two values match, generates an interrupt request signal (INTTMH1) and inverts the output level of TOH1.

Rewrite the value of CMP01 while the timer is stopped (TMHE1 = 0).

A reset signal generation clears this register to 00H.

Figure 9-2. Format of 8-bit Timer H Compare Register 01 (CMP01)

Address: FF1AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP01								

Caution CMP01 cannot be rewritten during timer count operation. CMP01 can be refreshed (the same value is written) during timer count operation.

(2) 8-bit timer H compare register 11 (CMP11)

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 CMP11 with the count value of the 8-bit timer counter H1 and, when the two values match, inverts the output level of TOH1. No interrupt request signal is generated.

In the carrier generator mode, the CMP11 register always compares the value set to CMP11 with the count value of the 8-bit timer counter H1 and, when the two values match, generates an interrupt request signal (INTTMH1). At the same time, the count value is cleared.

CMP11 can be refreshed (the same value is written) and rewritten during timer count operation.

If the value of CMP11 is rewritten while the timer is operating, the new value is latched and transferred to CMP11 when the count value of the timer matches the old value of CMP11, and then the value of CMP11 is changed to the new value. If matching of the count value and the CMP11 value and writing a value to CMP11 conflict, the value of CMP11 is not changed.

A reset signal generation clears this register to 00H.

Figure 9-3. Format of 8-bit Timer H Compare Register 11 (CMP11)

Address: FF1BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP11								

Caution In the PWM output mode and carrier generator mode, be sure to set 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).

9.3 Registers Controlling 8-bit Timer H1

The following four registers are used to control 8-bit timer H1.

- 8-bit timer H mode register 1 (TMHMD1)
- 8-bit timer H carrier control register 1 (TMCYC1)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) 8-bit timer H mode register 1 (TMHMD1)

This register controls the mode of 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 9-4. 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 (when using PLL)
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	$f_{IL}/2^6$	0.47 kHz (TYP.)			
1	1	0	$f_{IL}/2^{15}$	0.92 Hz (TYP.)			
1	1	1	f_{IL}	30 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 output of the TOH1/P30/TI51/INTP1 pin is determined by PM30 and P30, in addition to the TOH1 output.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{IL}: 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 9-5. 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 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the TOH1/P30/TI51/INTP1 pin for timer output, clear PM30 and the output latch of P30 to 0.

PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 9-6. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode register 3 of the 78K0/FB2-L. For the format of port mode register 3 of other products, refer to **(1) Port mode registers (PMxx)** in **4.3 Registers Controlling Port Function**.

9.4 Operation of 8-bit Timer H1

9.4.1 Operation as interval timer/square-wave output

When the 8-bit timer counter H1 and compare register 01 (CMP01) match, an interrupt request signal (INTTMH1) is generated and the 8-bit timer counter H1 is cleared to 00H.

Compare register 11 (CMP11) is not used in interval timer mode. Since a match of the 8-bit timer counter H1 and the CMP11 register is not detected even if the CMP11 register is set, timer output is not affected.

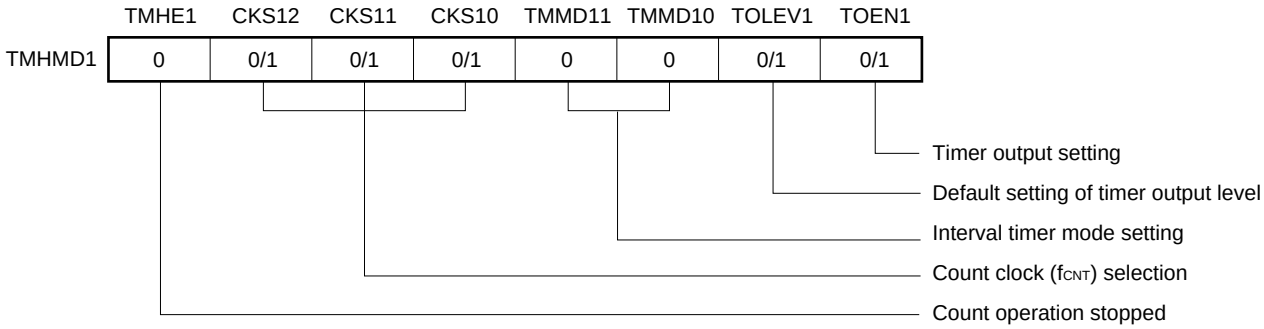
By setting bit 0 (TOEN1) of timer H mode register n (TMHMD1) to 1, a square wave of any frequency (duty = 50%) is output from TOH1.

Setting

<1> Set each register.

Figure 9-7. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register 1 (TMHMD1)



(ii) CMP01 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 TMHE1 = 1.

<3> When the values of the 8-bit timer counter H1 and the CMP01 register match, the INTTMH1 signal is generated and the 8-bit timer counter H1 is cleared to 00H.

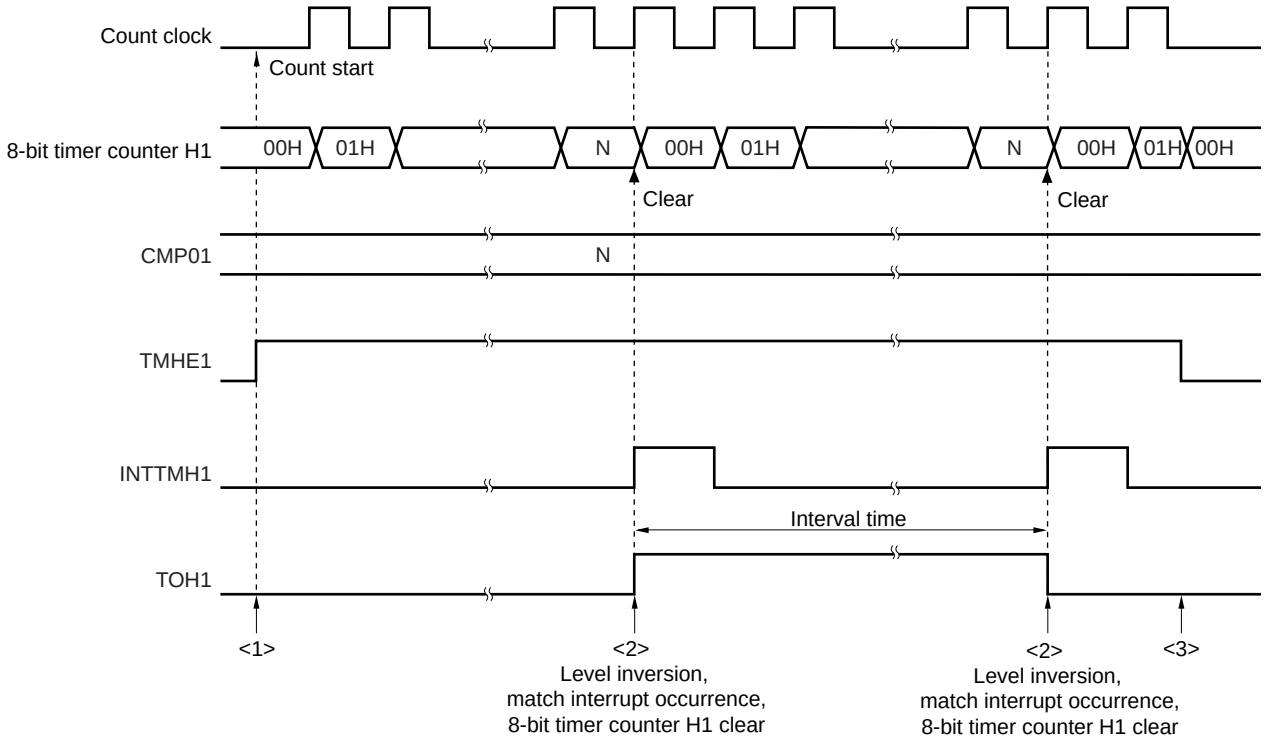
<4> Subsequently, the INTTMH1 signal is generated at the same interval. To stop the count operation, clear TMHE1 to 0.

Remarks 1. For the setting of the output pin, refer to **9.3 (3) Port mode register 3 (PM3)**.

2. For how to enable the INTTMH1 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 9-8. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation (Operation When 01H ≤ CMP01 ≤ FEH)



<1> The count operation is enabled by setting the TMHE1 bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

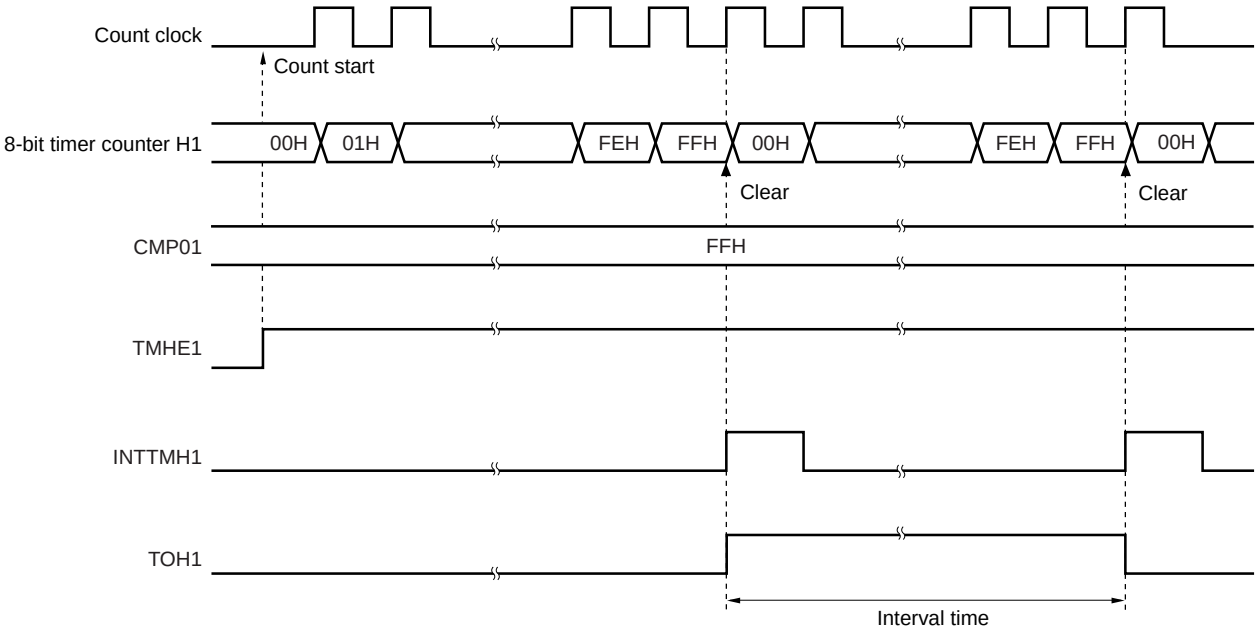
<2> When the value of the 8-bit timer counter H1 matches the value of the CMP01 register, the value of the timer counter is cleared, and the level of the TOH1 output is inverted. In addition, the INTTMH1 signal is output at the rising edge of the count clock.

<3> If the TMHE1 bit is cleared to 0 while timer H is operating, the INTTMH1 signal and TOH1 output are set to the default level. If they are already at the default level before the TMHE1 bit is cleared to 0, then that level is maintained.

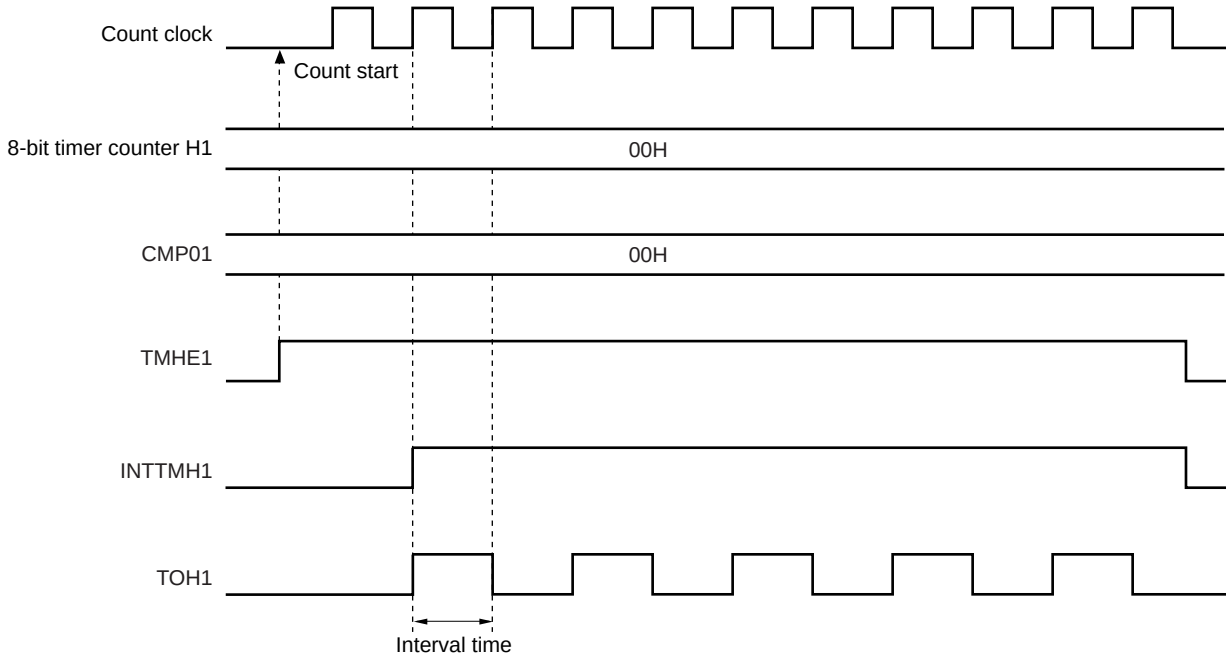
Remark 01H ≤ N ≤ FEH

Figure 9-8. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP01 = FFH



(c) Operation when CMP01 = 00H



9.4.2 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 01 (CMP01) controls the cycle of timer output (TOH1). Rewriting the CMP01 register during timer operation is prohibited.

The 8-bit timer compare register 11 (CMP11) controls the duty of timer output (TOH1). Rewriting the CMP11 register during timer operation is possible.

The operation in PWM output mode is as follows.

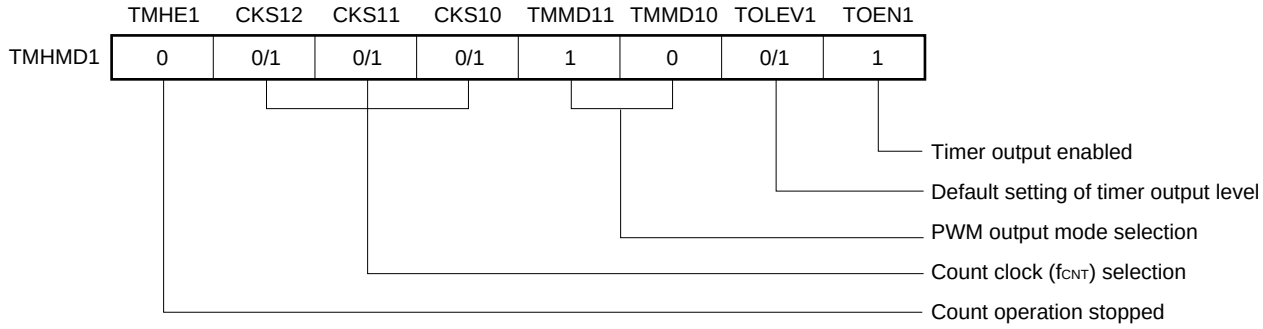
PWM output (TOH1 output) outputs an active level and 8-bit timer counter H1 is cleared to 0 when 8-bit timer counter H1 and the CMP01 register match after the timer count is started. PWM output (TOH1 output) outputs an inactive level when 8-bit timer counter H1 and the CMP11 register match.

Setting

<1> Set each register.

Figure 9-9. Register Setting in PWM Output Mode

(i) Setting timer H mode register 1 (TMHMD1)



(ii) Setting CMP01 register

- Compare value (N): Cycle setting

(iii) Setting CMP11 register

- Compare value (M): Duty setting

Remark 00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH

<2> The count operation starts when TMHE1 = 1.

<3> The CMP01 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 H1 and the CMP01 register match, the 8-bit timer counter H1 is cleared, an interrupt request signal (INTTMH1) is generated, and an active level is output. At the same time, the compare register to be compared with the 8-bit timer counter H1 is changed from the CMP01 register to the CMP11 register.

<4> When the 8-bit timer counter H1 and the CMP11 register match, an inactive level is output and the compare register to be compared with the 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register. At this time, the 8-bit timer counter H1 is not cleared and the INTTMH1 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 TMHE1 = 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 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 CMP11 register can be changed while the timer counter is operating. However, this takes a duration of three operating clocks (signal selected by the CKS12 to CKS10 bits of the TMHMD1 register) from when the value of the CMP11 register is changed until the value is transferred to the register.

2. 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).

3. Make sure that the CMP11 register setting value (M) and CMP01 register setting value (N) are within the following range.

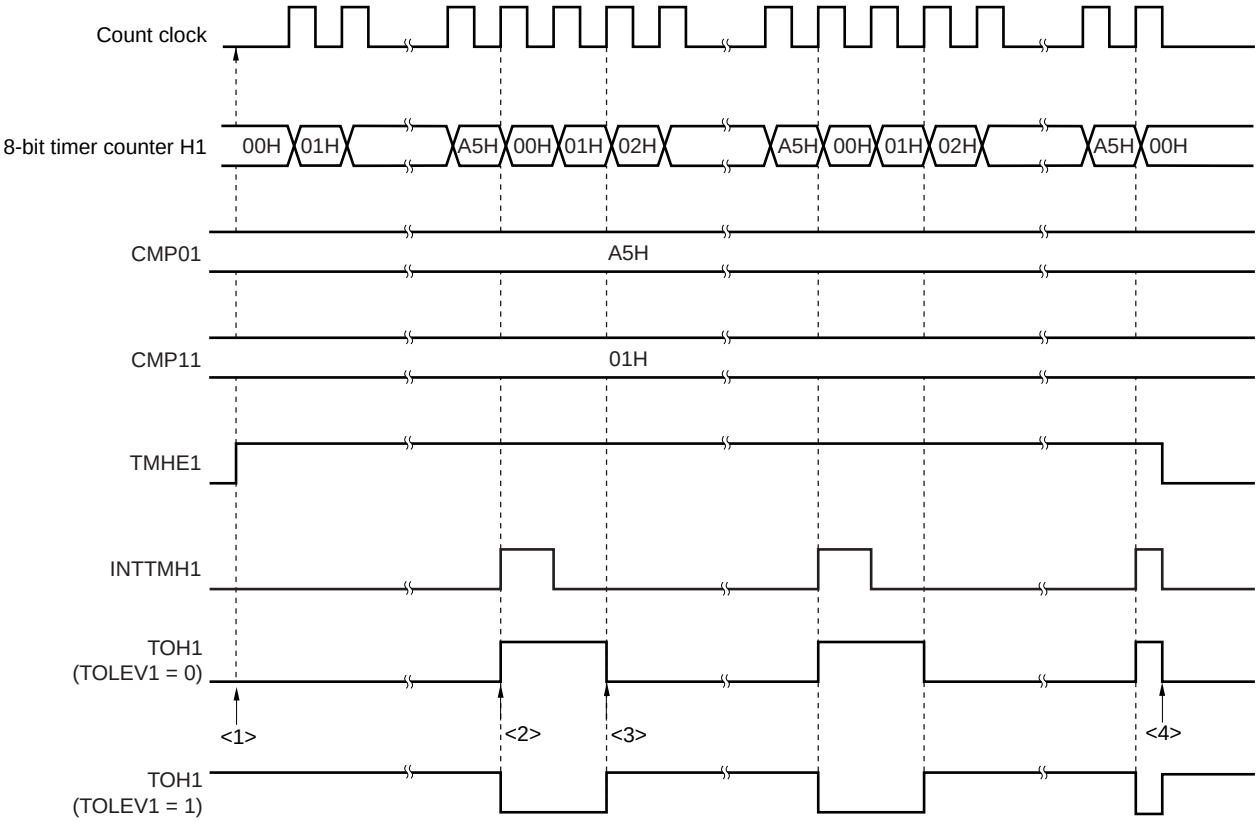
00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH

Remarks 1. For the setting of the output pin, refer to **9.3 (3) Port mode register 3 (PM3)**.

2. For details on how to enable the INTTMH1 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 9-10. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHE1 bit to 1. Start the 8-bit timer counter H1 by masking one count clock to count up. At this time, PWM output outputs an inactive level.

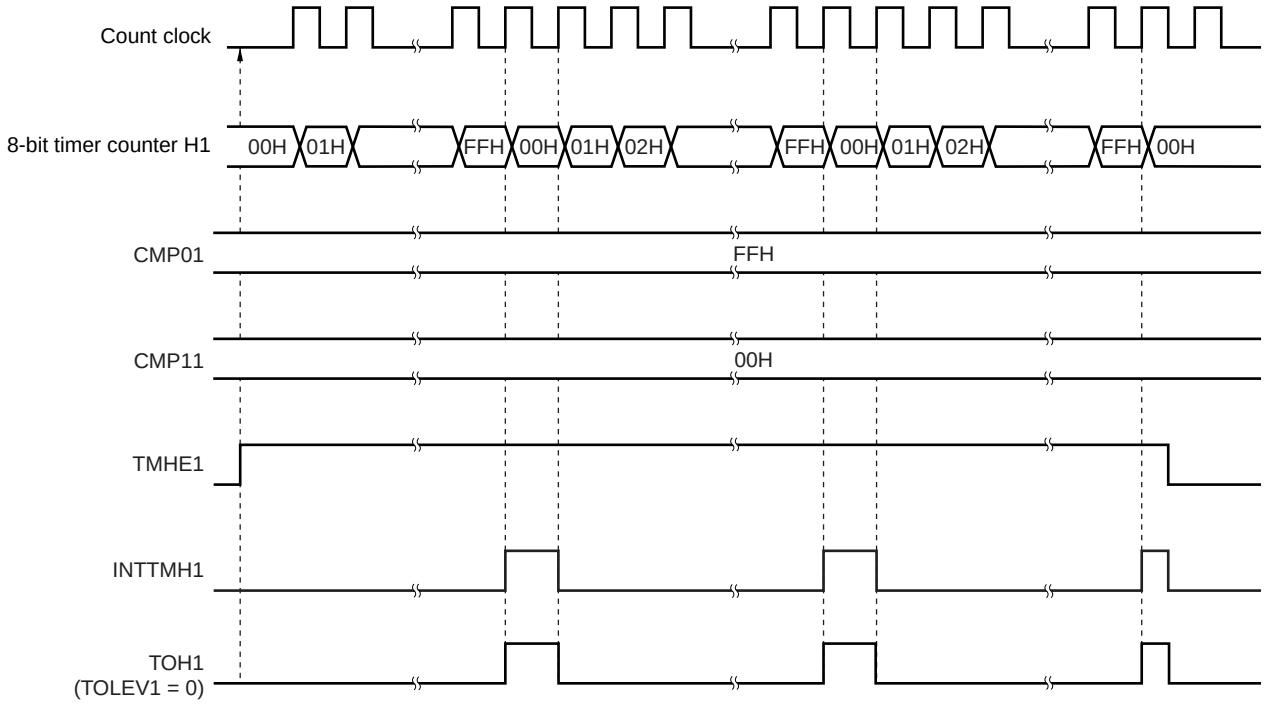
<2> When the values of the 8-bit timer counter H1 and the CMP01 register match, an active level is output. At this time, the value of the 8-bit timer counter H1 is cleared, and the INTTMH1 signal is output.

<3> When the values of the 8-bit timer counter H1 and the CMP11 register match, an inactive level is output. At this time, the 8-bit timer counter value is not cleared and the INTTMH1 signal is not output.

<4> Clearing the TMHE1 bit to 0 during timer H1 operation sets the INTTMH1 signal to the default and PWM output to an inactive level.

Figure 9-10. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP01 = FFH, CMP11 = 00H



(c) Operation when CMP01 = FFH, CMP11 = FEH

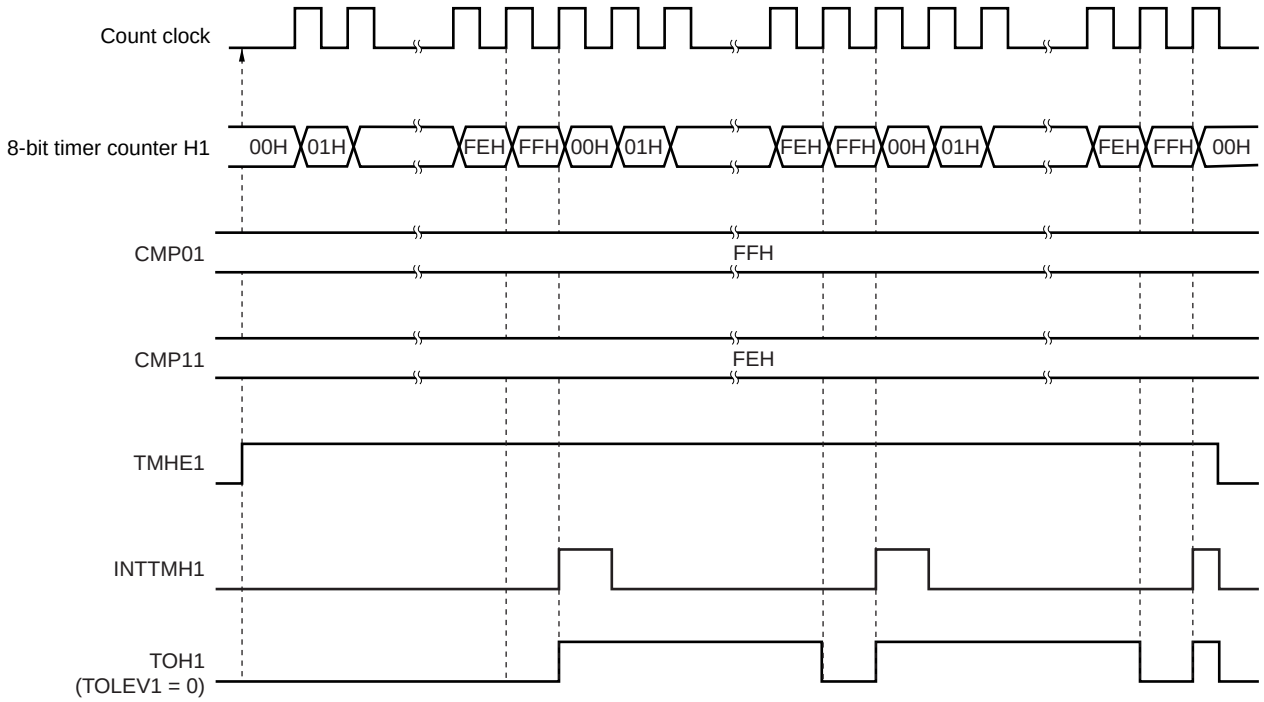


Figure 9-10. Operation Timing in PWM Output Mode (3/4)

(d) Operation when CMP01 = 01H, CMP11 = 00H

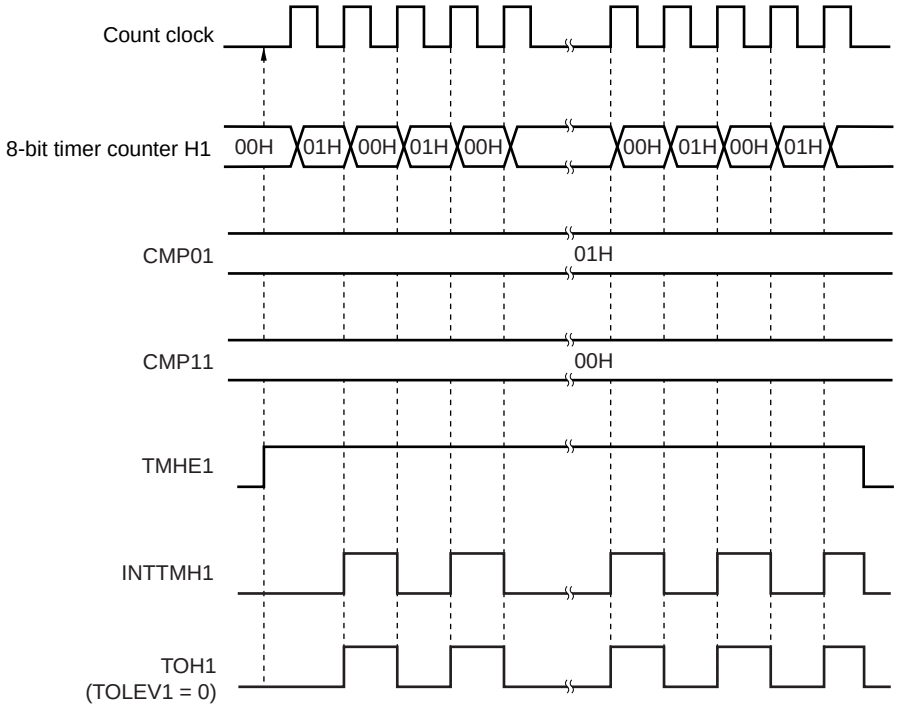
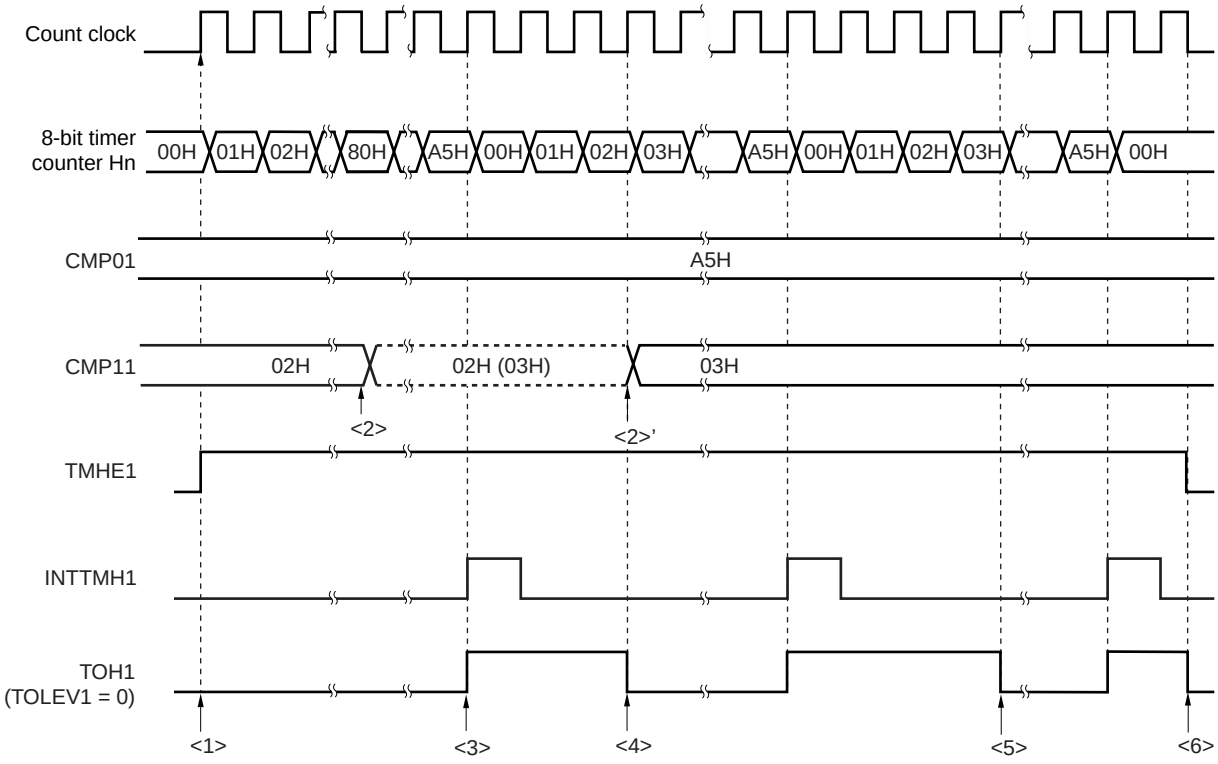


Figure 9-10. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP11 (CMP11 = 02H → 03H, CMP01 = A5H)



<1> The count operation is enabled by setting TMHE1 = 1. Start the 8-bit timer counter H1 by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> The CMP11 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 H1 and the CMP01 register match, the value of the 8-bit timer counter H1 is cleared, an active level is output, and the INTTMH1 signal is output.

<4> If the CMP11 register value is changed, the value is latched and not transferred to the register. When the values of the 8-bit timer counter H1 and the CMP11 register before the change match, the value is transferred to the CMP11 register and the CMP11 register value is changed (<2>').

However, three count clocks or more are required from when the CMP11 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 H1 and the CMP11 register after the change match, an inactive level is output. The 8-bit timer counter H1 is not cleared and the INTTMH1 signal is not generated.

<6> Clearing the TMHE1 bit to 0 during timer H1 operation sets the INTTMH1 signal to the default and PWM output to an inactive level.

9.4.3 Carrier generator operation

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.

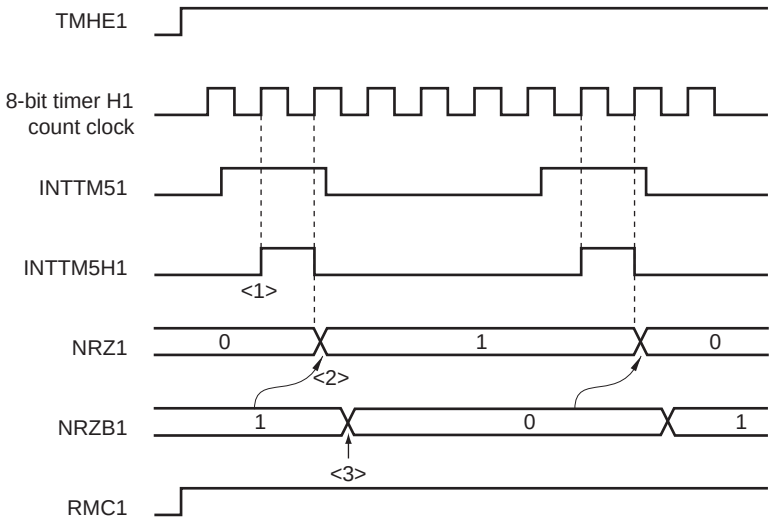
(2) Carrier output control

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 9-11. 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.

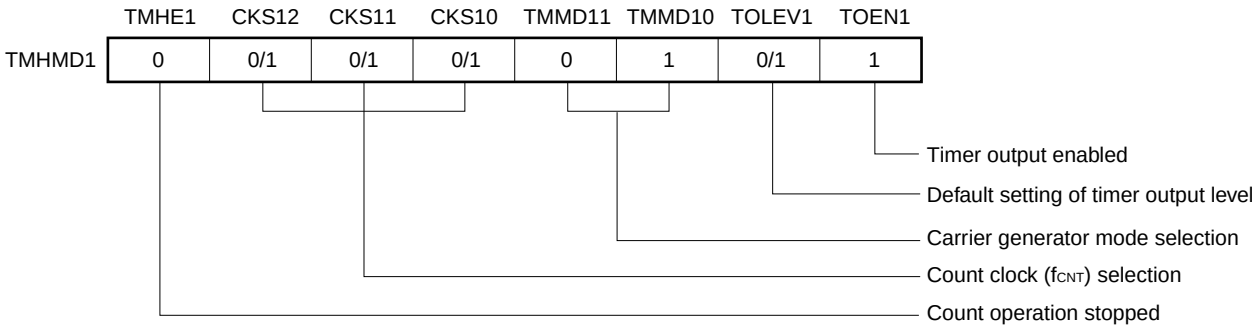
Remark INTTM5H1 is an internal signal and not an interrupt source.

Setting

<1> Set each register.

Figure 9-12. 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

- Refer to **8.3 Registers Controlling 8-bit Timer/Event Counter 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.

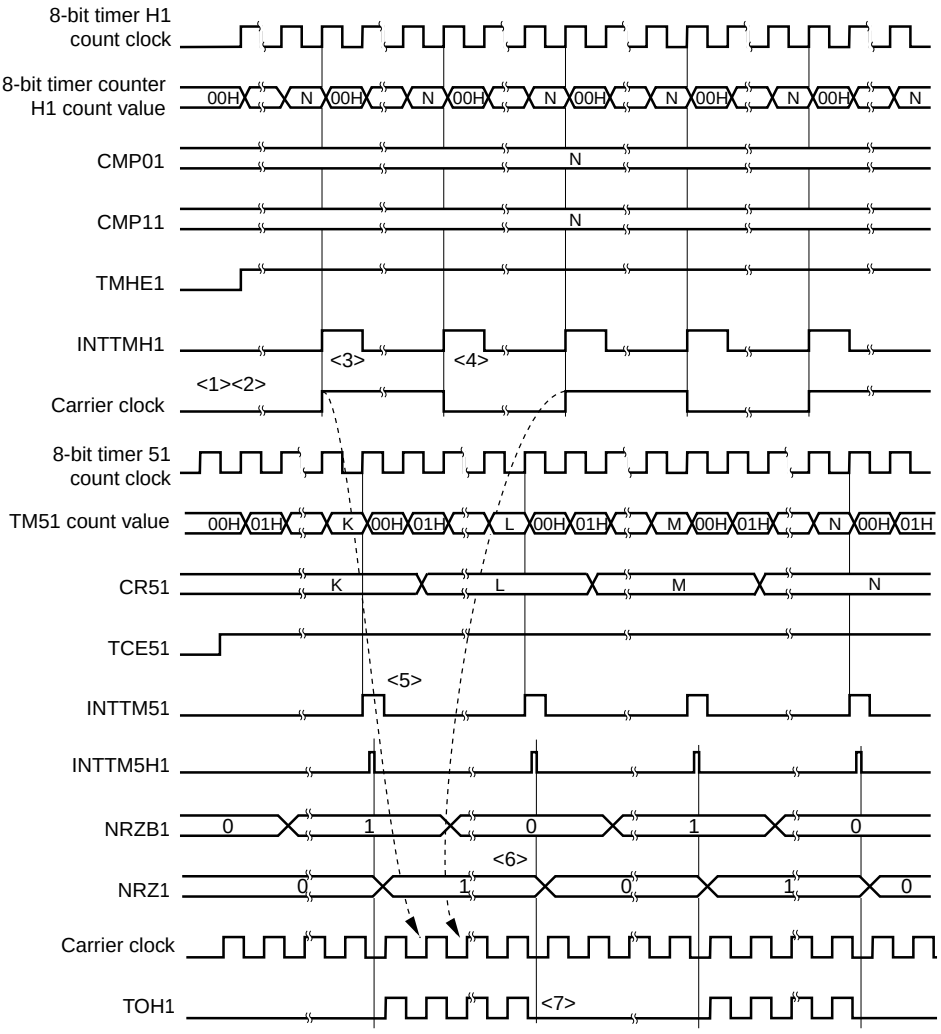
5. Be sure to set the RMC1 bit before the count operation is started.

Remarks 1. For the setting of the output pin, refer to **9.3 (3) Port mode register 3 (PM3)**.

2. For how to enable the INTTMH1 signal interrupt, refer to **CHAPTER 17 INTERRUPT FUNCTIONS**.

Figure 9-13. 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.

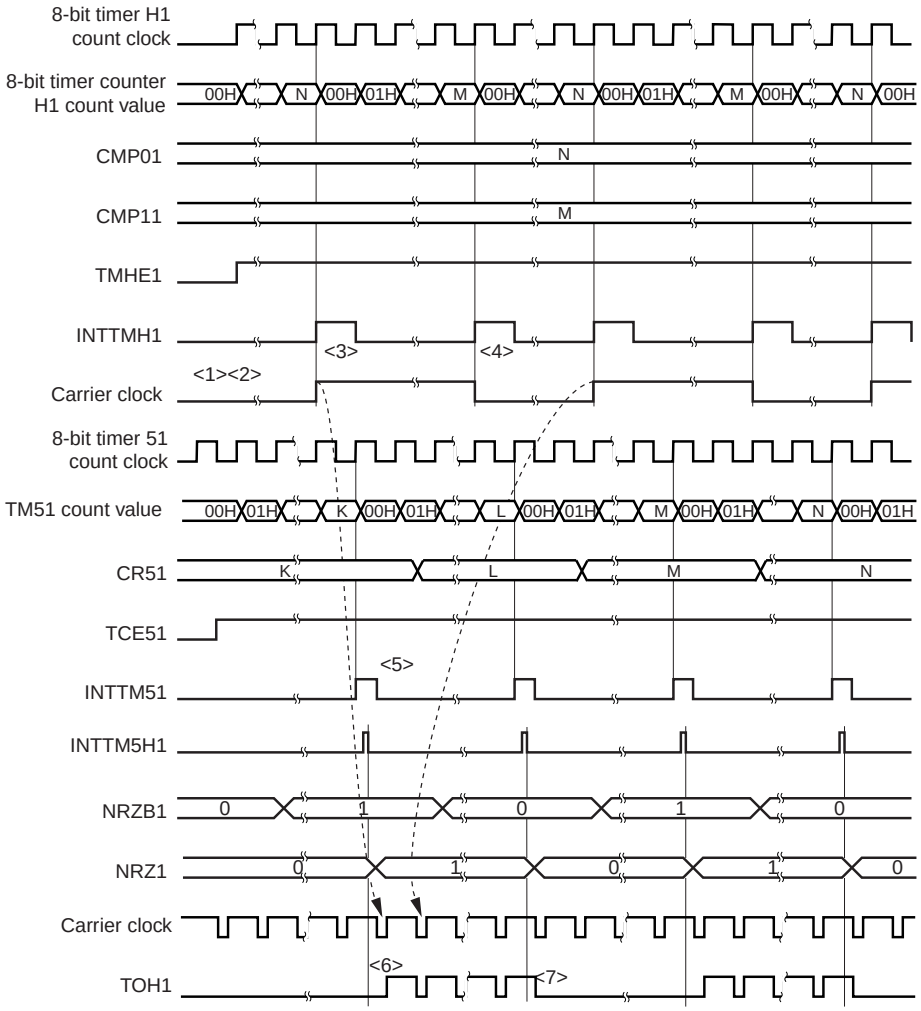
<6> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<7> When NRZ1 = 0 is set, the TOH1 output becomes low level.

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 9-13. 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.

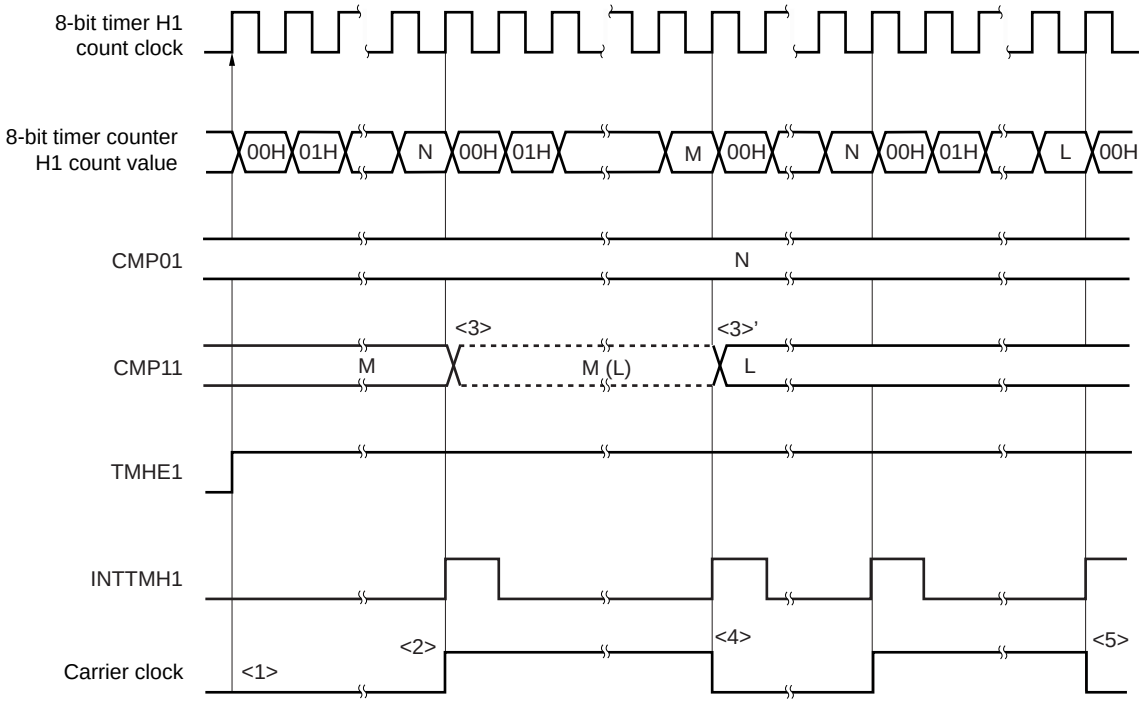
<6> A carrier signal is output at the first rising edge of the carrier clock if NRZ1 is set to 1.

<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 9-13. 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 CMP11 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).

CHAPTER 10 WATCHDOG TIMER

10.1 Functions of Watchdog Timer

The watchdog timer is mounted onto all 78K0/Fx2-L microcontroller products.

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 register (detection of an invalid check while the CPU hangs up)
- If the CPU accesses an area that is not set by the IMS register (excluding FB00H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

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, refer to **CHAPTER 19 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)

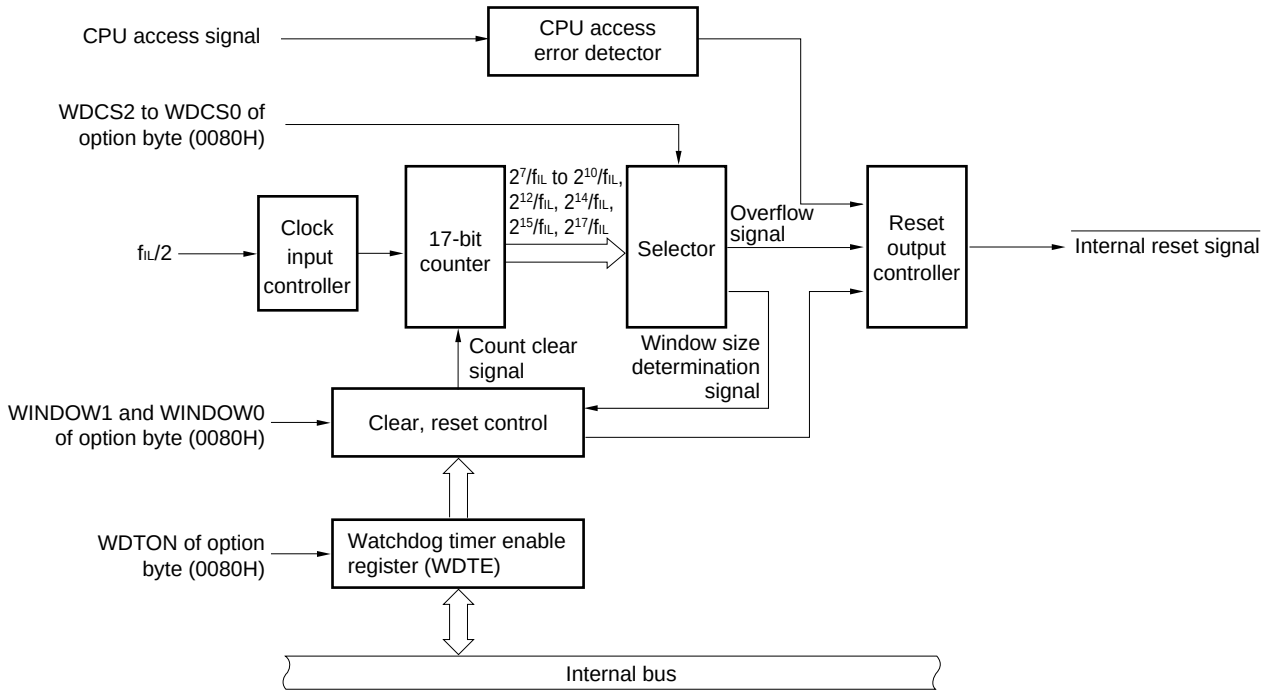
How the counter operation is controlled, overflow time, and window open period are set by the option byte.

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, refer to **CHAPTER 23 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, refer to **CHAPTER 23**).

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, refer to **10.4.2** and **CHAPTER 23**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, refer to **10.4.3** and **CHAPTER 23**).

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 register (detection of an invalid check during a CPU program loop)
   - If the CPU accesses an area not set by the IMS register (excluding FB00H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

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 $2/f_{IL}$ 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/PLL control 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	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3.97 s)

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

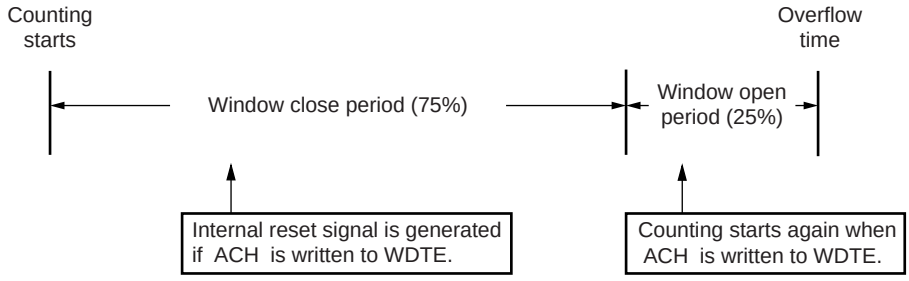
2. ( ): f_{IL} = 33 kHz (MAX.)

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 $2^{17}/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 3.64 s	0 to 2.43 s	0 to 1.21 s	None
Window open time	3.64 to 3.97 s	2.43 to 3.97 s	1.21 to 3.97 s	0 to 3.97 s

<When window open period is 25%>

- Overflow time:
  $2^{17}/f_{IL}$ (MAX.) = 2^{17}/33 kHz (MAX.) = 3.97 s
- Window close time:
  0 to $2^{17}/f_{IL}$ (MIN.) $\times$ (1 – 0.25) = 0 to 2^{17}/27 kHz (MIN.) $\times$ 0.75 = 0 to 3.64 s
- Window open time:
  $2^{17}/f_{IL}$ (MIN.) $\times$ (1 – 0.25) to $2^{17}/f_{IL}$ (MAX.) = $2^{17}/f_{IL}$ /27 kHz (MIN.) $\times$ 0.75 to 2^{17}/33 kHz (MAX.) = 3.64 to 3.97 s

CHAPTER 11 A/D CONVERTER

Item	78K0/FY2-L (16 Pins)	78K0/FA2-L (20 Pins)	78K0/FB2-L (30 Pins)
10-bit A/D converter	4 ch	6 ch	9 ch

11.1 Function of A/D Converter

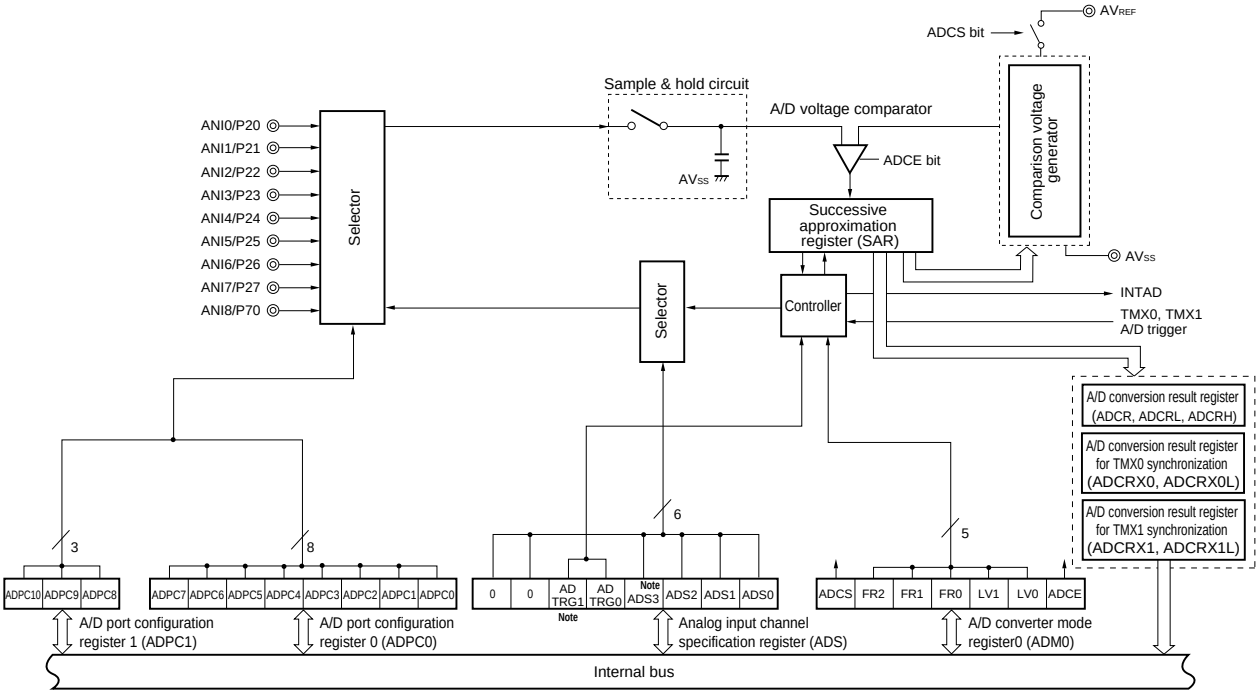
The A/D converter converts an analog input signal into a digital value, and consists of up to 9 channels (ANI0 to ANI8) with a resolution of 10 bits.

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 ANI8 . Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Remark A/D converter analog input pins differ depending on products.
- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

Figure 11-1. Block Diagram of A/D Converter



Note 78K0/FB2-L only.

Caution In the 78K0/FY2-L and 78K0/FA2-L, V_{SS} functions alternately as the ground potential of the A/D converter. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

Remark A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI8 pins

These are the analog input pins of the 9-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.

Remark A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) Comparison voltage generator

The comparison voltage generator is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared with an analog input. The operation of the comparison voltage generator is enabled or disabled by using the ADCS bit (bit 7 of the ADM0 register). The power consumption can be reduced by stopping the operation of the comparison voltage generator when A/D conversion is not performed.

(4) A/D voltage comparator

The A/D voltage comparator compares the sampled voltage values with the output voltage of the comparison voltage generator. The operation of the A/D voltage comparator is enabled or disabled by using the ADCE bit (bit 0 of the ADM0 register). The power consumption can be reduced by stopping the operation of the A/D voltage comparator when A/D conversion is not performed.

(5) Successive approximation register (SAR)

The SAR register is a 10-bit register that sets a result compared by the A/D voltage comparator, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR, ADCRH).

(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 lower 10 bits (the higher 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register L (ADCRL)

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 lower 8 bits of the A/D conversion result.

(8) 8-bit A/D conversion result register H (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.

(9) 10-bit A/D conversion result register for TMXn synchronization (ADCRXn)

If A/D conversion is started with the output of 16-bit timer Xn as the trigger, the conversion result is loaded from the successive approximation register and the A/D conversion result is held in the lower 10 bits (the higher 6 bits are fixed to 0) every time an A/D conversion ends.

(10) 8-bit A/D conversion result register L for TMXn synchronization (ADCRXnL)

If A/D conversion is started with the output of 16-bit timer Xn as the trigger, the conversion result is loaded from the successive approximation register and the lower 8 bits of the A/D conversion result are held in ADCRXnL register, every time an A/D conversion ends.

Caution When data is read from ADCR, ADCRL, ADCRH, ADCRX0, ADCRX1, ADCRX0L, and ADCRX1L, a wait cycle is generated. Do not read data from ADCR, ADCRL, ADCRH, ADCRX0, ADCRX1, ADCRX0L, and ADCRX1L when the peripheral hardware clock (f_{PRS}) is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(11) 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 all the specified A/D conversion has been completed, this controller generates an A/D conversion end interrupt request signal (INTAD).

(12) 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 ports 2 and 7 are used as a digital port.

The signal input to ANI0 to ANI8 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(13) AV_{SS} pin (78K0/FB2-L only)

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.

(14) V_{SS} pin

This is the ground potential pin. In the 78K0/FY2-L and 78K0/FA2-L, V_{SS} functions alternately as the ground potential of the A/D converter. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

11.3 Registers Used in A/D Converter

The A/D converter uses the following nine registers.

- A/D converter mode register 0 (ADM0)
- A/D port configuration registers 0, 1 (ADPC0, ADPC1)
- Analog input channel specification register (ADS)
- Port mode registers 2, 7 (PM2, PM7)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register L (ADCRL)
- 8-bit A/D conversion result register H (ADCRH)
- 10-bit A/D conversion result register for TMXn synchronization (ADCRXn)
- 8-bit A/D conversion result register L for TMXn synchronization (ADCRXnL)

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(1) A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.
ADM0 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 11-2. Format of A/D Converter Mode Register 0 (ADM0)

Address: FF28H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	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	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

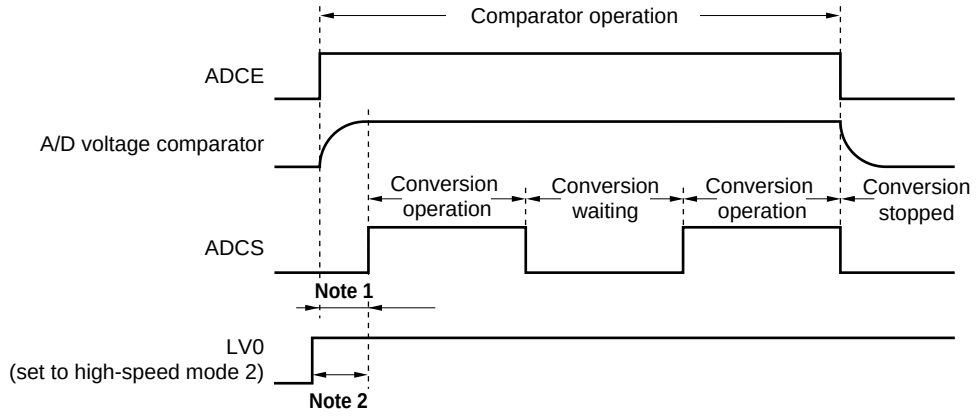
Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, refer to **Table 11-2 A/D Conversion Time Selection**.

2. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 11-1. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (only A/D voltage comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator operation)

Figure 11-3. Timing Chart When Comparator Is Used



Notes 1. To stabilize the internal circuit, the time from setting ADCE to 1 to setting ADCS to 1 must be 1 μs or longer.

2. To stabilize the internal circuit, the time from setting LV0 to 1 (high-speed mode 2) to setting ADCS to 1 must be 1 μs or longer (for operation mode setting, refer to **Table 11-2**).

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 ADM0, a wait cycle is generated. Do not write data to ADM0 when the peripheral hardware clock (f_{PRS}) is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

Table 11-2. A/D Conversion Time Selection (1/3)

(1) 4.0 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0			f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz (when using PLL)	
0	0	0	0	0	Standard	264/f_{PRS}	66.0 μs	33.0 μs	26.4 μs	13.2 μs	f_{PRS}/12
0	0	1				176/f_{PRS}	44.0 μs	22.0 μs	17.6 μs	8.8 μs	f_{PRS}/8
0	1	0				132/f_{PRS}	33.0 μs	16.5 μs	13.2 μs	6.6 μs	f_{PRS}/6
0	1	1				88/f_{PRS}	22.0 μs	11.0 μs	8.8 μs	Setting prohibited	f_{PRS}/4
1	0	0				66/f_{PRS}	16.5 μs	8.25 μs	6.6 μs		f_{PRS}/3
1	0	1				44/f_{PRS}	11.0 μs	Setting prohibited			f_{PRS}/2
1	1	0				33/f_{PRS}	8.25 μs	Setting prohibited			f_{PRS}/1.5
1	1	1				22/f_{PRS}	Setting prohibited				f_{PRS}
1	0	0	1	0	High-speed 1	66/f_{PRS}	16.5 μs	8.25 μs	6.6 μs	3.3 μs	f_{PRS}/3
1	1	0				33/f_{PRS}	8.25 μs	4.125 μs	3.3 μs	Setting prohibited	f_{PRS}/1.5
1	0	1	1	1	High-speed 2	44/f_{PRS}	11.0 μs	5.5 μs	4.4 μs	Setting prohibited	f_{PRS}/2
1	1	1				22/f_{PRS}	5.5 μs	Setting prohibited			f_{PRS}
Other than above					Setting prohibited						

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{PRS}: Peripheral hardware clock frequency

Table 11-2. A/D Conversion Time Selection (2/3)

(2) 2.7 V ≤ AV_{REF} < 4.0 V

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0			f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz (when using PLL)	
0	0	0	0	0	Standard	264/f_{PRS}	66.0 μs	33.0 μs	26.4 μs	13.2 μs	f_{PRS}/12
0	0	1				176/f_{PRS}	44.0 μs	22.0 μs	17.6 μs	Setting prohibited	f_{PRS}/8
0	1	0				132/f_{PRS}	33.0 μs	16.5 μs	13.2 μs		f_{PRS}/6
0	1	1				88/f_{PRS}	22.0 μs	Setting prohibited			f_{PRS}/4
1	0	0				66/f_{PRS}	16.5 μs	Setting prohibited			f_{PRS}/3
1	0	1				44/f_{PRS}	Setting prohibited				f_{PRS}/2
1	1	0				33/f_{PRS}	Setting prohibited				f_{PRS}/1.5
1	1	1				22/f_{PRS}	Setting prohibited				f_{PRS}
0	0	1	1	1	High-speed 2	176/f_{PRS}	44.0 μs	22.0 μs	17.6 μs	8.8 μs	f_{PRS}/8
0	1	0				132/f_{PRS}	33.0 μs	16.5 μs	13.2 μs	6.6 μs	f_{PRS}/6
0	1	1				88/f_{PRS}	22.0 μs	11.0 μs	8.8 μs	4.4 μs	f_{PRS}/4
1	0	0				66/f_{PRS}	16.5 μs	8.25 μs	6.6 μs	Setting prohibited	f_{PRS}/3
1	0	1				44/f_{PRS}	11.0 μs	5.5 μs	4.4 μs		f_{PRS}/2
1	1	0				33/f_{PRS}	8.25 μs	Setting prohibited			f_{PRS}/1.5
1	1	1				22/f_{PRS}	5.5 μs	Setting prohibited			f_{PRS}
Other than above					Setting prohibited						

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{PRS}: Peripheral hardware clock frequency

Table 11-2. A/D Conversion Time Selection (3/3)

(3) 1.8 V ≤ AV_{REF} < 2.7 V

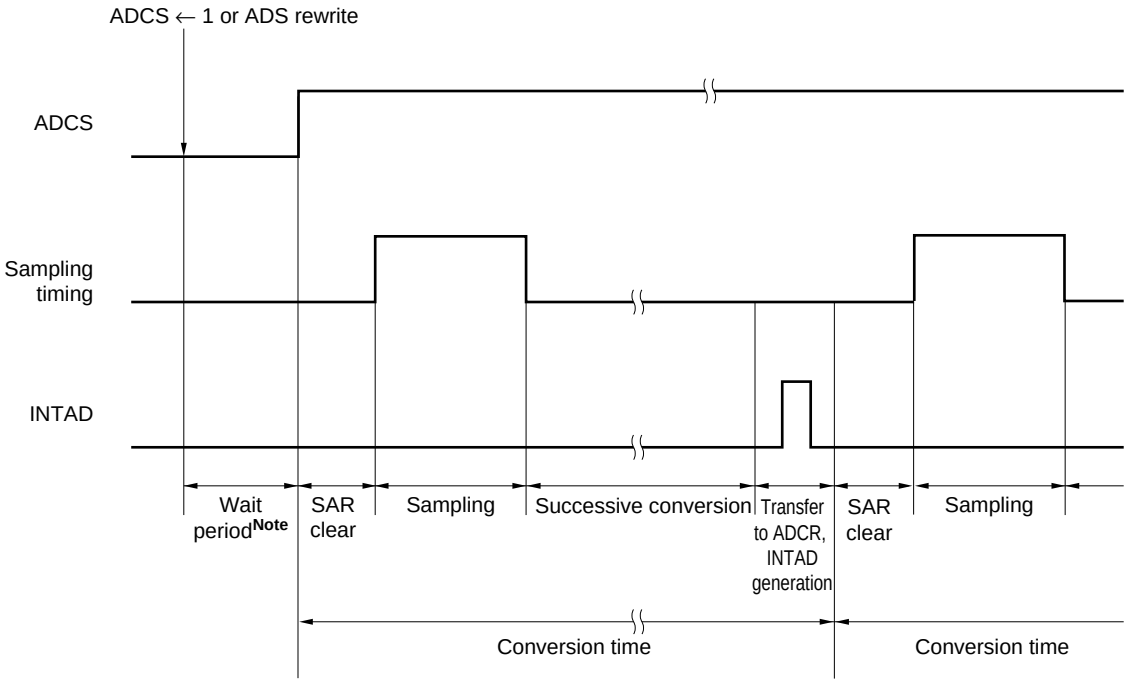
A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0			f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz (when using PLL)	
0	0	0	0	1	Low-voltage	528/f_{PRS}	Setting prohibited	66.0 μs	52.8 μs	Setting prohibited	f_{PRS}/12
0	0	1				352/f_{PRS}	Setting prohibited	44.0 μs	Setting prohibited	Setting prohibited	f_{PRS}/8
0	1	0				264/f_{PRS}	66.0 μs	Setting prohibited	Setting prohibited	Setting prohibited	f_{PRS}/6
0	1	1				176/f_{PRS}	44.0 μs	Setting prohibited	Setting prohibited	Setting prohibited	f_{PRS}/4
1	0	0				132/f_{PRS}	Setting prohibited				f_{PRS}/3
1	0	1				88/f_{PRS}	Setting prohibited				f_{PRS}/2
1	1	0				66/f_{PRS}	Setting prohibited				f_{PRS}/1.5
1	1	1				44/f_{PRS}	Setting prohibited				f_{PRS}
Other than above					Setting prohibited						

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 11-4. A/D Converter Sampling and A/D Conversion Timing



Note For details of wait period, refer to **CHAPTER 31 CAUTIONS FOR WAIT**.

(2) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The higher 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

The higher 2 bits of the conversion result are stored in FF09H and the lower 8 bits of the conversion result are stored in FF08H.

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

Reset signal generation clears this register to 0000H.

Figure 11-5. 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 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0, 1 (ADPC0, ADPC1), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. 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. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(3) 8-bit A/D conversion result register L (ADCRL)

This register is an 8-bit register that stores the A/D conversion result. The lower 8 bits of 10-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Figure 11-6. Format of 8-bit A/D Conversion Result Register L (ADCRL)

Address: FF08H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRL								

Cautions 1. When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0, 1 (ADPC0, ADPC1), the contents of ADCRL may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCRL, a wait cycle is generated. Do not read data from ADCRL when the peripheral hardware clock (f_{PRS}) is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(4) 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Figure 11-7. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FF0DH 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 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0, 1 (ADPC0, ADPC1), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. 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. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(5) 10-bit A/D conversion result register for TMXn synchronization (ADCRXn)

ADCRXn is a 16-bit register that holds the A/D conversion result when A/D conversion is started with the output of 16-bit timer Xn as the trigger. The higher 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

If A/D conversion is performed with the output of 16-bit timer X0 as the trigger, the higher 2 bits of the conversion result are stored in FF17H and the lower 8 bits in FF16H of ADCRX0.

If A/D conversion is performed with the output of 16-bit timer X1 as the trigger, the higher 2 bits of the conversion result are stored in FF19H and the lower 8 bits in FF18H of ADCRX1.

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

Reset signal generation clears ADCRXn to 0000H.

Figure 11-8. Format of 10-Bit A/D Conversion Result Register for TMXn Synchronization (ADCRXn)

Address: FF16H, FF17H (ADCRX0), FF18H, FF19H (ADCRX1) After reset: 0000H R

Symbol	FF17H (ADCRX0), FF19H (ADCRX1)								FF16H (ADCRX0), FF18H (ADCRX1)							
ADCRXn (n = 0, 1)	0	0	0	0	0	0										

Cautions 1. When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0, 1 (ADPC0, ADPC1), the contents of ADCRXn may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCRXn, a wait cycle is generated. Do not read data from ADCRXn when the peripheral hardware clock (f_{PRS}) is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L

n = 0, 1 : 78K0/FB2-L

(6) 8-bit A/D conversion result register L for TMXn synchronization (ADCRXnL)

ADCRXnL is an 8-bit register that holds the A/D conversion result when A/D conversion is started with the output of 16-bit timer Xn as the trigger.

The lower 8 bits of the 10-bit resolution are stored in ADCRX0L if A/D conversion is performed with the output of 16-bit timer X0 as the trigger or in ADCRX1L if A/D conversion is performed with the output of 16-bit timer X1 as the trigger.

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

Reset signal generation clears ADCRXnL to 00H.

Figure 11-9. Format of 8-bit A/D conversion result register L for TMXn synchronization (ADCRXnL)

Address: FF16H (ADCRX0L), FF18H (ADCRX1L) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRXnL (n = 0, 1)								

Cautions 1. When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0, 1 (ADPC0, ADPC1), the contents of ADCRXnL may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCRXnL, a wait cycle is generated. Do not read data from ADCRXnL when the peripheral hardware clock (f_{PRS}) is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

(7) Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted and sets the A/D conversion start method.

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

Reset signal generation clears this register to 00H.

Remark A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

Figure 11-10. Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FF0EH After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
ADS	0	0	ADTRG1 Note 2	ADTRG0	ADS3 Note 2	ADS2 Note 1	ADS1	ADS0

Figure 11-10. Format of Analog Input Channel Specification Register (ADS) (2/2)

ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	ANI0	P20/ANI0 pin
0	0	0	1	ANI1	P21/ANI1 pin
0	0	1	0	ANI2	P22/ANI2 pin
0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	ANI4Note 1	P24/ANI4 pinNote 1
0	1	0	1	ANI5Note 1	P25/ANI5 pinNote 1
0	1	1	0	ANI6Note 2	P26/ANI6 pinNote 2
0	1	1	1	ANI7Note 2	P27/ANI7 pinNote 2
1	0	0	0	ANI8Note 2	P70/ANI8 pinNote 2
Other than above				Setting prohibited	

ADTRG1	ADTRG0	A/D conversion start method selectionNote 3
0	0	Normal start (software trigger mode)
0	1	TMX0 synchronization (timer trigger mode set by A/D conversion trigger signal of TMX0)
1	0	TMX1 synchronizationNote 2 (timer trigger mode set by A/D conversion trigger signal of TMX1)
1	1	Setting prohibited

Notes 1. Setting permitted in 78K0/FA2-L and 78K0/FB2-L.

2. Setting permitted in 78K0/FB2-L.

3. Switching the A/D conversion start method should be done after stopping the A/D conversion operation (clearing (0) ADCS).

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 7 (PM2, PM7).

2. 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. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(8) A/D port configuration registers 0, 1 (ADPC0, ADPC1)

ADPC0 switches the P20/ANI0 to P27/ANI7 pins to digital I/O or analog input of port. Each bit of ADPC0 corresponds to a pin of port 2 and can be specified in 1-bit units.

ADPC1 switches the ANI8/P70 pin to digital I/O or analog input of port. Each bit of ADPC1 corresponds to a pin of P70 in port 1 and can be specified in 1-bit units.

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

Reset signal generation clears ADPC0 and ADPC1 to 00H.

Remark A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

Figure 11-11. Format of A/D Port Configuration Register 0 (ADPC0)

(1) 78K0/FY2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPCS3	ADPCS2	ADPCS1	ADPCS0

(2) 78K0/FA2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	ADPCS5	ADPCS4	ADPCS3	ADPCS2	ADPCS1	ADPCS0

(3) 78K0/FB2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	ADPCS7	ADPCS6	ADPCS5	ADPCS4	ADPCS3	ADPCS2	ADPCS1	ADPCS0

ADPCSn	Digital I/O or analog input selection (n = 0 to 7)
0	Analog input
1	Digital I/O

Cautions 1. Set the pin set to analog input to the input mode by using port mode register 2 (PM2).

2. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the peripheral hardware clock is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

Figure 11-12. Format of A/D Port Configuration Register 1 (ADPC1) (78K0/FB2-L Only)

Address: FF2FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC1	0	0	0	0	0	0	0	ADPC8

ADPCS8	Digital I/O or analog input selection
0	Analog input
1	Digital I/O

Cautions 1. Set the pin set to analog input to the input mode by using port mode register 7 (PM7).

2. If data is written to ADPC1, a wait cycle is generated. Do not write data to ADPC1 when the peripheral hardware clock is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(9) Port mode registers 2, 7 (PM2, PM7)

When using the ANI0/P20 to ANI7/P27 and ANI8/P70 pins for analog input port, set PM20 to PM27 and PM70 to 1. The output latches of P20 to P27 and P70 at this time may be 0 or 1.

If PM20 to PM27 and PM70 are set to 0, they cannot be used as analog input port pins.

PM2 and PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remark A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

Figure 11-13. Format of Port Mode Register 2 (PM2)

(1) 78K0/FY2-L

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

Caution Be sure to set bits 4 to 7 of PM2 to 1.

(2) 78K0/FA2-L

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20

Caution Be sure to set bits 6 and 7 of PM2 to 1.

(3) 78K0/FB2-L

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)

Figure 11-14. Format of Port Mode Register 7 (PM7) (78K0/FB2-L Only)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	1	1	1	1	1	PM70

PM70	P70 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

When using P20/ANI0 to P27/ANI7 and P70/ANI8, set the registers according to the pin function to be used (refer to **Tables 11-3** to **11-6**).

Table 11-3. Setting Functions of P2n/ANIn Pin

ADPC0 Register	PM2 Register	ADS Register	P2n/ANIn Pin
Digital I/O selection	Input mode	Selects ANIn.	Setting prohibited
		Does not select ANIn.	Digital input
	Output mode	Selects ANIn.	Setting prohibited
		Does not select ANIn.	Digital output
Analog input selection	Input mode	Selects ANIn.	Analog input (to be converted into digital signal)
		Does not select ANIn.	Analog input (not to be converted into digital signal)
	Output mode	–	Setting prohibited

Remarks 1. ADPC0: A/D port configuration register 0
PM2: Port mode register 2
ADS: Analog input channel specification register
2. n = 0 to 2, 7

Table 11-4. Setting Functions of P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins

ADPC0 Register	PM2 Register	CMPmEN bit (m = 0 to 2)	ADS Register (n = 3 to 5)	P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins
Digital I/O selection	Input mode	–	Selects ANIn.	Setting prohibited
			Does not select ANIn.	Digital input
	Output mode	–	Selects ANIn.	Setting prohibited
			Does not select ANIn.	Digital output
Analog input selection	Input mode	0	Selects ANIn.	Analog input (to be converted into digital signal)
			Does not select ANIn.	Analog input (not to be converted into digital signal)
		1	Selects ANIn.	Analog input (to be converted into digital signal), and comparator input
			Does not select ANIn.	Comparator input
	Output mode	–	–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
CMPmEN: Bit 7 of comparator m control register (CmCTL)
ADS: Analog input channel specification register

Table 11-5. Setting Functions of P26/ANI6/CMPCOM Pin

ADPC0 Register	PM2 Register	CmMODSEL1 bit (m = 0 to 2)	CmMODSEL0 bit (m = 0 to 2)	ADS Register	P26/ANI6/CMPCOM Pin
Digital I/O selection	Input mode	–		Selects ANI6.	Setting prohibited
				Does not select ANI6.	Digital input
	Output mode	–		Selects ANI6.	Setting prohibited
				Does not select ANI6.	Digital output
Analog input selection	Input mode	CmMODSEL1 = 0, or CmMODSEL0 = 0		Selects ANI6.	Analog input (to be converted into digital signal)
				Does not select ANI6.	Analog input (not to be converted into digital signal)
		CmMODSEL1 = 1, CmMODSEL0 = 1		Selects ANI6.	Analog input (to be converted into digital signal), and comparator common input
				Does not select ANI6.	Comparator common input
	Output mode	–		–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
CmMODSEL1, CmMODSEL0: Bits 4 and 3 of comparator m control register (CmCTL)
ADS: Analog input channel specification register

Table 11-6. Setting Functions of P70/ANI8 Pin

ADPC1 Register	PM7 Register	ADS Register	P70/ANI8 Pin
Digital I/O selection	Input mode	Selects ANI8.	Setting prohibited
		Does not select ANI8.	Digital input
	Output mode	Selects ANI8.	Setting prohibited
		Does not select ANI8.	Digital output
Analog input selection	Input mode	Selects ANI8.	Analog input (to be converted into digital signal)
		Does not select ANI8.	Analog input (not to be converted into digital signal)
	Output mode	–	Setting prohibited

Remark ADPC1: A/D port configuration register 1
PM7: Port mode register 7
ADS: Analog input channel specification register

11.4 A/D Converter Operations

11.4.1 Basic operations of A/D converter (software trigger mode)

<1> Set the A/D conversion time and the operation mode by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of the A/D converter mode register 0 (ADM0).
<2> Set bit 0 (ADCE) of ADM0 to 1 to start the operation of the A/D voltage comparator.
<3> Set channels for A/D conversion to analog input by using the A/D port configuration registers 0 and 1 (ADPC0, ADPC1) and set to input mode by using port mode registers 2 and 7 (PM2, PM7).
<4> Select one channel for A/D conversion by using the analog input channel specification register (ADS).
<5> Start the conversion operation by setting bit 7 (ADCS) of ADM0 to 1.
(<6> to <13> 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 comparison voltage generator outputs (1/2) AV_{REF} voltage.
<9> The voltage difference between the output voltage of the comparison voltage generator 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 output voltage of the comparison voltage generator 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 output voltage of the comparison voltage generator and sampled voltage are compared and bit 8 of SAR is manipulated as follows.
- Analog input voltage ≥ Output voltage of comparison voltage generator: Bit 8 = 1
- Analog input voltage < Output voltage of comparison voltage generator: Bit 8 = 0

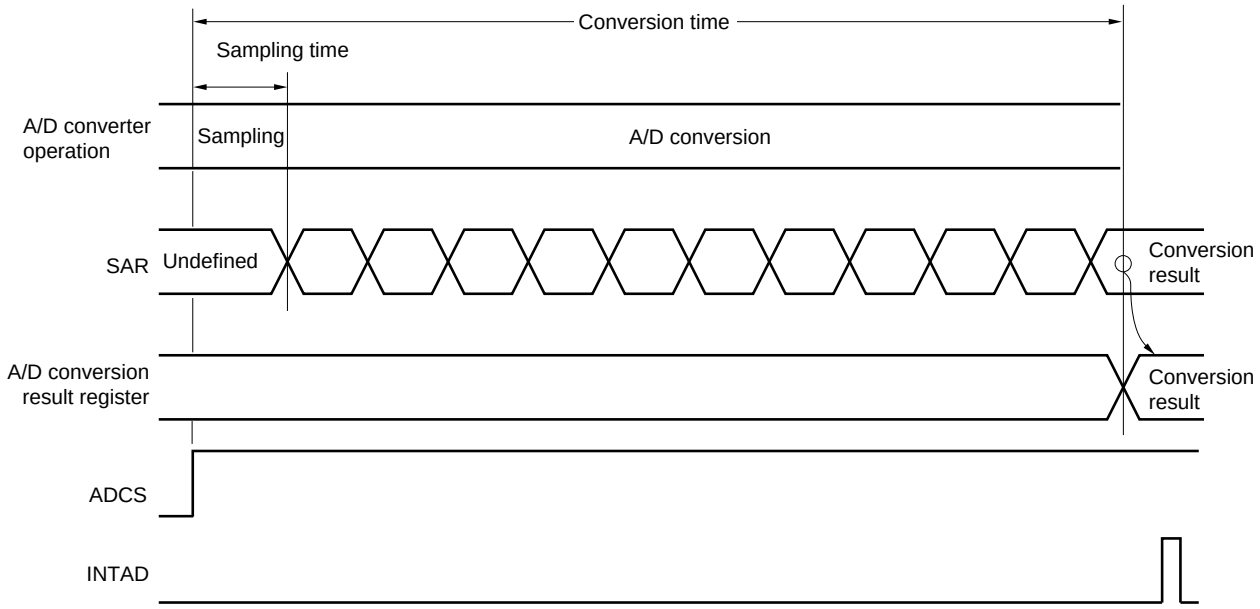
<11> Comparison is continued in this way up to bit 0 of SAR.
<12> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH, ADCRL) and then latched.
At the same time, the A/D 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>.

Cautions 1. Make sure the period of <2> to <5> is 1 μs or more.
2. If the timing of <2> is earlier than that of <4>, <2> may be performed any time.
3. When switching from software trigger mode to timer trigger mode, switch the operation mode and input channel after stopping the A/D conversion operation (clearing (0) ADCS).

Remark Three types of A/D conversion result registers are available.

- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store higher 8-bit of A/D conversion value
- ADCRL (8 bits): Store lower 8-bit of A/D conversion value

Figure 11-15. Basic Operation of A/D Converter (Software Trigger Mode)



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) 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, ADCRL) to 0000H or 00H.

11.4.2 Basic operation of A/D converter (timer trigger mode)

<1> Set the A/D conversion time and the operation mode by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of the A/D converter mode register 0 (ADM0).

<2> Set bit 0 (ADCE) of ADM0 to 1 to start the operation of the A/D voltage comparator.

<3> Set channels for A/D conversion to analog input by using the A/D port configuration registers n (ADPCn) and set to input mode by using port mode registers 2 and 7 (PM2, PM7).

<4> Select TMXn synchronization by using bits 4 and 5 (ADTRGn) of the analog input channel specification register (ADS).

<5> Select one channel for A/D conversion by using the analog input channel specification register (ADS).

<6> Set the timer trigger wait state by setting (1) bit 7 (ADCS) of ADM0.

(<7> to <15> are operations performed by hardware.)

<7> A conversion operation is started when a trigger signal (TMXn output) is detected.

<8> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<9> 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.

<10> Bit 9 of the successive approximation register (SAR) is set. The comparison voltage generator outputs (1/2) AV_{REF} voltage.

<11> The voltage difference between the output voltage of the comparison voltage generator 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.

<12> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The output voltage of the comparison voltage generator 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 output voltage of the comparison voltage generator and sampled voltage are compared and bit 8 of SAR is manipulated as follows.

- Analog input voltage ≥ Output voltage of comparison voltage generator: Bit 8 = 1
- Analog input voltage < Output voltage of comparison voltage generator: Bit 8 = 0

<13> Comparison is continued in this way up to bit 0 of SAR.

<14> 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 (TMXn synchronization: ADCRXn, ADCRXnL) and then latched.

At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<15> Repeat steps <8> to <14>, until ADCS is cleared to 0.

To stop the A/D converter, clear ADCS to 0.

To restart A/D conversion from the status of ADCE = 1, start from <6>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <6>. To change a channel of A/D conversion, start from <5>.

Cautions 1. Make sure the period of <2> to <6> is 1 μs or more.

2. If the timing of <2> is earlier than that of <5>, <2> may be performed any time.

3. When switching from timer trigger mode to software trigger mode, switch the operation mode and input channel after stopping the A/D conversion operation (clearing (0) ADCS).

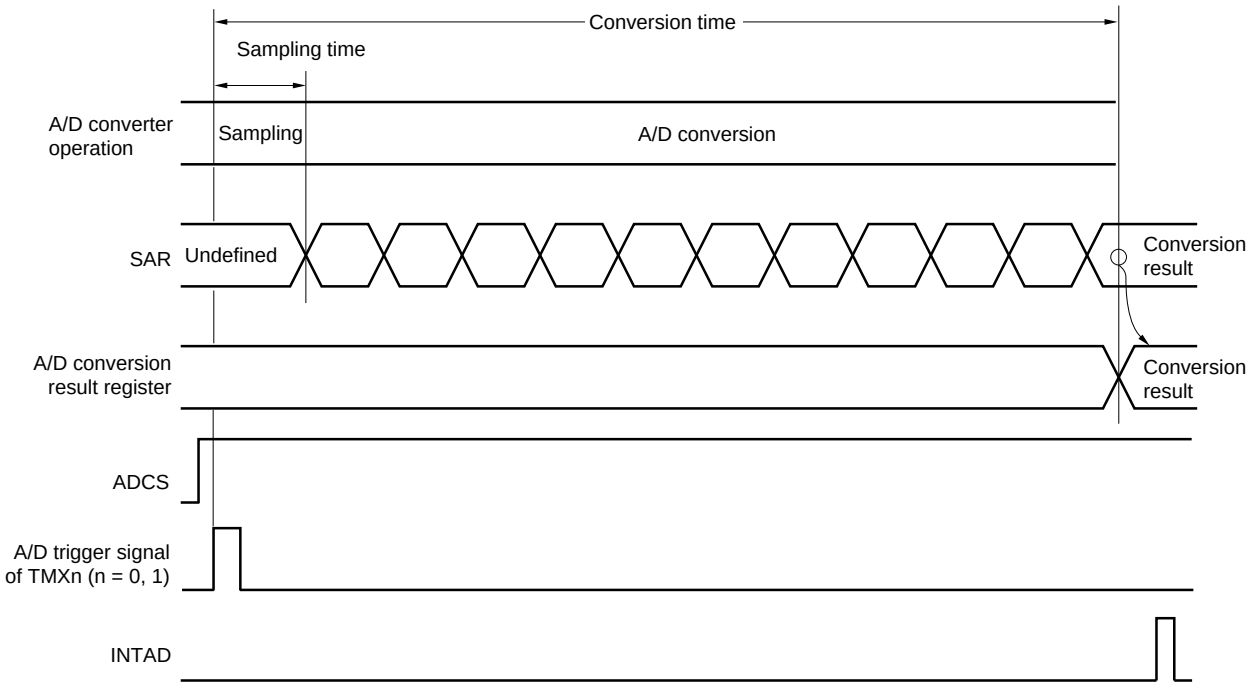
Remarks1. Two types of A/D conversion result registers are available.

- ADCRXn (16 bits) : Store 10-bit A/D conversion value
- ADCRXnL (8 bits) : Store lower 8-bit of A/D conversion value

2. n = 0 : 78K0/FY2-L, 78K0/FA2-L

n = 0, 1 : 78K0/FB2-L

Figure 11-16. Basic Operation of A/D Converter (Timer Trigger Mode)



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed for the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation will be initialized. If the ADCS bit is set (1), conversion is started from the beginning after the A/D trigger signal of TMXn is detected.

Reset signal generation clears the A/D conversion result register (ADCRXn, ADCRXnL) to 0000H or 00H.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L

n = 0, 1 : 78K0/FB2-L

11.4.3 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI8) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$ADCR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

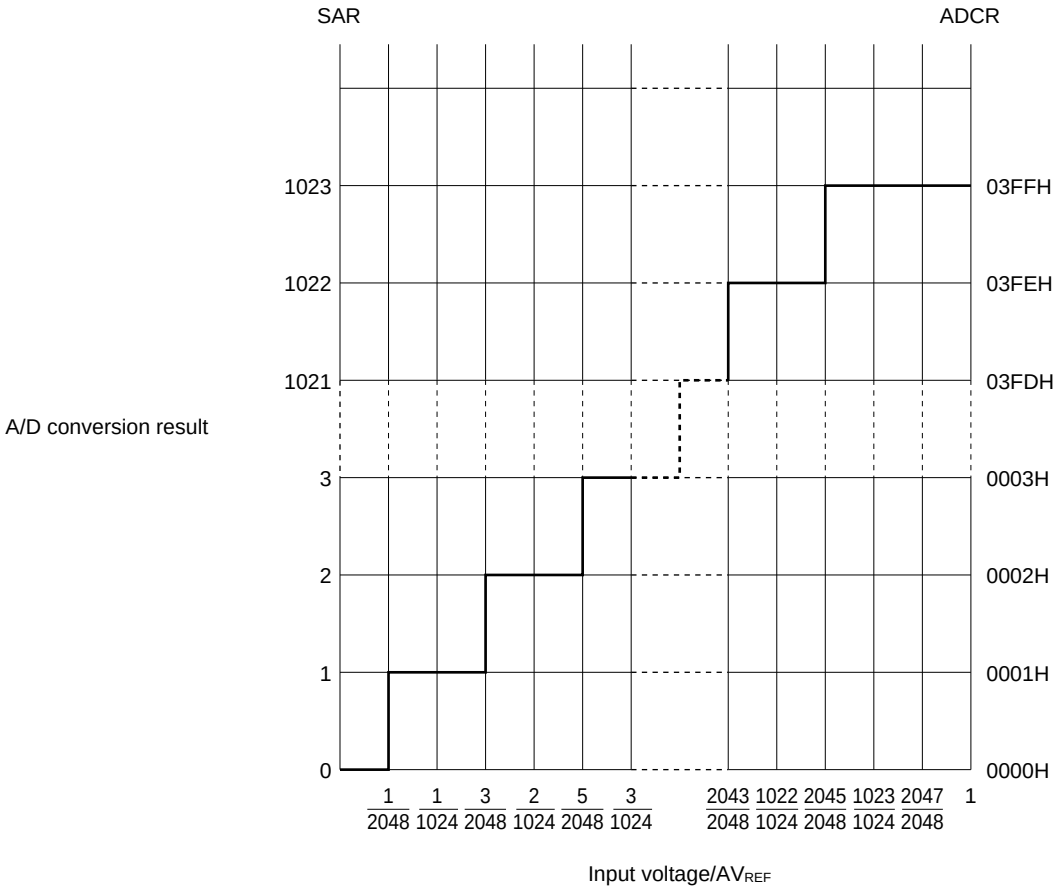
ADCR: 10-bit A/D conversion result register (ADCR) value

Remark A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

Figure 11-14 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-17. Relationship between Analog Input Voltage and A/D Conversion Result



11.4.4 A/D converter trigger mode selection

Two trigger modes for setting the A/D conversion start timing are available. These trigger modes are set by using the analog input channel specification register (ADS).

- Software trigger mode
- Timer trigger mode

(1) Software trigger mode

If a normal start is set by setting the ADTRGn bit, A/D conversion of the analog input channel selected by ADS will be started by setting ADCS = 1.

After the A/D conversion ends, A/D conversion is successively repeated, unless ADCS = 0 is set.

(2) Timer trigger mode

If TMXn synchronization is set by setting the ADTRGn bit, A/D conversion of the analog input channel selected by the analog input channel specification register (ADS) will be started when the A/D trigger signal of TMXn is detected after ADCS = 1 is set.

After the A/D conversion ends, A/D conversion is successively repeated after the A/D trigger signal of TMXn is detected, unless ADCS = 0 is set.

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

Caution Switching the trigger mode should be done after stopping the A/D conversion operation (clearing (0) ADCS).

11.4.5 A/D converter operation mode

One channel of analog input is selected by the analog input channel specification register (ADS) and A/D conversion is executed.

(1) A/D conversion operation

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 executed.

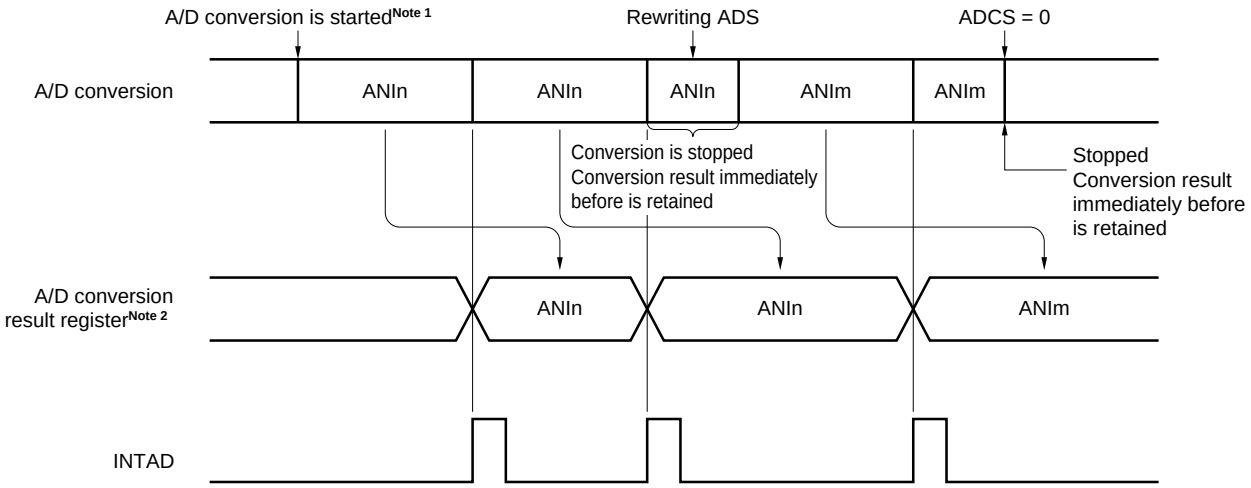
When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register, and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

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.

Remark A/D converter analog input pins differ depending on products.
- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

Figure 11-18. A/D Conversion Operation



Notes 1. Software trigger mode: A/D conversion is started by setting (1) ADCS.
Timer trigger mode: A/D conversion is started when a timer trigger signal (TMX0 or TMX1 output (78K0/FB2-L only)) is detected after ADCS is set (1).

2. Software trigger mode: ADCR, ADCRH, ADCRL registers
Timer trigger mode: ADCRX0, ADCRX0L registers (TMX0 synchronization)
ADCRX1, ADCRX1L registers (TMX1 synchronization) (78K0/FB2-L only)

Remarks 1. n = 0 to 8 (it depends on products)

2. m = 0 to 8 (it depends on products)

The setting methods are described below.

(2) Setting of Software trigger mode

<1> Set the A/D conversion time and the operation mode by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of the A/D converter mode register 0 (ADM0).

<2> Set bit 0 (ADCE) of ADM0 to 1.

<3> Set the channel to be used to analog input by using the A/D port configuration registers 0 and 1 (ADPC0, ADPC1) and port mode registers 2 and 7 (PM2, PM7).

<4> Select a channel to be used by using the analog input channel specification register (ADS).

<5> Set bit 7 (ADCS) of ADM0 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, ADCRL).

<Change the channel>

<8> Set bit 0 (ADMK) of the interrupt mask flag register 1L (MK1L) to 1**Note**.

<9> Change the channel by using ADS to start A/D conversion.

<10> Clear bit 0 (ADIF) of the interrupt request flag register 1L (IF1L) to 0.

<11> Clear ADMK to 0**Note**.

<12> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

<13> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH, ADCRL).

<Complete A/D conversion>

<14> Clear ADCS to 0.

<15> Clear ADCE to 0.

Note Execute this only if interrupt servicing is used for A/D conversion.

Cautions 1. Make sure the period of <2> to <5> is 1 μs or more.

2. If the timing of <2> is earlier than that of <4>, <2> may be performed any time.

3. <2> can be omitted. However, ignore data of the first conversion after <5> in this case.

4. The period from <6> to <12> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM0. The period from <9> to <12> is the conversion time set using FR2 to FR0, LV1, and LV0.

5. When switching from software trigger mode to timer trigger mode, switch the operation mode and input channel after stopping the A/D conversion operation (clearing (0) ADCS).

(3) Setting of Timer trigger mode

<1> Set the A/D conversion time and the operation mode by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of the A/D converter mode register 0 (ADM0).

<2> Set bit 0 (ADCE) of ADM0 to 1.

<3> Set the channel to be used to analog input by using the A/D port configuration registers 0 and 1 (ADPCn) and port mode registers 2 and 7 (PM2, PM7).

<4> Select TMXn synchronization by using bits 4 and 5 (ADTRGn) of the analog input channel specification register (ADS).

<5> Select a channel to be used by using the analog input channel specification register (ADS).

<6> Set the timer trigger wait state by setting (1) bit 7 (ADCS) of ADM0.

<7> A conversion operation is started when a trigger signal (TMXn output) is detected.

<8> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

<9> Transfer the A/D conversion data to the A/D conversion result register (ADCRXn, ADCRXnL).

<Change the channel>

<10> Set bit 0 (ADMK) of the interrupt mask flag register 1L (MK1L) to 1[Note].

<11> Change the channel by using ADS to start A/D conversion.

<12> Clear bit 0 (ADIF) of the interrupt request flag register 1L (IF1L) to 0.

<13> Clear ADMK to 0[Note].

<14> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

<15> Transfer the A/D conversion data to the A/D conversion result register (ADCRXn, ADCRHXnL).

<Complete A/D conversion>

<16> Clear ADCS to 0.

<17> Clear ADCE to 0.

Note Execute this only if interrupt servicing is used for A/D conversion.

Cautions 1. Make sure the period of <2> to <6> is 1 μs or more.

2. If the timing of <2> is earlier than that of <5>, <2> may be performed any time.

3. <2> can be omitted. However, ignore data of the first conversion after <7> in this case.

4. The period from <8> to <14> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM0. The period from <11> to <14> is the conversion time set using FR2 to FR0, LV1, and LV0.

5. When switching from timer trigger mode to software trigger mode, switch the operation mode and input channel after stopping the A/D conversion operation (clearing (0) ADCS).

Remark n = 0 : 78K0/FY2-L, 78K0/FA2-L

n = 0, 1 : 78K0/FB2-L

11.5 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 11-19. Overall Error

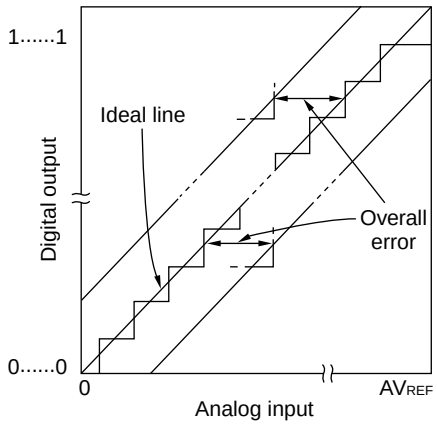
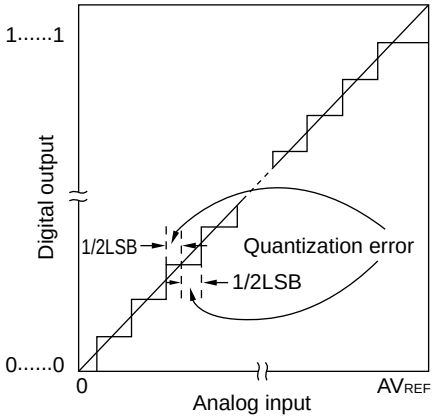


Figure 11-20. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 11-21. Zero-Scale Error

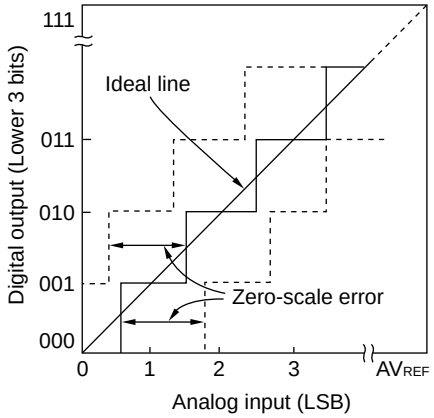


Figure 11-22. Full-Scale Error

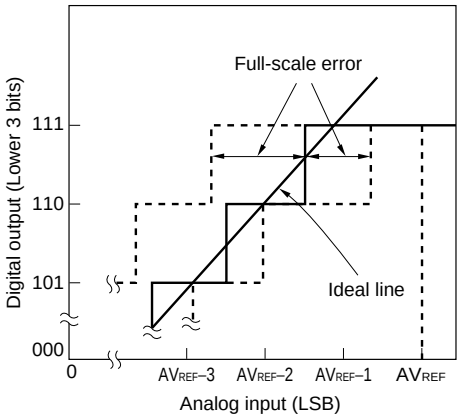


Figure 11-23. Integral Linearity Error

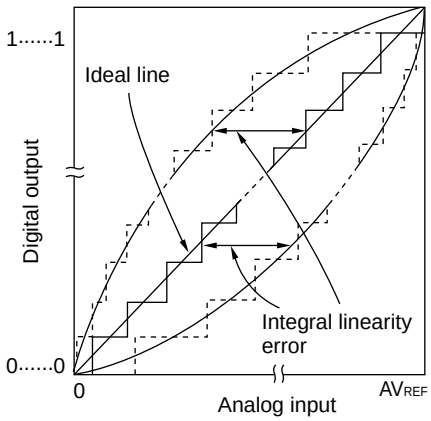
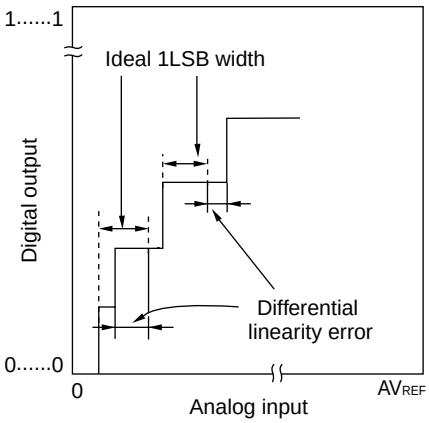


Figure 11-24. Differential Linearity Error



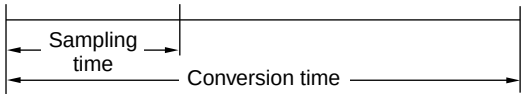
(8) Conversion time

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.



11.6 Cautions for A/D Converter

(1) Operating current in STOP mode

To satisfy the DC characteristics of the power supply current in STOP mode, clear bits 7 (ADCS) and 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 before executing a STOP instruction.

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 ANI8

Observe the rated range of the ANI0 to ANI8 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 write and A/D conversion result register read by instruction upon the end of conversion

A/D conversion result register read has priority. After the read operation, the new conversion result is written to A/D conversion result register.

<2> Conflict between A/D conversion result register write and A/D converter mode register 0 (ADM0) write, analog input channel specification register (ADS), or A/D port configuration registers 0, 1 (ADPC0, ADPC1) write upon the end of conversion

ADM0, ADS, ADPC0, or ADPC1 write has priority. A/D conversion result register 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 ANI8.

<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 11-22 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.

Remarks 1. A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

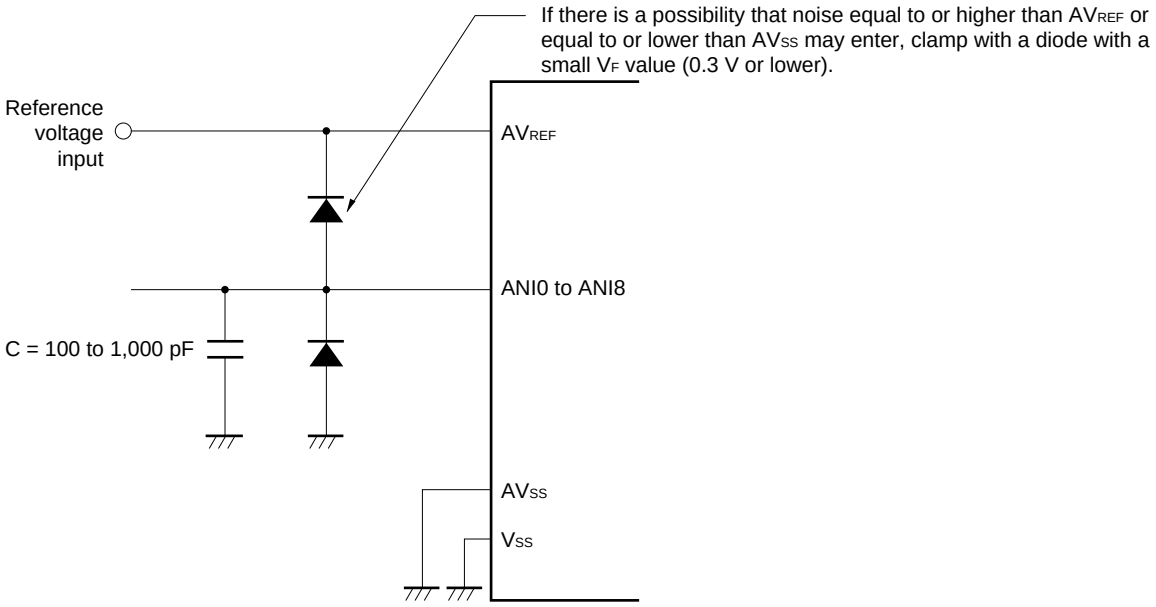
2. A/D conversion result registers differ depending on the trigger mode.

- Software trigger mode: ADCR, ADCRH, ADCRL registers
- Timer trigger mode: ADCRXn, ADCRXnL registers (TMXn synchronization)

3. n = 0 : 78K0/FY2-L, 78K0/FA2-L

n = 0, 1 : 78K0/FB2-L

Figure 11-25. Analog Input Pin Connection



(5) ANI0/P20 to ANI7/P27 and ANI8/P70

<1> The analog input pins (ANI0 to ANI7 and ANI8) are also used as digital I/O port pins (P20 to P27 and P70). When A/D conversion is performed with any of ANI0 to ANI7 and ANI8 selected, do not access P20 to P27 and P70 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> To use the ANI0/P20 to ANI7/P27 and ANI8/P70 pins for digital I/O port, it is recommended to the furthest ANI4/P24 pin from AV_{REF}. To use these pins as analog input, it is recommended to select a pin to use starting with the closest pin to AV_{SS}.

<3> 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 ANI8 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flow 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 ANI8 pins (refer to **Figure 11-22**).

Remark A/D converter analog input pins differ depending on products.

- 78K0/FY2-L: ANI0 to ANI3
- 78K0/FA2-L: ANI0 to ANI5
- 78K0/FB2-L: ANI0 to ANI8

(7) AV_{REF} pin input impedance

A series resistor string of several tens of kΩ is connected between the AV_{REF} and AV_{SS} pins.

Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV_{REF} and AV_{SS} pins, resulting in a large reference voltage error.

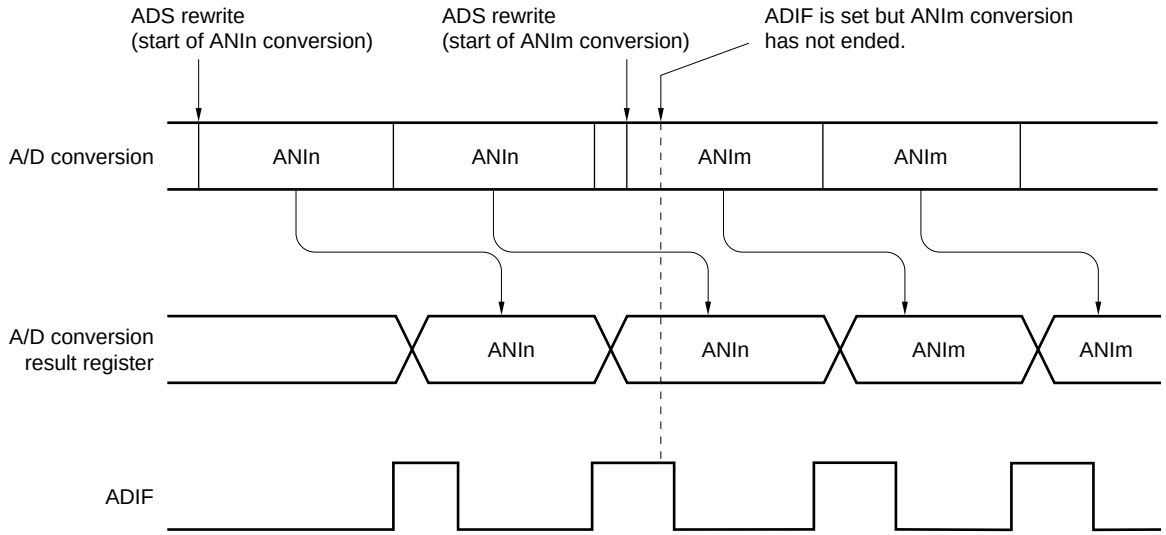
(8) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 11-26. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 8 (it depends on products)

2. m = 0 to 8 (it depends on products)

(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 read operation

When a write operation is performed to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0, 1 (ADPC0, ADPC1), the contents of A/D conversion result register may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. 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 11-27. Internal Equivalent Circuit of ANIn Pin

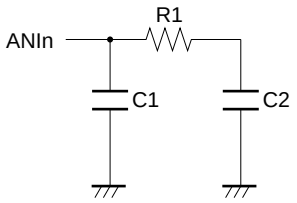


Table 11-7. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	Mode	R1	C1	C2
4.0 V ≤ AV_{REF} ≤ 5.5 V	Normal	5.2 kΩ	8 pF	6.3 pF
	High-speed 1	5.2 kΩ		
	High-speed 2	7.8 kΩ		
2.7 V ≤ AV_{REF} < 4.0 V	Normal	18.6 kΩ		
	High-speed 2	7.8 kΩ		
1.8 V ≤ AV_{REF} < 4.0 V	Low voltage	169.8 kΩ		

Remarks 1. The resistance and capacitance values shown in Table 11-7 are not guaranteed values.

2. n = 0 to 8 (it depends on products)

3. A/D conversion result registers differ depending on the trigger mode.

- Software trigger mode: ADCR, ADCRH, ADCRL registers
- Timer trigger mode: ADCRXm, ADCRXmL registers (TMXm synchronization)

4. m = 0 : 78K0/FY2-L, 78K0/FA2-L

m = 0, 1 : 78K0/FB2-L

CHAPTER 12 COMPARATORS

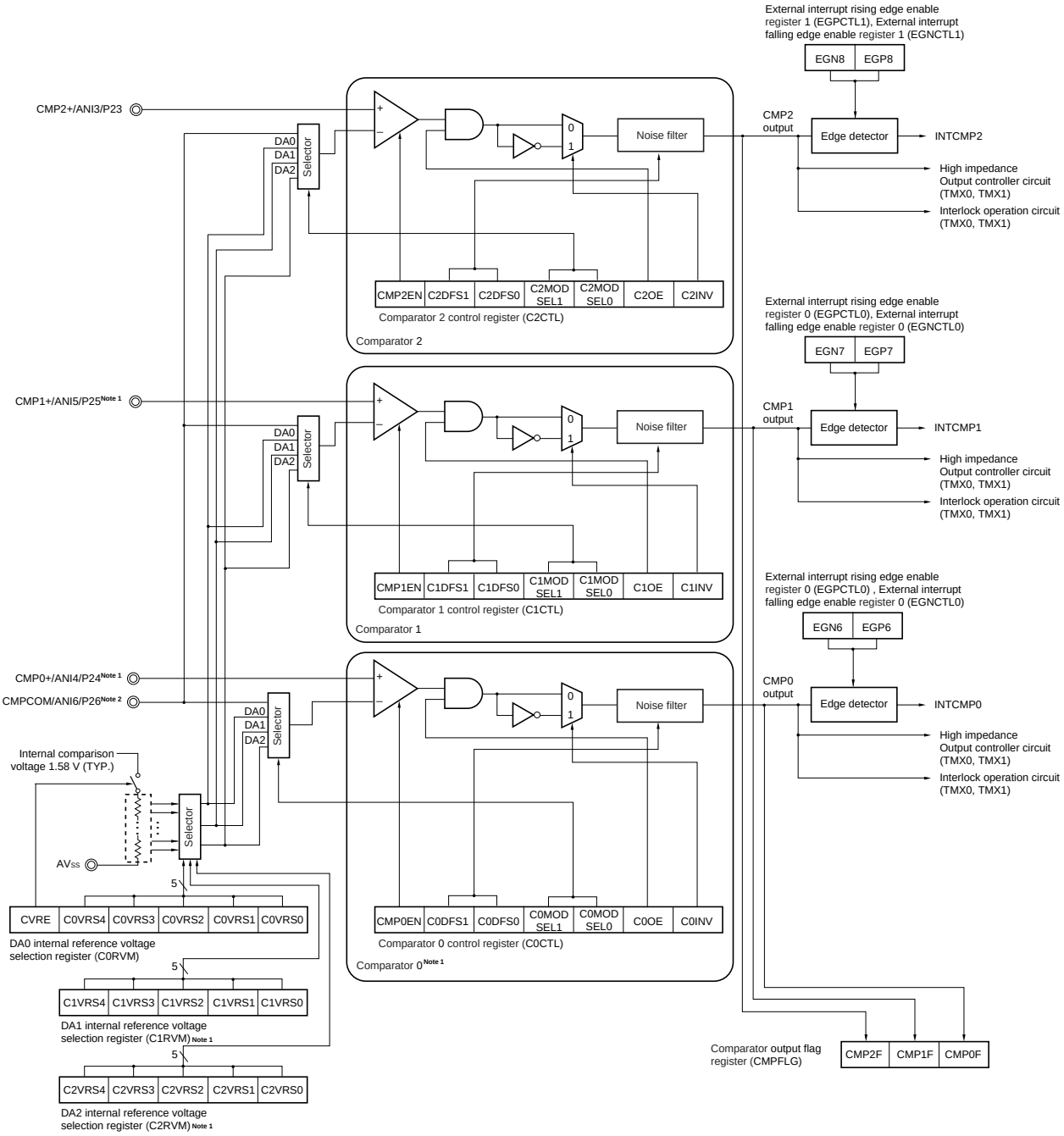
	78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
Comparator 0	Not mounted	Mounted	
Comparator 1	Not mounted	Mounted	
Comparator 2	Mounted		

12.1 Features of Comparator

Comparator has the following functions.

- A comparator is equipped with three channels (comparator 0 to 2).
- The following reference voltages can be selected.
  <1> Internal reference voltage: 3 (reference voltage level: 1.58 V (TYP.) divided by 32)
  <2> Input voltage from comparator common pin (CMPCOM) (78K0/FB2-L only)
- An interrupt signal can be generated by detecting the valid edge of the comparator output. The valid edge can be set by using the EGPn and EGNn bits (n = 6 to 8) (refer to **CHAPTER 17 INTERRUPT FUNCTIONS**).
- The comparator output can be used as the PWM output of 16-bit timers X0 and X1, a timer counter reset, and a capture trigger (refer to **CHAPTER 6 16-BIT TIMERS X0 AND X1**).
- The elimination width of the noise elimination digital filter can be selected.

Figure 12-1. Block Diagram of Comparator



Notes 1. 78K0/FA2-L and 78K0/FB2-L only

2. 78K0/FB2-L only

12.2 Configurations of Comparator

The comparators consist of the following hardware.

Table 12-1. Configurations of Comparator

Item	Configuration
Control registers	Comparator n control registers (CnCTL)
	DAn internal reference voltage selection registers (CnRVM)
	Comparator output flag register (CMPFLG)
	A/D configuration register 0 (ADPC0)
	External interrupt rising edge enable registers (EGPCTL0, EGPCTL1)
	External interrupt falling edge enable registers (EGNCTL0, EGNCTL1)
	Port mode register 2 (PM2)

12.3 Registers Controlling Comparators

The comparators use the following seven registers.

- Comparator n control registers (CnCTL)
- DAn internal reference voltage selection registers (CnRVM)
- Comparator output flag register (CMPFLG)
- A/D configuration register 0 (ADPC0)
- External interrupt rising edge enable registers (EGPCTL0, EGPCTL1)
- External interrupt falling edge enable registers (EGNCTL0, EGNCTL1)
- Port mode register 2 (PM2)

(1) Comparator n control registers (CnCTL)

This register is used to control the operation of comparator n, enable or disable comparator output, reverse the output, and set the noise elimination width.

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

Reset signal generation clears this register to 00H.

Remark 78K0/FY2-L: n = 2

78K0/FA2-L, 78K0/FB2-L: n = 0 to 2

Figure 12-2. Format of Comparator 0 Control Register (C0CTL) (78K0/FA2-L, 78K0/FB2-L)

Address: FF62H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
C0CTL	CMP0EN	C0DFS1	C0DFS0	C0MODSEL1	C0MODSEL0	0	C0OE	C0INV

CMP0EN	Comparator 0 operation control
0	Stops operation
1	Enables operation Enables input to the external pins (CMP0+) on the positive and negative sides of comparator 0

C0DFS1	C0DFS0	Noise elimination width setting
0	0	Noise filter unused
0	1	$2/f_{PRS}$
1	0	$2^2/f_{PRS}$
1	1	$2^4/f_{PRS}$

C0MODSEL1	C0MODSEL0	Reference voltage selection
0	0	Internal reference voltage: DA0
0	1	Internal reference voltage: DA1
1	0	Internal reference voltage: DA2
1	1	Internal reference voltage: CMPCOM**Note**

C0OE	Enabling or disabling of comparator output
0	Disables output (output signal = fixed to low level)
1	Enables output

C0INV	Output reversal setting
0	Forward
1	Reverse

Note Setting prohibited in the 78K0/FA2-L.

Cautions 1. Rewrite C0DFS1, C0DFS0, C0MODSEL1, C0MODSEL0, C0INV after setting the comparator 0 operation to the disabled state (CMP0EN = 0).

2. With the noise elimination width, an extra peripheral hardware clock frequency (f_{PRS}) may be eliminated from the setting value.

3. If the comparator output noise interval is within "set noise elimination width + 1 clock", an illegal waveform may be output.

4. To use the internal reference voltage, enable (CVRE = 1) operation of the internal reference voltage before enabling (CMP0EN = 1) the comparator operation.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 12-3. Format of Comparator 1 Control Register (C1CTL) (78K0/FA2-L, 78K0/FB2-L)

Address: FF64H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
C1CTL	CMP1EN	C1DFS1	C1DFS0	C1MODSEL1	C1MODSEL0	0	C1OE	C1INV

CMP1EN	Comparator 1 operation control
0	Stops operation
1	Enables operation Enables input to the external pins (CMP1+) on the positive and negative sides of comparator 1

C1DFS1	C1DFS0	Noise elimination width setting
0	0	Noise filter unused
0	1	$2/f_{PRS}$
1	0	$2^2/f_{PRS}$
1	1	$2^4/f_{PRS}$

C1MODSEL1	C1MODSEL0	Reference voltage selection
0	0	Internal reference voltage: DA0
0	1	Internal reference voltage: DA1
1	0	Internal reference voltage: DA2
1	1	Internal reference voltage: CMPCOM**Note**

C1OE	Enabling or disabling of comparator output
0	Disables output (output signal = fixed to low level)
1	Enables output

C1INV	Output reversal setting
0	Forward
1	Reverse

Note Setting prohibited in the 78K0/FA2-L.

Cautions 1. Rewrite C1DFS1, C1DFS0, C1MODSEL1, C1MODSEL0, C1INV after setting the comparator 1 operation to the disabled state (CMP1EN = 0).

2. With the noise elimination width, an extra peripheral hardware clock frequency (f_{PRS}) may be eliminated from the setting value.

3. If the comparator output noise interval is within "set noise elimination width + 1 clock", an illegal waveform may be output.

4. To use the internal reference voltage, enable (CVRE = 1) operation of the internal reference voltage before enabling (CMP1EN = 1) the comparator operation.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 12-4. Format of Comparator 2 Control Register (C2CTL)

Address: FF66H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
C2CTL	CMP2EN	C2DFS1	C2DFS0	C2MODSEL1	C2MODSEL0	0	C2OE	C2INV

CMP2EN	Comparator 2 operation control
0	Stops operation
1	Enables operation Enables input to the external pins (CMP2+) on the positive and negative sides of comparator 2

C2DFS1	C2DFS0	Noise elimination width setting
0	0	Noise filter unused
0	1	$2/f_{PRS}$
1	0	$2^2/f_{PRS}$
1	1	$2^4/f_{PRS}$

C2MODSEL1	C2MODSEL0	Reference voltage selection
0	0	Internal reference voltage: DA0
0	1	Internal reference voltage: DA1
1	0	Internal reference voltage: DA2
1	1	Internal reference voltage: CMPCOM[Note]

C2OE	Enabling or disabling of comparator output
0	Disables output (output signal = fixed to low level)
1	Enables output

C2INV	Output reversal setting
0	Forward
1	Reverse

Note Setting prohibited in the 78K0/FY2-L and 78K0/FA2-L.

Cautions 1. Rewrite C2DFS1, C2DFS0, C2MODSEL1, C2MODSEL0, C2INV after setting the comparator 2 operation to the disabled state (CMP2EN = 0).

2. With the noise elimination width, an extra peripheral hardware clock frequency (f_{PRS}) may be eliminated from the setting value.

3. If the comparator output noise interval is within "set noise elimination width + 1 clock", an illegal waveform may be output.

4. To use the internal reference voltage, enable (CVRE = 1) operation of the internal reference voltage before enabling (CMP2EN = 1) the comparator operation.

Remark f_{PRS}: Peripheral hardware clock frequency

(2) DAn internal reference voltage selection register (CnRVM)

This register is used to set the internal reference voltage level of comparator.

This register also controls the internal reference voltage generation operation by using bit 7 (CVRE) of C0RVM.

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

Reset signal generation clears this register to 00H.

Figure 12-5. Format of DA0 Internal Reference Voltage Selection Register (C0RVM)

Address: FF63H After reset: 00H R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
C0RVM	CVRE	0	0	C0VRS4	C0VRS3	C0VRS2	C0VRS1	C0VRS0

CVRE	Internal reference voltage generation operation control
0	Stops operation
1	Enables operation

C0VRS4	C0VRS3	C0VRS2	C0VRS1	C0VRS0	Reference voltage level (DA0) setting
0	0	0	0	0	0.05 V (TYP.)
0	0	0	0	1	0.1 V (TYP.)
0	0	0	1	0	0.15 V (TYP.)
0	0	0	1	1	0.2 V (TYP.)
0	0	1	0	0	0.25 V (TYP.)
0	0	1	0	1	0.3 V (TYP.)
0	0	1	1	0	0.35 V (TYP.)
0	0	1	1	1	0.4 V (TYP.)
0	1	0	0	0	0.44 V (TYP.)
0	1	0	0	1	0.49 V (TYP.)
0	1	0	1	0	0.54 V (TYP.)
0	1	0	1	1	0.59 V (TYP.)
0	1	1	0	0	0.64 V (TYP.)
0	1	1	0	1	0.69 V (TYP.)
0	1	1	1	0	0.74 V (TYP.)
0	1	1	1	1	0.79 V (TYP.)
1	0	0	0	0	0.84 V (TYP.)
1	0	0	0	1	0.89 V (TYP.)
1	0	0	1	0	0.94 V (TYP.)
1	0	0	1	1	0.99 V (TYP.)
1	0	1	0	0	1.04 V (TYP.)
1	0	1	0	1	1.09 V (TYP.)
1	0	1	1	0	1.14 V (TYP.)
1	0	1	1	1	1.19 V (TYP.)
1	1	0	0	0	1.23 V (TYP.)
1	1	0	0	1	1.28 V (TYP.)
1	1	0	1	0	1.33 V (TYP.)
1	1	0	1	1	1.38 V (TYP.)
1	1	1	0	0	1.43 V (TYP.)
1	1	1	0	1	1.48 V (TYP.)
1	1	1	1	0	1.53 V (TYP.)
1	1	1	1	1	1.58 V (TYP.)

Caution To change the reference voltage level when the internal reference voltage generation operation is enabled (CVRE = 1), a voltage stabilization wait time is required. See Figure 12-12 Example of Procedure for Changing Internal Reference Voltage for the setting method.

Figure 12-6. Format of DA1 Internal Reference Voltage Selection Register (C1RVM)

Address: FF65H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
C1RVM	0	0	0	C1VRS4	C1VRS3	C1VRS2	C1VRS1	C1VRS0

C1VRS4	C1VRS3	C1VRS2	C1VRS1	C1VRS0	Reference voltage level (DA1) setting
0	0	0	0	0	0.05 V (TYP.)
0	0	0	0	1	0.1 V (TYP.)
0	0	0	1	0	0.15 V (TYP.)
0	0	0	1	1	0.2 V (TYP.)
0	0	1	0	0	0.25 V (TYP.)
0	0	1	0	1	0.3 V (TYP.)
0	0	1	1	0	0.35 V (TYP.)
0	0	1	1	1	0.4 V (TYP.)
0	1	0	0	0	0.44 V (TYP.)
0	1	0	0	1	0.49 V (TYP.)
0	1	0	1	0	0.54 V (TYP.)
0	1	0	1	1	0.59 V (TYP.)
0	1	1	0	0	0.64 V (TYP.)
0	1	1	0	1	0.69 V (TYP.)
0	1	1	1	0	0.74 V (TYP.)
0	1	1	1	1	0.79 V (TYP.)
1	0	0	0	0	0.84 V (TYP.)
1	0	0	0	1	0.89 V (TYP.)
1	0	0	1	0	0.94 V (TYP.)
1	0	0	1	1	0.99 V (TYP.)
1	0	1	0	0	1.04 V (TYP.)
1	0	1	0	1	1.09 V (TYP.)
1	0	1	1	0	1.14 V (TYP.)
1	0	1	1	1	1.19 V (TYP.)
1	1	0	0	0	1.23 V (TYP.)
1	1	0	0	1	1.28 V (TYP.)
1	1	0	1	0	1.33 V (TYP.)
1	1	0	1	1	1.38 V (TYP.)
1	1	1	0	0	1.43 V (TYP.)
1	1	1	0	1	1.48 V (TYP.)
1	1	1	1	0	1.53 V (TYP.)
1	1	1	1	1	1.58 V (TYP.)

Caution To change the reference voltage level when the internal reference voltage generation operation is enabled (CVRE = 1), a voltage stabilization wait time is required. See Figure 12-12 Example of Procedure for Changing Internal Reference Voltage for the setting method.

Figure 12-7. Format of DA2 Internal Reference Voltage Selection Register (C2RVM)

Address: FF67H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
C2RVM	0	0	0	C2VRS4	C2VRS3	C2VRS2	C2VRS1	C2VRS0

C2VRS4	C2VRS3	C2VRS2	C2VRS1	C2VRS0	Reference voltage level (DA2) setting
0	0	0	0	0	0.05 V (TYP.)
0	0	0	0	1	0.1 V (TYP.)
0	0	0	1	0	0.15 V (TYP.)
0	0	0	1	1	0.2 V (TYP.)
0	0	1	0	0	0.25 V (TYP.)
0	0	1	0	1	0.3 V (TYP.)
0	0	1	1	0	0.35 V (TYP.)
0	0	1	1	1	0.4 V (TYP.)
0	1	0	0	0	0.44 V (TYP.)
0	1	0	0	1	0.49 V (TYP.)
0	1	0	1	0	0.54 V (TYP.)
0	1	0	1	1	0.59 V (TYP.)
0	1	1	0	0	0.64 V (TYP.)
0	1	1	0	1	0.69 V (TYP.)
0	1	1	1	0	0.74 V (TYP.)
0	1	1	1	1	0.79 V (TYP.)
1	0	0	0	0	0.84 V (TYP.)
1	0	0	0	1	0.89 V (TYP.)
1	0	0	1	0	0.94 V (TYP.)
1	0	0	1	1	0.99 V (TYP.)
1	0	1	0	0	1.04 V (TYP.)
1	0	1	0	1	1.09 V (TYP.)
1	0	1	1	0	1.14 V (TYP.)
1	0	1	1	1	1.19 V (TYP.)
1	1	0	0	0	1.23 V (TYP.)
1	1	0	0	1	1.28 V (TYP.)
1	1	0	1	0	1.33 V (TYP.)
1	1	0	1	1	1.38 V (TYP.)
1	1	1	0	0	1.43 V (TYP.)
1	1	1	0	1	1.48 V (TYP.)
1	1	1	1	0	1.53 V (TYP.)
1	1	1	1	1	1.58 V (TYP.)

Caution To change the reference voltage level when the internal reference voltage generation operation is enabled (CVRE = 1), a voltage stabilization wait time is required. See Figure 12-12 Example of Procedure for Changing Internal Reference Voltage for the setting method.

(3) Comparator output flag register (CMPFLG)

This register indicates the comparator output level.

This register is read-only by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-8. Format of Comparator Output Flag Register (CMPFLG)

(a) 78K0/FY2-L

Address: FF69H After reset: 00H R

Symbol	7	6	5	4	3	<2>	1	0
CMPFLG	0	0	0	0	0	CMP2F	0	0

(b) 78K0/FA2-L, 78K0/FB2-L

Address: FF69H After reset: 00H R

Symbol	7	6	5	4	3	<2>	<1>	<0>
CMPFLG	0	0	0	0	0	CMP2F	CMP1F	CMP0F

CMPnF	Comparator n output level (n = 0 to 2)
0	Low level
1	High level

(4) A/D port configuration register 0 (ADPC0)

ADPC0 switches the P20/ANI0 to P27/ANI7 pins to digital I/O or analog input of port. Each bit of ADPC0 corresponds to a pin of port 2 and can be specified in 1-bit units.

When CMP0+/P24/ANI4, CMP1+/P25/ANI5, and CMP2+/P23/ANI3 pins, and CMPCOM/P26/ANI6 pin[Note] are used for the comparator input and comparator common input respectively, set these pins to analog input by ADPC0.

ADPC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears ADPC0 to 00H.

Note 78K0/FB2-L only

Figure 12-9. Format of A/D Port Configuration Register 0 (ADPC0)

(a) 78K0/FY2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPCS3	ADPCS2	ADPCS1	ADPCS0

(b) 78K0/FA2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	ADPCS5	ADPCS4	ADPCS3	ADPCS2	ADPCS1	ADPCS0

(c) 78K0/FB2-L

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	ADPCS7	ADPCS6	ADPCS5	ADPCS4	ADPCS3	ADPCS2	ADPCS1	ADPCS0

ADPCSn	Digital I/O or analog input selection (n = 0 to 7)
0	Analog input
1	Digital I/O

Cautions 1. Set the pin set to analog input to the input mode by using port mode register 2 (PM2).

2. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the peripheral hardware clock is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.

(5) External interrupt rising edge enable registers (EGPCTL0, EGPCTL1), external interrupt falling edge enable registers (EGNCTL0, EGNCTL1)

EGPCTL0, EGPCTL1, EGNCTL0, and EGNCTL1 are the registers that set the INTCMP0 to INTCMP2 valid edges.

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

Reset signal generation clears these registers to 00H.

Figure 12-10. Format of External Interrupt Rising Edge Enable Registers (EGPCTL0, EGPCTL1) and External Interrupt Falling Edge Enable Registers (EGNCTL0, EGNCTL1) (1/3)

(a) 78K0/FY2-L

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL0	0	0	0	0	0	0	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL0	0	0	0	0	0	0	EGN1	EGN0

Address: FF4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL1	0	0	0	0	0	0	0	EGP8

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL1	0	0	0	0	0	0	0	EGN8

EGPn	EGNn	INTCMP2 valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution Be sure to clear bits 2 to 7 of EGPCTL0 and EGNCTL0 to 0 in the 78K0/FY2-L.

Remark n = 0, 1, 8

Figure 12-10. Format of External Interrupt Rising Edge Enable Registers (EGPCTL0, EGPCTL1) and External Interrupt Falling Edge Enable Registers (EGNCTL0, EGNCTL1) (2/3)

(b) 78K0/FA2-L

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL0	EGP7	EGP6	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL0	EGN7	EGN6	0	0	EGN3	EGN2	EGN1	EGN0

Address: FF4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL1	0	0	0	0	0	0	0	EGP8

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL1	0	0	0	0	0	0	0	EGN8

EGPn	EGNn	INTCMP to INTCMP2 valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution Be sure to clear bits 4 to 7 of EGPCTL0 and EGNCTL0 to 0 in the 78K0/FA2-L.

Remark n = 0 to 3, 6 to 8

Figure 12-10. Format of External Interrupt Rising Edge Enable Registers (EGPCTL0, EGPCTL1) and External Interrupt Falling Edge Enable Registers (EGNCTL0, EGNCTL1) (3/3)

(c) 78K0/FB2-L

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FF4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL1	0	0	0	0	0	0	0	EGP8

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL1	0	0	0	0	0	0	0	EGN8

EGPn	EGNn	INTCMP0 to INTCMP2 valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution Be sure to clear bits 1 to 7 of EGPCTL1 and EGNCTL1 to 0 in the 78K0/FB2-L.

Remark n = 0 to 8

Table 12-2 shows the interrupt request signal corresponding to EGPn and EGNn.

Table 12-2 Interrupt Request Signal Corresponding to EGPn and EGNn

(a) 78K0/FY2-L

Detection Enable Bit		Edge Detection Signal	Interrupt Request Signal
EGP8	EGN8	CMP2 output	INTCMP2

(b) 78K0/FA2-L, 78K0/FB2-L

Detection Enable Bit		Edge Detection Signal	Interrupt Request Signal
EGP6	EGN6	CMP0 output	INTCMP0
EGP7	EGN7	CMP1 output	INTCMP1
EGP8	EGN8	CMP2 output	INTCMP2

Remark n = 8: 78K0/FY2-L
n = 6 to 8: 78K0/FA2-L, 78K0/FB2-L

(6) Port mode register 2 (PM2)

This register is used to set port 2 input or output in 1-bit units.

When CMP0+/P24/ANI4, CMP1+/P25/ANI5, and CMP2+/P23/ANI3 pins, and CMPCOM/P26/ANI6 pin[Note] are used for the comparator input and comparator common input respectively, set PM23 to PM25, PM26 bits to 1.

The output latches of P23 to P25, P26 at this time may be 0 or 1.

PM2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Note 78K0/FB2-L only

Figure 12-11. Format of Port Mode Register 2 (PM2)

(a) 78K0/FY2-L

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

Caution Be sure to set bits 4 to 7 of PM2 to 1.

(b) 78K0/FA2-L

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20

Caution Be sure to set bits 6 and 7 of PM2 to 1.

(c) 78K0/FB2-L

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)

Caution If the internal reference voltage is used, the port function shared by the CMPCOM pin can be used in input mode. Using the port function in output mode, however, is prohibited. Also, accessing port register 2 (P2) is prohibited.

When using P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+, P26/ANI6/CMPCOM, set the registers according to the pin function to be used (refer to **Tables 12-3** and **12-4**).

Table 12-3. Setting Functions of P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins

ADPC0 Register	PM2 Register	CMPmEN bit (m = 0 to 2)	ADS Register (n = 3 to 5)	P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins
Digital I/O selection	Input mode	–	Selects ANIn.	Setting prohibited
			Does not select ANIn.	Digital input
	Output mode	–	Selects ANIn.	Setting prohibited
			Does not select ANIn.	Digital output
Analog input selection	Input mode	0	Selects ANIn.	Analog input (to be converted into digital signal)
			Does not select ANIn.	Analog input (not to be converted into digital signal)
		1	Selects ANIn.	Analog input (to be converted into digital signal), and comparator input
			Does not select ANIn.	Comparator input
	Output mode	–	–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
CMPmEN: Bit 7 of comparator m control register (CmCTL)
ADS: Analog input channel specification register

Table 12-4. Setting Functions of P26/ANI6/CMPCOM Pin

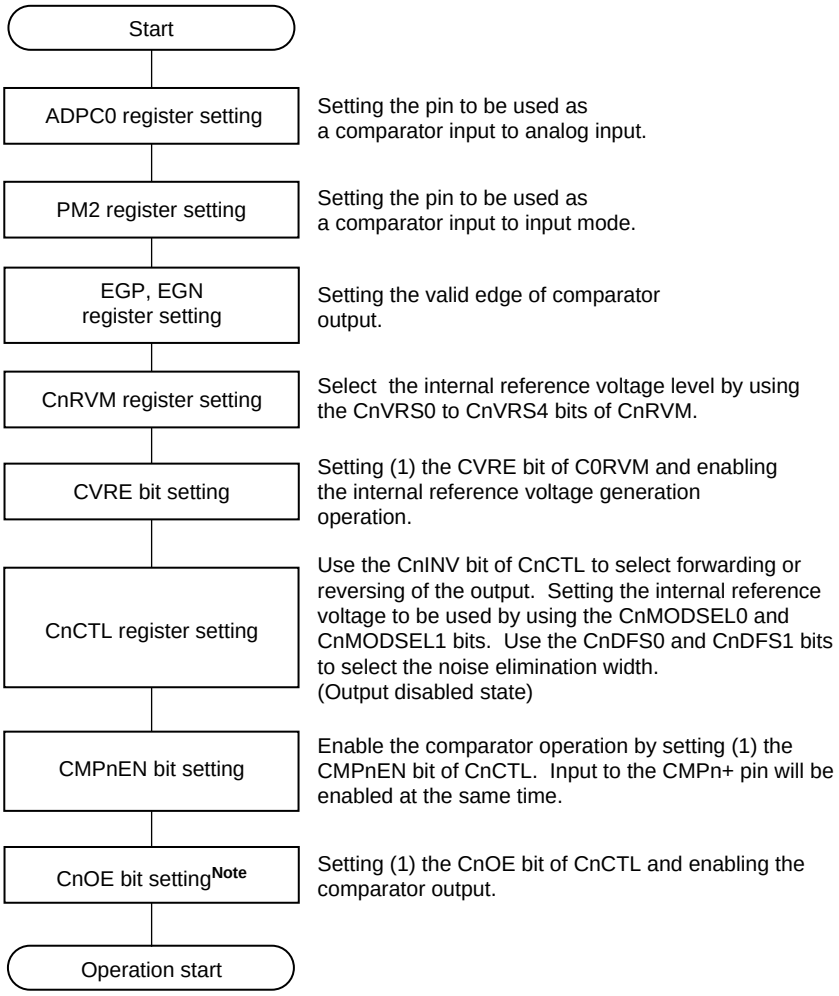
ADPC0 Register	PM2 Register	CmMODSEL1 bit (m = 0 to 2)	CmMODSEL0 bit (m = 0 to 2)	ADS Register	P26/ANI6/CMPCOM Pin
Digital I/O selection	Input mode	–		Selects ANI6.	Setting prohibited
				Does not select ANI6.	Digital input
	Output mode	–		Selects ANI6.	Setting prohibited
				Does not select ANI6.	Digital output
Analog input selection	Input mode	CmMODSEL1 = 0, or CmMODSEL0 = 0		Selects ANI6.	Analog input (to be converted into digital signal)
				Does not select ANI6.	Analog input (not to be converted into digital signal)
		CmMODSEL1 = 1, CmMODSEL0 = 1		Selects ANI6.	Analog input (to be converted into digital signal), and comparator common input
				Does not select ANI6.	Comparator common input
	Output mode	–		–	Setting prohibited

Remark ADPC0: A/D port configuration register 0
PM2: Port mode register 2
CmMODSEL1, CmMODSEL0: Bits 4 and 3 of comparator m control register (CmCTL)
ADS: Analog input channel specification register

12.4 Operations of Comparators

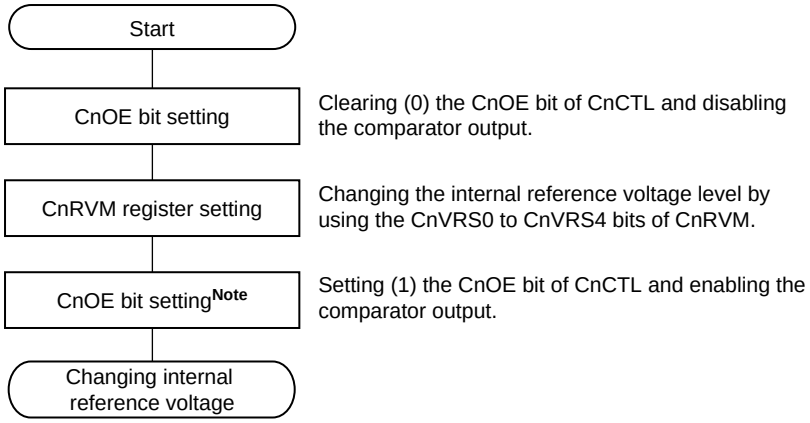
12.4.1 Starting comparator operation (using internal reference voltage for comparator reference voltage)

Figure 12-12. Example of Setting Procedure When Starting Comparator Operation (Using Internal Reference Voltage for Comparator Reference Voltage)



Note Set the CnOE bit to 1 when at least 20 μs have elapsed after having set the CVRE bit.

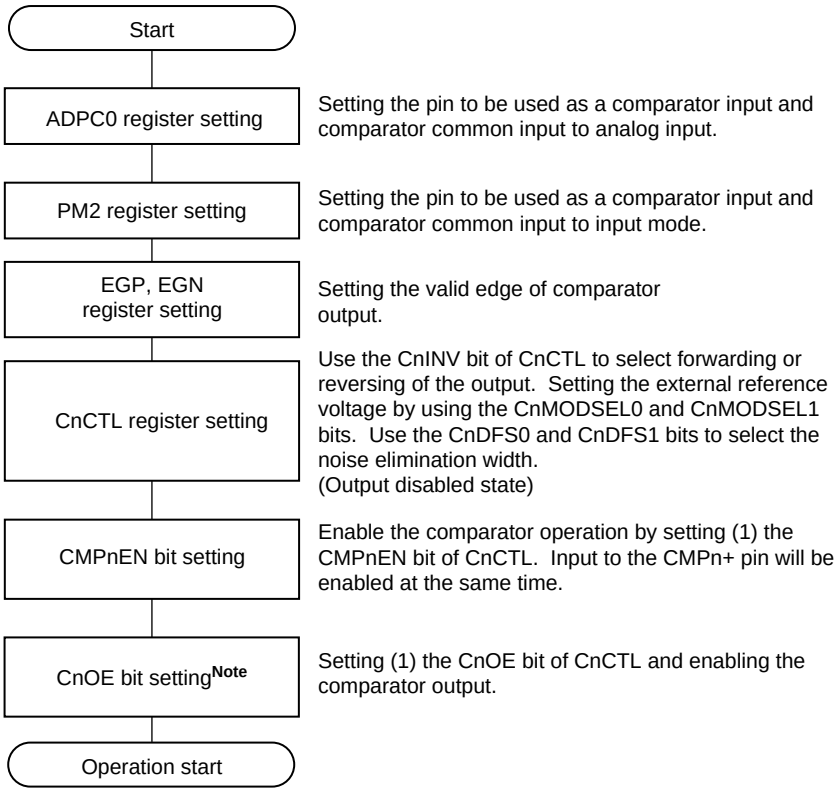
Figure 12-13. Example of Setting Procedure When Changing Internal Reference Voltage



Note Set the CnOE bit to 1 when at least 5 μs have elapsed after having set the CnRVM register.

12.4.2 Starting comparator operation (using input voltage from CMPCOM pin for comparator reference voltage)

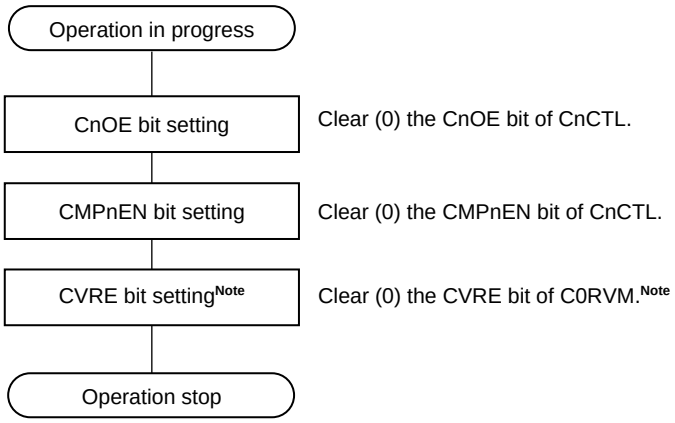
Figure 12-14. Example of Setting Procedure When Starting Comparator Operation (Using Input Voltage from Comparator Common (CMPCOM) Pin for Comparator Reference Voltage (78K0/FB2-L Only))



Note Set the CnOE bit to 1 when at least 1 μs have elapsed after having set the CMPnEN bit.

12.4.3 Stopping comparator operation

Figure 12-15. Example of Setting Procedure When Stopping Comparator Operation



Note Only when using the internal reference voltage.

CHAPTER 13 SERIAL INTERFACE UART6

13.1 Functions of Serial Interface UART6

Serial interface UART6 are mounted onto all 78K0/Fx2-L microcontroller products.
Serial interface UART6 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

(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, refer to **13.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode supports the LIN (Local Interconnect Network)-bus. The functions of this mode are outlined below.
For details, refer to **13.4.2 Asynchronous serial interface (UART) mode** and **13.4.3 Dedicated baud rate generator**.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD6: Transmit data output pin
  RxD6: 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 (SBF reception flag provided).

Cautions 1. The TxD6 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 UART6 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART6 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6 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 POWER6 = 0, RXE6 = 0, and TXE6 = 0.

3. Set POWER6 = 1 and then set TXE6 = 1 (transmission) or RXE6 = 1 (reception) to start communication.

4. TXE6 and RXE6 are synchronized by the base clock (f_{XCLK6}) set by CKSR6. To enable transmission or reception again, set TXE6 or RXE6 to 1 at least two clocks of the base clock after TXE6 or RXE6 has been cleared to 0. If TXE6 or RXE6 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.

Caution 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 the continuous transmission function if the interface is used in LIN communication operation.

Remark 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.

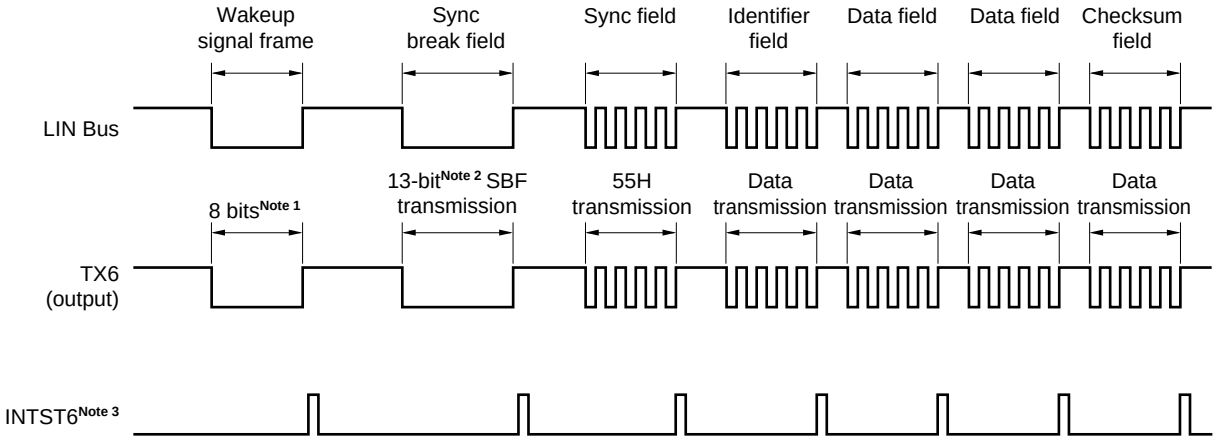
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

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 13-1 and 13-2 outline the transmission and reception operations of LIN.

Figure 13-1. LIN Transmission Operation



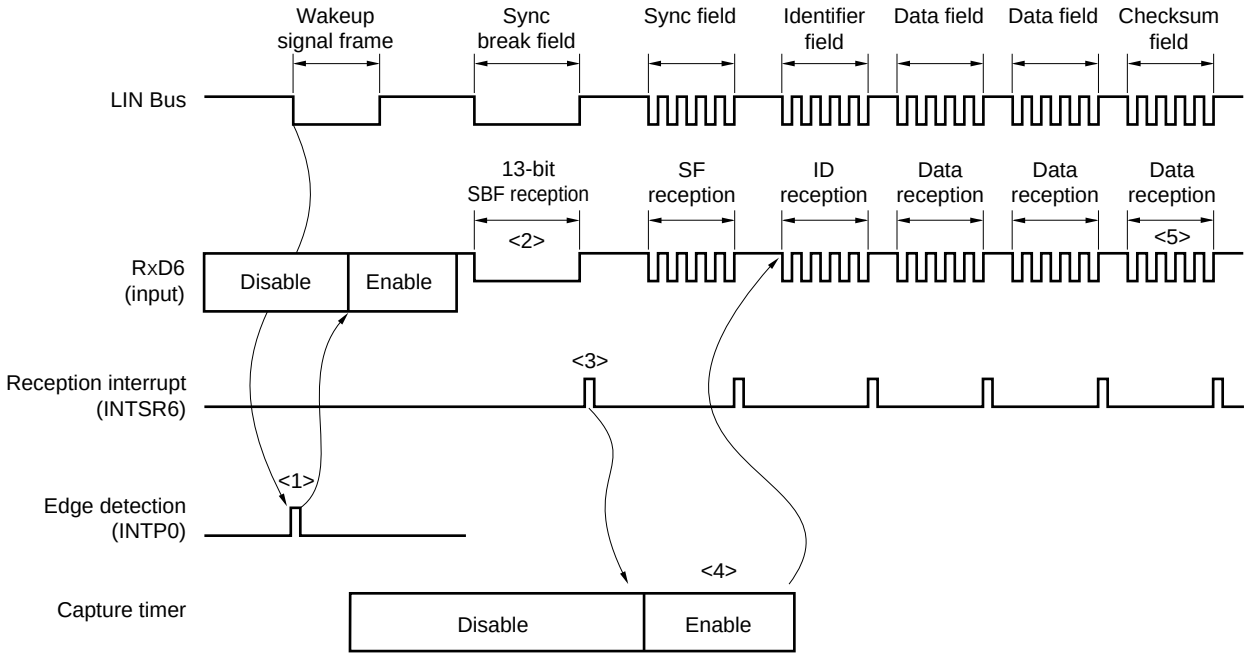
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6) (refer to **13.4.2 (2) (h) SBF transmission**).

3. INTST6 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 13-2. LIN Reception Operation



Reception processing is as follows.

<1> The wakeup signal is detected at the edge of the pin, and enables UART6 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 (refer to **7.4.8 Pulse width measurement operation**). Detection of errors OVE6, PE6, and FE6 is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6 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 UART6 after SF reception, and then re-set baud rate generator control register 6 (BRGC6).

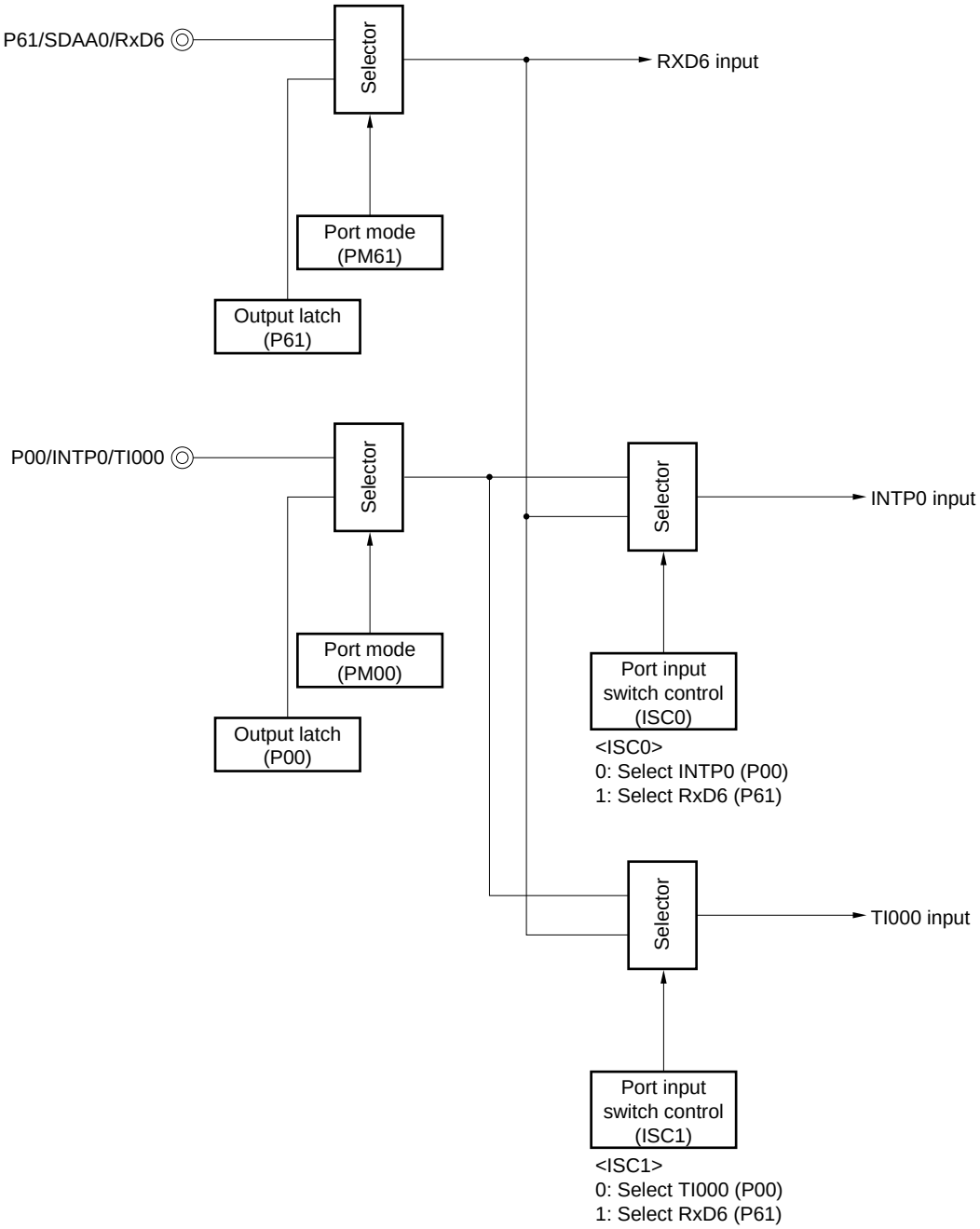
<5> Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

Figure 13-3 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 (RxD6) can be input to the external interrupt (INTP0) and 16-bit timer/event counter 00 by port input switch control (ISC0/ISC1), without connecting RxD6 and INTP0/TI000 externally.

Figure 13-3. Port Configuration for LIN Reception Operation



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (refer to **Figure 13-11**)

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 UART6

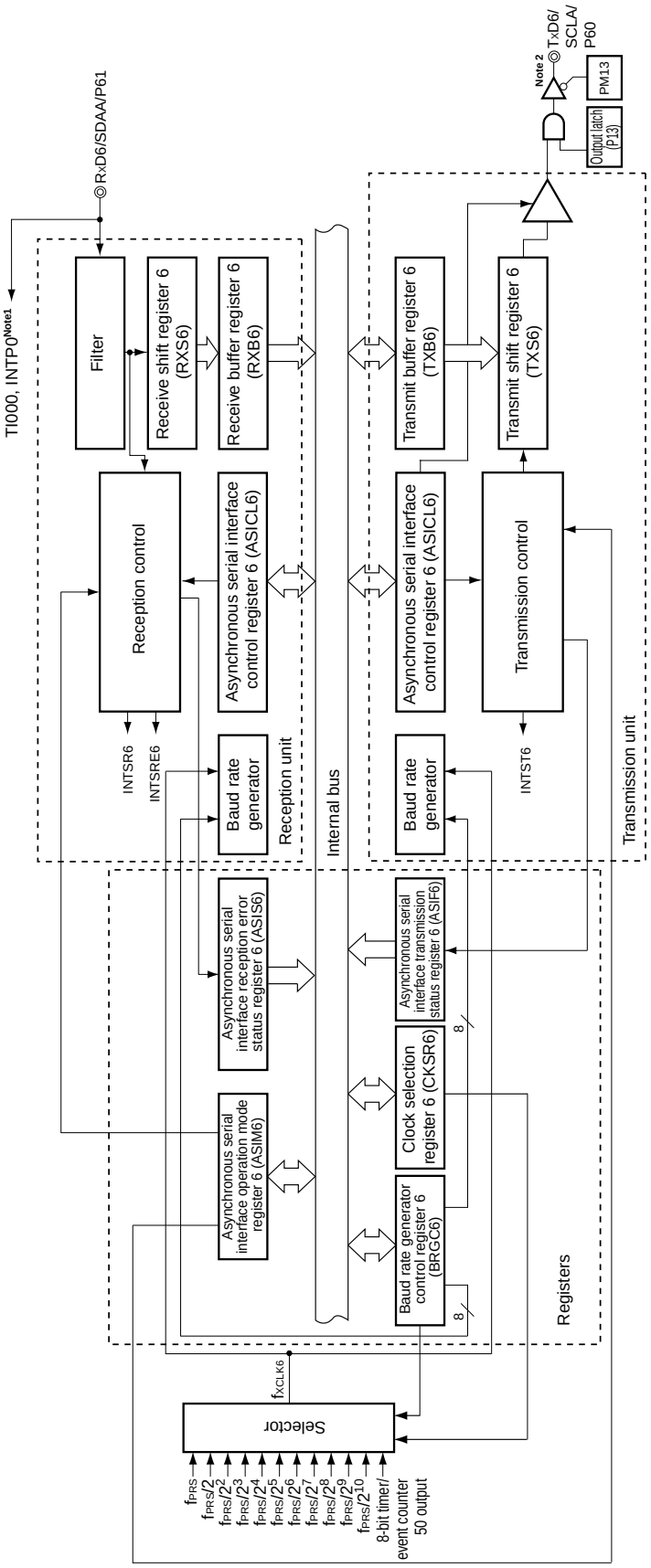
13.2 Configuration of Serial Interface UART6

Serial interface UART6 includes the following hardware.

Table 13-1. Configuration of Serial Interface UART6

Item	Configuration
Registers	Receive buffer register 6 (RXB6) Receive shift register 6 (RXS6) Transmit buffer register 6 (TXB6) Transmit shift register 6 (TXS6)
Control registers	Asynchronous serial interface operation mode register 6 (ASIM6) Asynchronous serial interface reception error status register 6 (ASIS6) Asynchronous serial interface transmission status register 6 (ASIF6) Clock selection register 6 (CKSR6) Baud rate generator control register 6 (BRGC6) Asynchronous serial interface control register 6 (ASICL6) Input switch control register (ISC) Port mode register 6 (PM6) Port register 6 (P6) Port output mode register 6 (POM6)

Figure 13-4. Block Diagram of Serial Interface UART6



Notes 1. Selectable with input switch control register (ISC).

2. When using the P60/TxD6/SCLA0 pin as the data output of serial interface UART6, clear POM60 to 0

(1) Receive buffer register 6 (RXB6)

This 8-bit register stores parallel data converted by receive shift register 6 (RXS6).

Each time 1 byte of data has been received, new receive data is transferred to this register from RXS6. 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 RXB6 and the MSB of RXB6 is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6 and the LSB of RXB6 is always 0.

If an overrun error (OVE6) occurs, the receive data is not transferred to RXB6.

RXB6 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 6 (RXS6)

This register converts the serial data input to the RxD6 pin into parallel data.

RXS6 cannot be directly manipulated by a program.

(3) Transmit buffer register 6 (TXB6)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6.

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 TXB6 when bit 1 (TXBF6) of asynchronous serial interface transmission status register 6 (ASIF6) is 1.

2. Do not refresh (write the same value to) TXB6 by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) are 1 or when bits 7 and 5 (POWER6, RXE6) of ASIM6 are 1).

3. Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.

(4) Transmit shift register 6 (TXS6)

This register transmits the data transferred from TXB6 from the TxD6 pin as serial data. Data is transferred from TXB6 immediately after TXB6 is written for the first transmission, or immediately before INTST6 occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6 and transmitted from the TxD6 pin at the falling edge of the base clock.

TXS6 cannot be directly manipulated by a program.

13.3 Registers Controlling Serial Interface UART6

Serial interface UART6 is controlled by the following ten registers.

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 6 (PM6)
- Port register 6 (P6)
- Port output mode register 6 (POM6)

(1) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial communication operations of serial interface UART6.

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

Reset signal generation sets this register to 01H.

Remark ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 13-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	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

TXE6	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. The output of the TxD6 pin is fixed to the high level (when TXDLV6 = 0) and the input from the RxD6 pin is fixed to the high level when POWER6 = 0 during transmission.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Figure 13-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (2/2)

PS61	PS60	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.

CL6	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6	Enables/disables occurrence of reception completion interrupt in case of error
0	“INTSRE6” occurs in case of error (at this time, INTSR6 does not occur).
1	“INTSR6” occurs in case of error (at this time, INTSRE6 does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE6) of asynchronous serial interface reception error status register 6 (ASIS6) is not set and the error interrupt does not occur.

Cautions
1. **To start the transmission, set POWER6 to 1 and then set TXE6 to 1. To stop the transmission, clear TXE6 to 0, and then clear POWER6 to 0.**
2. **To start the reception, set POWER6 to 1 and then set RXE6 to 1. To stop the reception, clear RXE6 to 0, and then clear POWER6 to 0.**
3. **Set POWER6 to 1 and then set RXE6 to 1 while a high level is input to the RxD6 pin. If POWER6 is set to 1 and RXE6 is set to 1 while a low level is input, reception is started.**
4. **TXE6 and RXE6 are synchronized by the base clock (f_{XCLK6}) set by CKSR6. To enable transmission or reception again, set TXE6 or RXE6 to 1 at least two clocks of the base clock after TXE6 or RXE6 has been cleared to 0. If TXE6 or RXE6 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.**
5. **Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.**
6. **Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.**
7. **Fix the PS61 and PS60 bits to 0 when used in LIN communication operation.**
8. **Clear TXE6 to 0 before rewriting the SL6 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL6 bit.**
9. **Make sure that RXE6 = 0 when rewriting the ISRM6 bit.**

(2) Asynchronous serial interface reception error status register 6 (ASIS6)

This register indicates an error status on completion of reception by serial interface UART6. It includes three error flag bits (PE6, FE6, OVE6).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6) or bit 5 (RXE6) of ASIM6 to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS6 and then read UART receive buffer register 6 (RXB6) to clear the error flag.

Figure 13-6. Format of Asynchronous Serial Interface Reception Error Status Register 6 (ASIS6)

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	Status flag indicating parity error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If the parity of transmit data does not match the parity bit on completion of UART reception

FE6	Status flag indicating framing error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If the stop bit is not detected on completion of UART reception

OVE6	Status flag indicating overrun error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If receive data is set to the RXB6 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE6 bit differs depending on the set values of the PS61 and PS60 bits of asynchronous serial interface operation mode register 6 (ASIM6).

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 6 (RXB6) but discarded.

4. If data is read from ASIS6, a wait cycle is generated. Do not read data from ASIS6 when the peripheral hardware clock (f_{PRS}) is stopped. For details, refer to CHAPTER 30 CAUTIONS FOR WAIT.

(3) Asynchronous serial interface transmission status register 6 (ASIF6)

This register indicates the status of transmission by serial interface UART6. It includes two status flag bits (TXBF6 and TXSF6).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6 register after data has been transferred from the TXB6 register to the TXS6 register.

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6) or bit 6 (TXE6) of ASIM6 to 0 clears this register to 00H.

Figure 13-7. Format of Asynchronous Serial Interface Transmission Status Register 6 (ASIF6)

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	Transmit buffer data flag
0	If POWER6 = 0 or TXE6 = 0, or if data is transferred to UART transmit shift register 6 (TXS6)
1	If data is written to transmit buffer register 6 (TXB6) (if data exists in TXB6)

TXSF6	Transmit shift register data flag
0	If POWER6 = 0 or TXE6 = 0, or if the next data is not transferred from transmit buffer register 6 (TXB6) after completion of transfer
1	If data is transferred from transmit buffer register 6 (TXB6) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 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 TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

(4) Clock selection register 6 (CKSR6)

This register selects the base clock of serial interface UART6.

CKSR6 can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Remark CKSR6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 13-8. Format of Clock Selection Register 6 (CKSR6)

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	Base clock (f_{XCLK6}) selection[Note]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz (when using PLL)
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
Other than above				Setting prohibited				

Note If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

Caution Make sure POWER6 = 0 when rewriting TPS63 to TPS60.

Remark f_{PRS}: Peripheral hardware clock frequency

(5) Baud rate generator control register 6 (BRGC6)

This register sets the division value of the 8-bit counter of serial interface UART6.

BRGC6 can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Remark BRGC6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 13-9. Format of Baud Rate Generator Control Register 6 (BRGC6)

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	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	$f_{XCLK6}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK6}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK6}/6$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

Cautions 1. Make sure that bit 6 (TXE6) and bit 5 (RXE6) of the ASIM6 register = 0 when rewriting the MDL67 to MDL60 bits.

2. The baud rate is the output clock of the 8-bit counter divided by 2.

Remarks 1. f_{XCLK6}: Frequency of base clock selected by the TPS63 to TPS60 bits of CKSR6 register

2. k: Value set by MDL67 to MDL60 bits (k = 4, 5, 6, ..., 255)

3. ×: Don't care

(6) Asynchronous serial interface control register 6 (ASICL6)

This register controls the serial communication operations of serial interface UART6.

ASICL6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 16H.

Caution ASICL6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1). However, do not set both SBRT6 and SBTT6 to 1 by a refresh operation during SBF reception (SBRF6 = 1) or SBF transmission (until INTST6 occurs since SBTT6 has been set (1)), because it may re-trigger SBF reception or SBF transmission.

Figure 13-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (1/2)

Address: FF58H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 13-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (2/2)

SBL62	SBL61	SBL60	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.

DIR6	First-bit specification
0	MSB
1	LSB

TXDLV6	Enables/disables inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

Cautions
1. **In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF6 flag is held (1).**
2. **Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1. After setting the SBRT6 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 SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.**
4. **Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1. After setting the SBTT6 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 SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.**
6. **Do not set the SBRT6 bit to 1 during reception, and do not set the SBTT6 bit to 1 during transmission.**
7. **Before rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.**
8. **When the TXDLV6 bit is set to 1 (inverted TxD6 output), the TxD6/SCLA0/P60 pin cannot be used as a general-purpose port, regardless of the settings of POWER6 and TXE6. When using the TxD6/SCLA0/P60 pin as a general-purpose port, clear the TXDLV6 bit to 0 (normal TxD6 output).**

(7) Input switch control register (ISC)

The input switch control register (ISC) is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception.

The signal input from the RxD6 pin is selected as the input source of INTP0 and TI000 when ISC0 and ISC1 are set to 1 (refer to **Figure 13-3 Port Configuration for LIN Reception Operation**).

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

Reset signal generation sets this register to 00H.

Figure 13-11. 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
1	RxD6

ISC0	INTP0 input source selection
0	INTP0
1	RxD6

Remark 78K0/FY2-L, 78K0/FA2-L: TI000/INTP0/P00, RxD6/SDAA0/P61
78K0/FB2-L: TI000/INTP0/P00, P121/X1/TOOLC0/<TI000>/<INTP0>, RxD6/SDAA0/P61

(8) Port mode register 6 (PM6)

This register set port 6 input/output in 1-bit units.

PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

When using the P60/TxD6/SCLA0 pin for serial interface data output, clear PM60 to 0 and set the output latch of P60 to 1.

When using the P61/RxD6/SDAA0 pin for serial interface data input, set PM61 to 1. The output latch of P61 at this time may be 0 or 1.

Figure 13-12. Format of Port Mode Register 6 (PM6)

Address: FF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(9) Port output mode register 6 (POM6)

This register sets the output mode of P60 and P61 in 1-bit units.

When using the P60/TxD6/SCLA0 pin as the data output of serial interface UART6/DALI, clear POM60 to 0.

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

Reset signal generation clears this register to 00H.

Figure 13-13. Format of Port Output Mode Register 6 (POM6)

Address: FF2AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM6	0	0	0	0	0	0	POM61	POM60

POM6n	P6n pin output mode selection (n = 0, 1)
0	Normal output (CMOS output) mode
1	N-ch open drain output (V_{DD} tolerance) mode

13.4 Operation of Serial Interface UART6

Serial interface UART6 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; 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 (POWER6, TXE6, and RXE6) of ASIM6 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6 (ASIM6).

ASIM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	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].

TXE6	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The output of the TxD6 pin is fixed to high level (when TXDLV6 = 0) and the input from the RxD6 pin is fixed to high level when POWER6 = 0 during transmission.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Caution Clear POWER6 to 0 after clearing TXE6 and RXE6 to 0 to stop the operation.
To start the communication, set POWER6 to 1, and then set TXE6 or RXE6 to 1.

Remark To use the RxD6/SDAA0/P61 and TxD6/SCLA0/P60 pins as general-purpose port pins, refer to **CHAPTER 4 PORT FUNCTIONS**.

13.4.2 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 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 6 (PM6)
- Port register 6 (P6)
- Port output mode register 6 (POM6)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6 register (refer to **Figure 13-8**).
<2> Set the BRGC6 register (refer to **Figure 13-9**).
<3> Set bits 0 to 4 (ISRM6, SL6, CL6, PS60, PS61) of the ASIM6 register (refer to **Figure 13-5**).
<4> Set bits 0 and 1 (TXDLV6, DIR6) of the ASICL6 register (refer to **Figure 13-10**).
<5> Set bit 7 (POWER6) of the ASIM6 register to 1.
<6> Set bit 6 (TXE6) of the ASIM6 register to 1. → Transmission is enabled.
Set bit 5 (RXE6) of the ASIM6 register to 1. → Reception is enabled.
<7> Write data to TXB6 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

POWER6	TXE6	RXE6	PM60	P60	PM61	P61	POM60	POM61	UART6 Operation	Pin Function	
										TxD6/ SCLA0/ P60	RxD6/ SDAA0/ P61
0	0	0	×Note	×Note	×Note	×Note	×Note	×Note	Stop	P60	P61
			0	1	0	1	1	1		SCLA0	SDAA0
1	0	1	×Note	×Note	1	×	×Note	×	Reception	P60	RxD6
	1	0	0	1	×Note	×Note	0	×Note	Transmission	TxD6	P61
	1	1	0	1	1	×	0	×	Transmission/ reception	TxD6	RxD6

Note Can be set as port function.

Remark × : don't care
POWER6 : Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6 : Bit 6 of ASIM6
RXE6 : Bit 5 of ASIM6
PM6× : Port mode register
P6× : Port output latch
POM60, POM61 : Bits 0 and 1 of Port output mode register 6 (POM6)

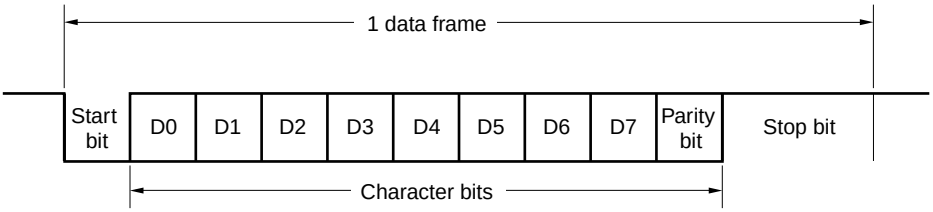
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

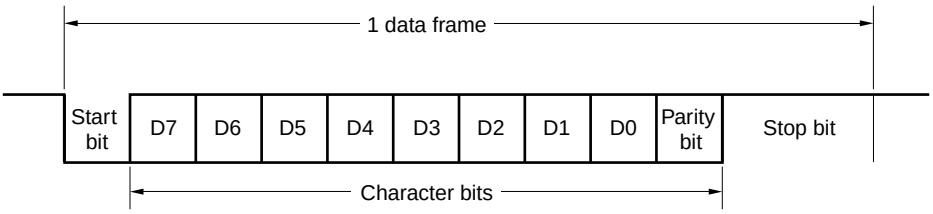
Figures 13-14 and 13-15 show the format and waveform example of the normal transmit/receive data.

Figure 13-14. 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

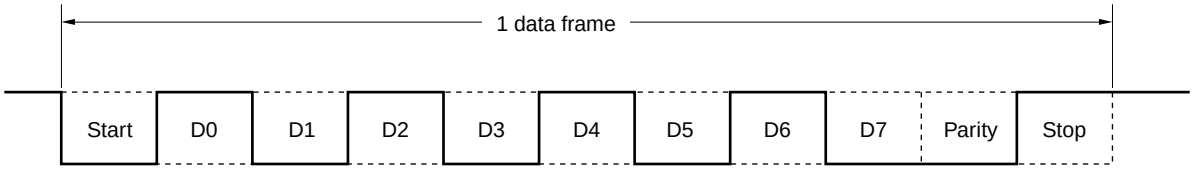
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6 (ASIM6).

Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).

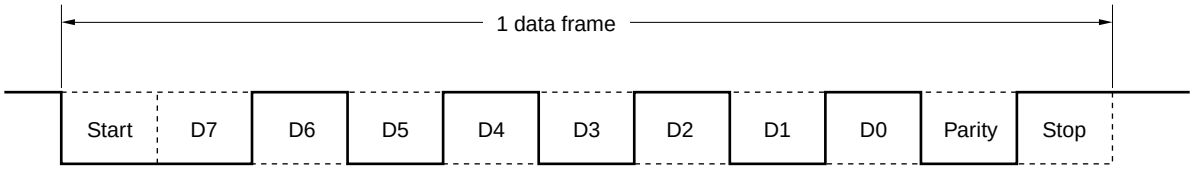
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 13-15. Example of Normal UART Transmit/Receive Data Waveform

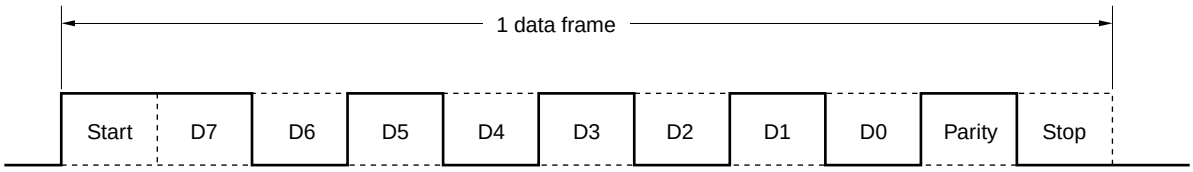
1. **Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



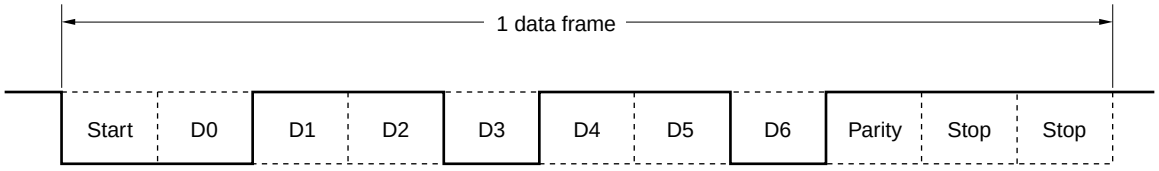
2. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



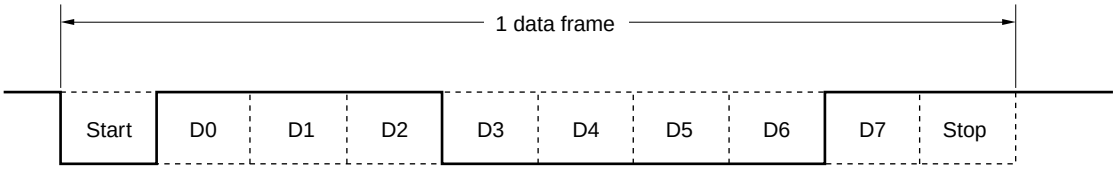
3. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6 pin inverted output**



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**



(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 PS61 and PS60 bits to 0 when the device 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 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and bit 6 (TXE6) of ASIM6 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6 (TXB6). The start bit, parity bit, and stop bit are automatically appended to the data.

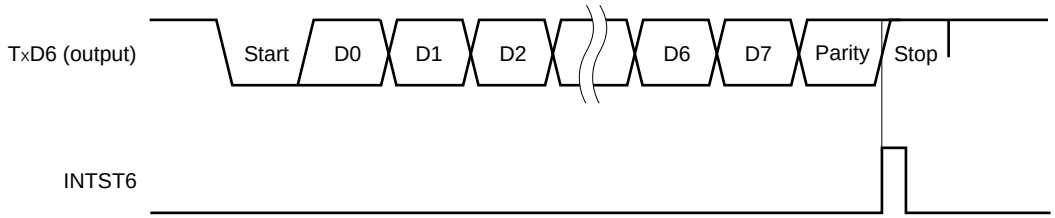
When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the transmit data is sequentially output from TXS6 to the TxD6 pin. When transmission is completed, the parity and stop bits set by ASIM6 are appended and a transmission completion interrupt request (INTST6) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6.

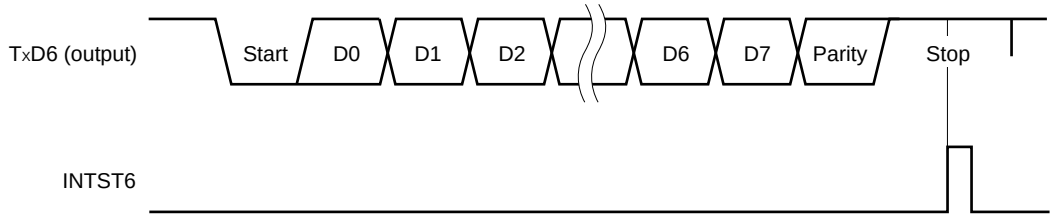
Figure 13-16 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

Figure 13-16. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6 (TXB6) as soon as transmit shift register 6 (TXS6) has started its shift operation. Consequently, even while the INTST6 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 TXB6 register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6 register to check the transmission status and whether the TXB6 register can be written, and then write the data.

Cautions 1. The TXBF6 and TXSF6 flags of the ASIF6 register change from "10" to "11", and to "01" during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6 and TXSF6 flags for judgment. Read only the TXBF6 flag when executing continuous transmission.

2. When the device is use in LIN communication operation, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

TXBF6	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6 flag.

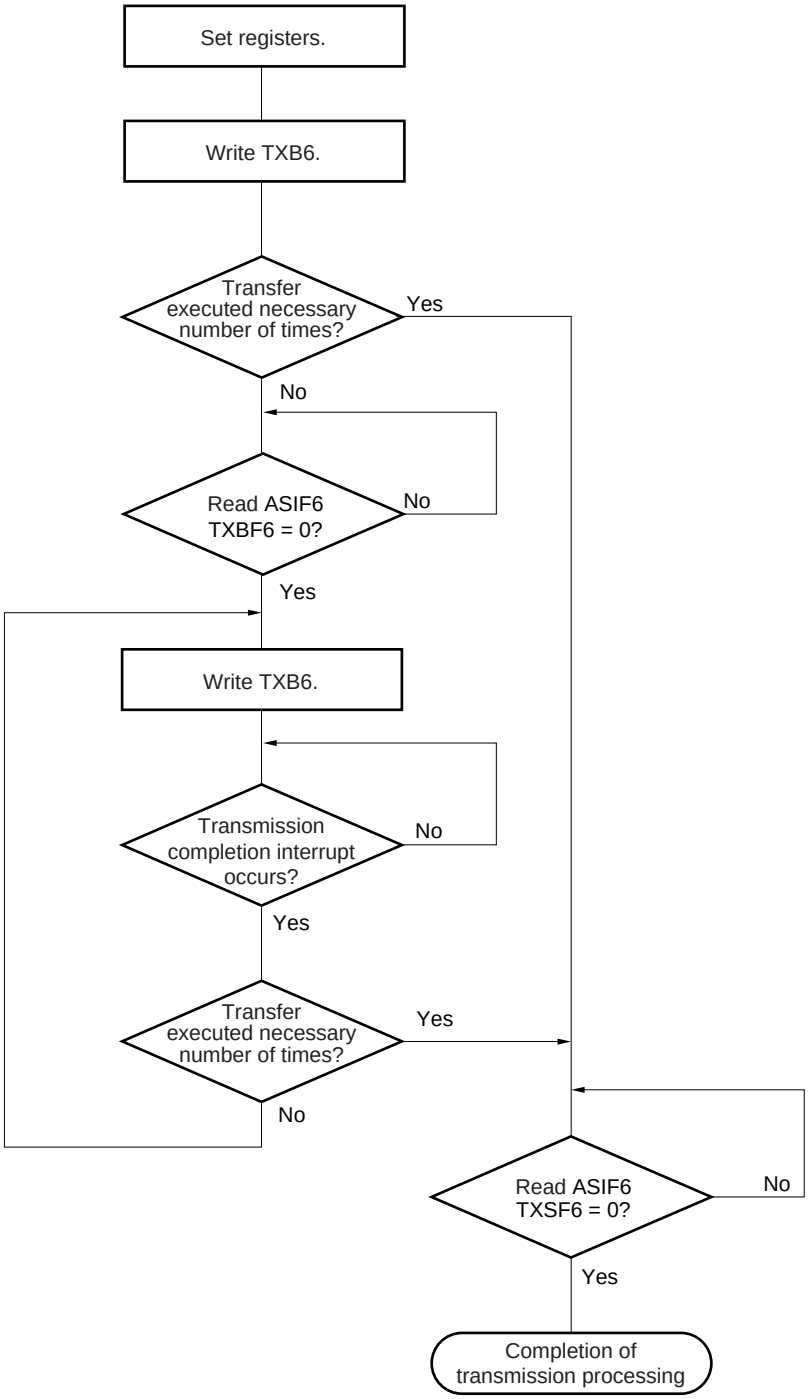
TXSF6	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

2. During continuous transmission, the next transmission may complete before execution of INTST6 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 TXSF6 flag.

Figure 13-17 shows an example of the continuous transmission processing flow.

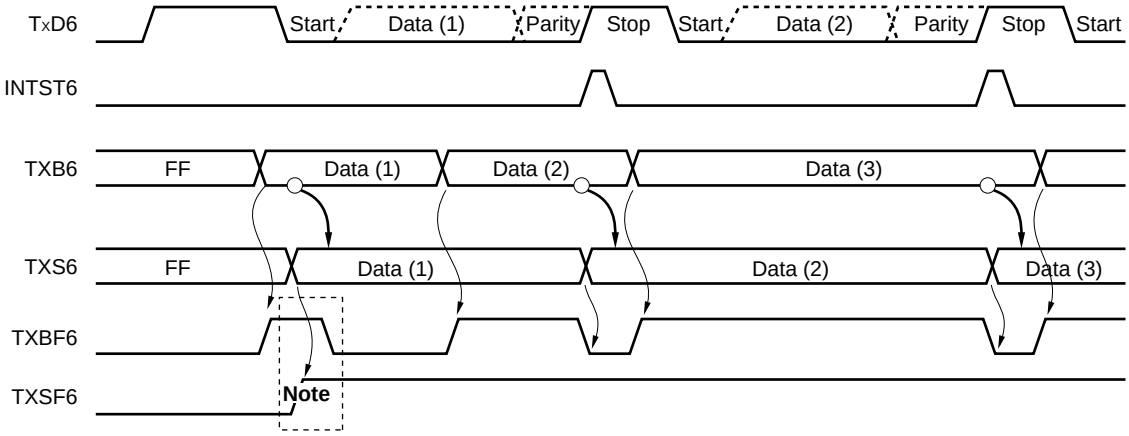
Figure 13-17. Example of Continuous Transmission Processing Flow



Remark TXB6: Transmit buffer register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6 (transmit buffer data flag)
TXSF6: Bit 0 of ASIF6 (transmit shift register data flag)

Figure 13-18 shows the timing of starting continuous transmission, and Figure 13-19 shows the timing of ending continuous transmission.

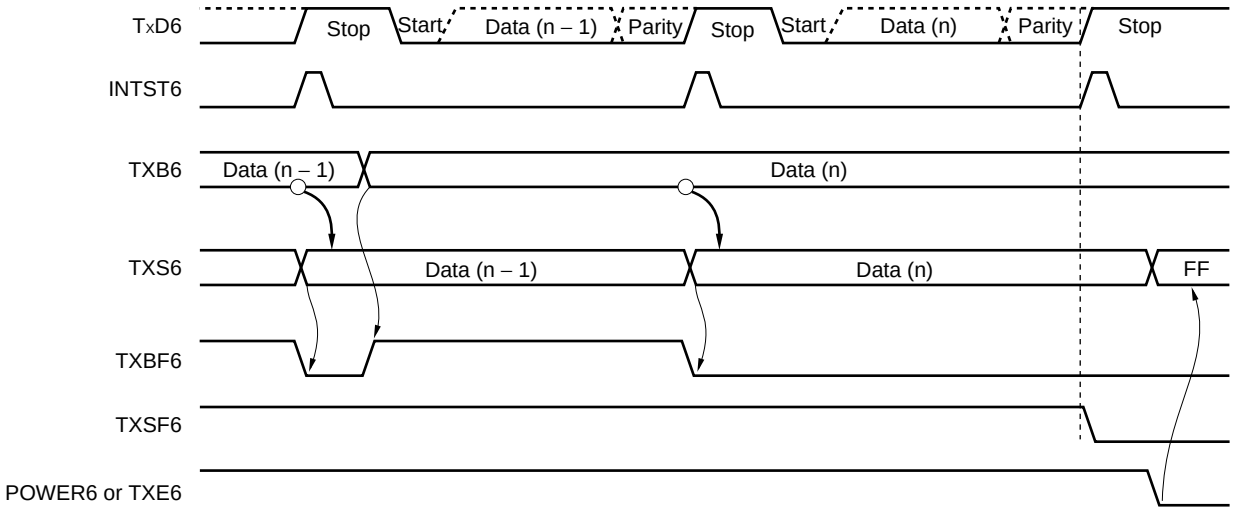
Figure 13-18. Timing of Starting Continuous Transmission



Note When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

Remark TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: Transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6

Figure 13-19. Timing of Ending Continuous Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: Transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6
POWER6: Bit 7 of asynchronous serial interface operation mode register (ASIM6)
TXE6: Bit 6 of asynchronous serial interface operation mode register (ASIM6)

(e) Normal reception

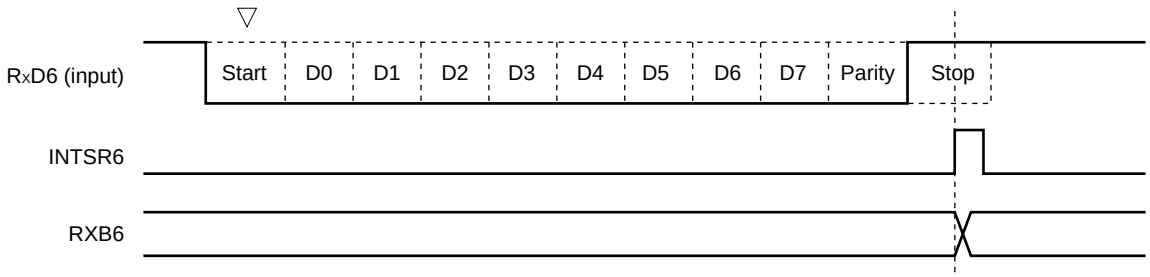
Reception is enabled and the RxD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6 pin input is detected. When the set value of baud rate generator control register 6 (BRGC6) has been counted, the RxD6 pin input is sampled again (in Figur▽14-20). If the RxD6 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 6 (RXS6) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6) is generated and the data of RXS6 is written to receive buffer register 6 (RXB6). If an overrun error (OVE6) occurs, however, the receive data is not written to RXB6.

Even if a parity error (PE6) occurs while reception is in progress, reception continues to the reception position of the stop bit, and a reception error interrupt (INTSR6/INTSRE6) is generated on completion of reception.

Figure 13-20. Reception Completion Interrupt Request Timing



Cautions 1. If a reception error occurs, read ASIS6 and then RXB6 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 6 (ASIS6) before reading RXB6.

(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 6 (ASIS6) is set as a result of data reception, a reception error interrupt request (INTSR6/INTSRE6) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6 in the reception error interrupt (INTSR6/INTSRE6) servicing (refer to **Figure 13-6**).

The contents of ASIS6 are cleared to 0 when ASIS6 is read.

Table 13-3. Cause of Reception Error

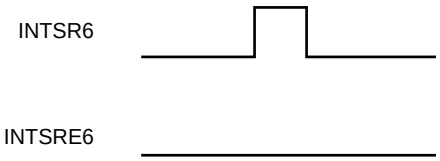
Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).

The reception error interrupt can be separated into reception completion interrupt (INTSR6) and error interrupt (INTSRE6) by clearing bit 0 (ISRM6) of asynchronous serial interface operation mode register 6 (ASIM6) to 0.

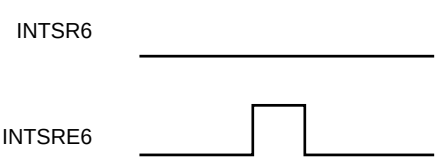
Figure 13-21. Reception Error Interrupt

1. If ISRM6 is cleared to 0 (reception completion interrupt (INTSR6) and error interrupt (INTSRE6) are separated)

(a) No error during reception

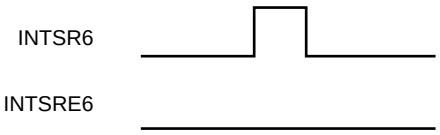


(b) Error during reception

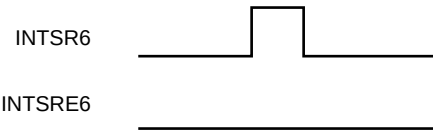


2. If ISRM6 is set to 1 (error interrupt is included in INTSR6)

(a) No error during reception



(b) Error during reception



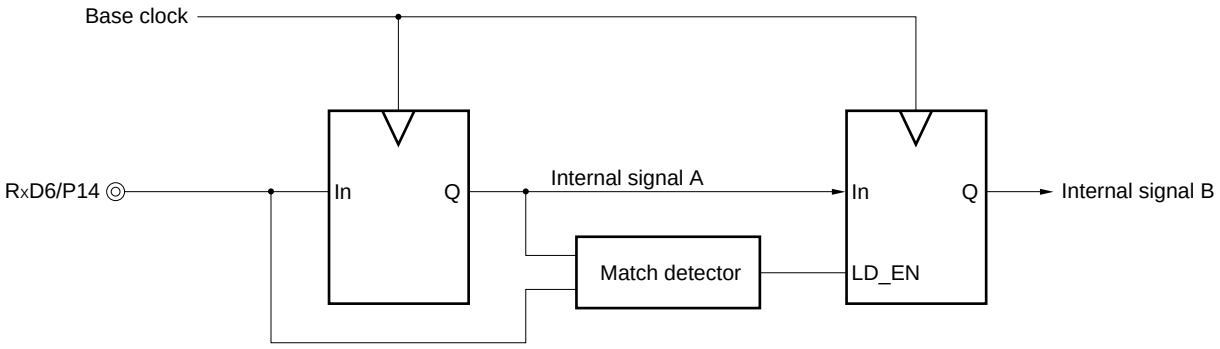
(g) Noise filter of receive data

The RxD6 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 13-22, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 13-22. Noise Filter Circuit



(h) SBF transmission

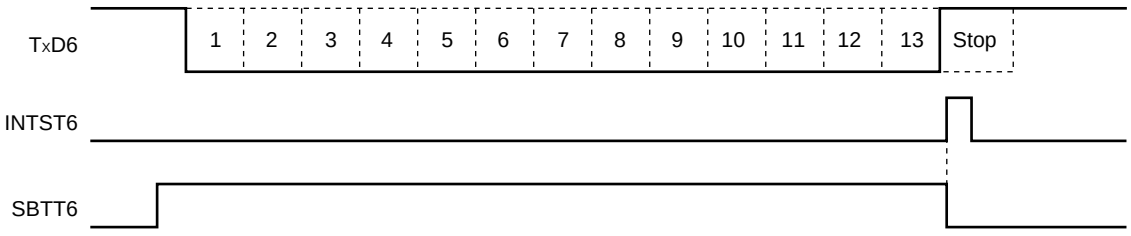
When the device is use in LIN communication operation, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, refer to **Figure 13-1 LIN Transmission Operation**.

When bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1, the TxD6 pin outputs high level. Next, when bit 6 (TXE6) of ASIM6 is set to 1, the transmission enabled status is entered, and SBF transmission is started by setting bit 5 (SBTT6) of asynchronous serial interface control register 6 (ASICL6) to 1.

Thereafter, a low level of bits 13 to 20 (set by bits 4 to 2 (SBL62 to SBL60) of ASICL6) is output. Following the end of SBF transmission, the transmission completion interrupt request (INTST6) is generated and SBTT6 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 6 (TXB6), or until SBTT6 is set to 1.

Figure 13-23. SBF Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Transmission completion interrupt request
SBTT6: Bit 5 of asynchronous serial interface control register 6 (ASICL6)

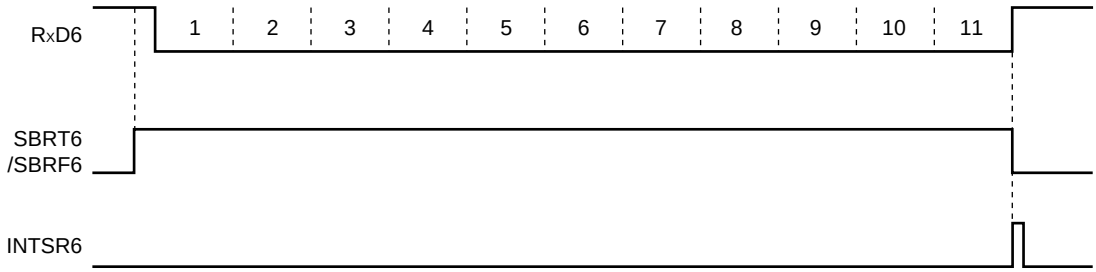
(i) SBF reception

When the device is used in LIN communication operation, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, refer to **Figure 13-2 LIN Reception Operation**. Reception is enabled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1. SBF reception is enabled when bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1. In the SBF reception enabled status, the RxD6 pin is sampled and the start bit is detected in the same manner as the normal reception enable status.

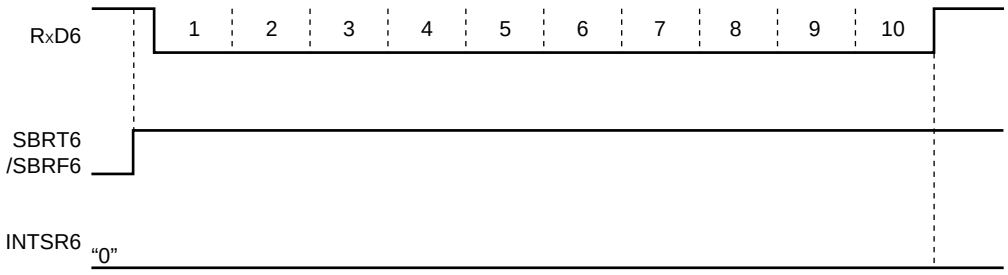
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal processing. At this time, the SBRF6 and SBRT6 bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6, PE6, and FE6 (bits 0 to 2 of asynchronous serial interface reception error status register 6 (ASIS6)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6 (RXS6) and receive buffer register 6 (RXB6) 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 SBRF6 and SBRT6 bits are not cleared.

Figure 13-24. 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)



Remark RxD6: RxD6 pin (input)
SBRT6: Bit 6 of asynchronous serial interface control register 6 (ASICL6)
SBRF6: Bit 7 of ASICL6
INTSR6: Reception completion interrupt request

13.4.3 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 UART6.

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 (TPS63 to TPS60) of clock selection register 6 (CKSR6) is supplied to each module when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is 1. This clock is called the base clock and its frequency is called f_{XCLK6}. The base clock is fixed to low level when POWER6 = 0.

- Transmission counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 6 (TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.

  It starts counting when POWER6 = 1 and TXE6 = 1.

  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6 (TXB6).

  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 POWER6 or TXE6 is cleared to 0.

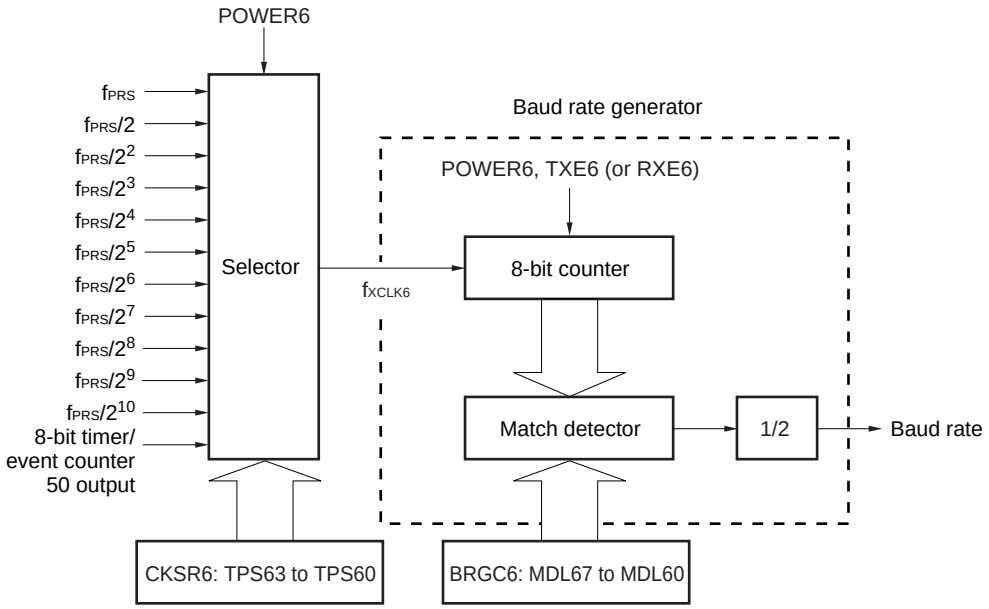
- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 5 (RXE6) of asynchronous serial interface operation mode register 6 (ASIM6) 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 13-25. Configuration of Baud Rate Generator



Remark POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
CKSR6: Clock selection register 6
BRGC6: Baud rate generator control register 6

(2) Generation of serial clock

A serial clock to be generated can be specified by using clock selection register 6 (CKSR6) and baud rate generator control register 6 (BRGC6).

The clock to be input to the 8-bit counter can be set by bits 3 to 0 (TPS63 to TPS60) of CKSR6 and the division value ($f_{XCLK6}/4$ to $f_{XCLK6}/255$) of the 8-bit counter can be set by bits 7 to 0 (MDL67 to MDL60) of BRGC6.

13.4.4 Calculation of baud rate

(1) Baud rate calculation expression

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK6}}{2 \times k}$ [bps]

f_{XCLK6}: Frequency of base clock selected by TPS63 to TPS60 bits of CKSR6 register

k: Value set by MDL67 to MDL60 bits of BRGC6 register (k = 4, 5, 6, ..., 255)

Table 13-4. Set Value of TPS63 to TPS60

TPS63	TPS62	TPS61	TPS60	Base Clock (f_{XCLK6}) Selection[Note]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz (when using PLL)
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
Other than above				Setting prohibited				

Note If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

(2) Error of baud rate

The baud rate error can be calculated by the following expression.

- Error (%) = $\left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1\right) \times 100$ [%]

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 MDL67 to MDL60 bits of BRGC6 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 [%]

(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 (when using PLL)			
	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	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	116279	–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

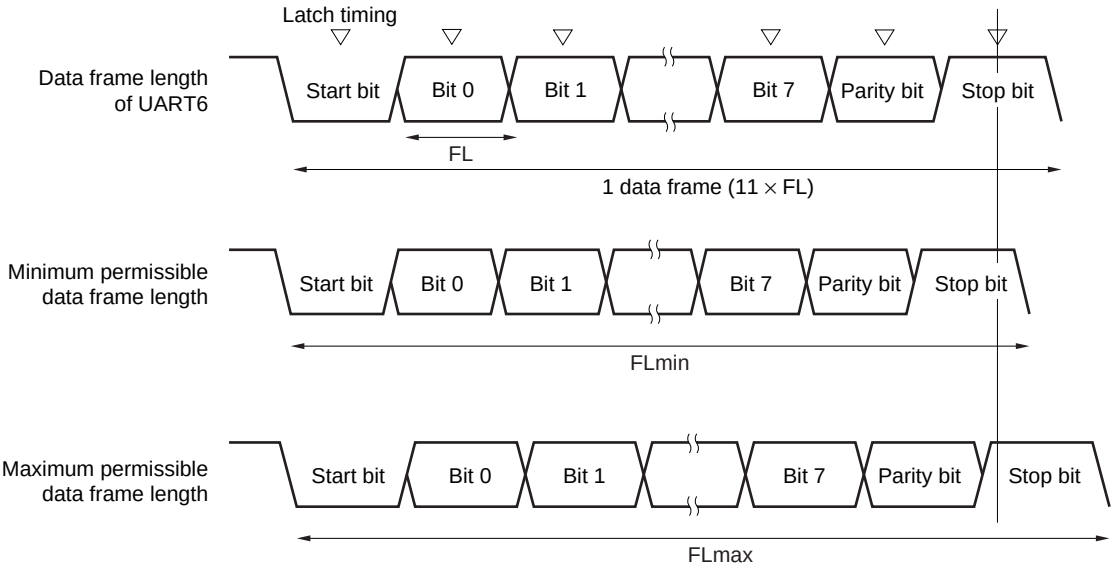
Remark TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock (f_{XCLK6}))
k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) (k = 4, 5, 6, ..., 255)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(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 13-26. Permissible Baud Rate Range During Reception



As shown in Figure 13-26, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6 (BRGC6) 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 UART6
k: Set value of BRGC6
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 UART6 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
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%

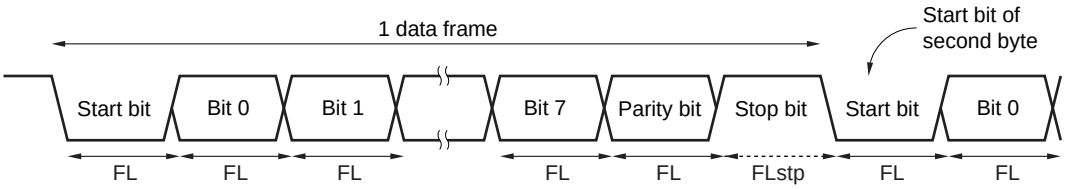
Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC6

(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 13-27. 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_{XCLK6}, the following expression is satisfied.

$$FLstp = FL + 2/f_{XCLK6}$$

Therefore, the data frame length during continuous transmission is:

$$\text{Data frame length} = 11 \times FL + 2/f_{XCLK6}$$

CHAPTER 14 SERIAL INTERFACE IICA

14.1 Functions of Serial Interface IICA

Serial interface IICA is mounted onto all 78K0/Fx2-L microcontroller products.
Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLA0) line and a serial data bus (SDAA0) 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 SCLA0 and SDAA0 pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICA0) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP bit of IICA control register 1 (IICACTL1).

Figure 14-1 shows a block diagram of serial interface IICA.

Figure 14-1. Block Diagram of Serial Interface IICA

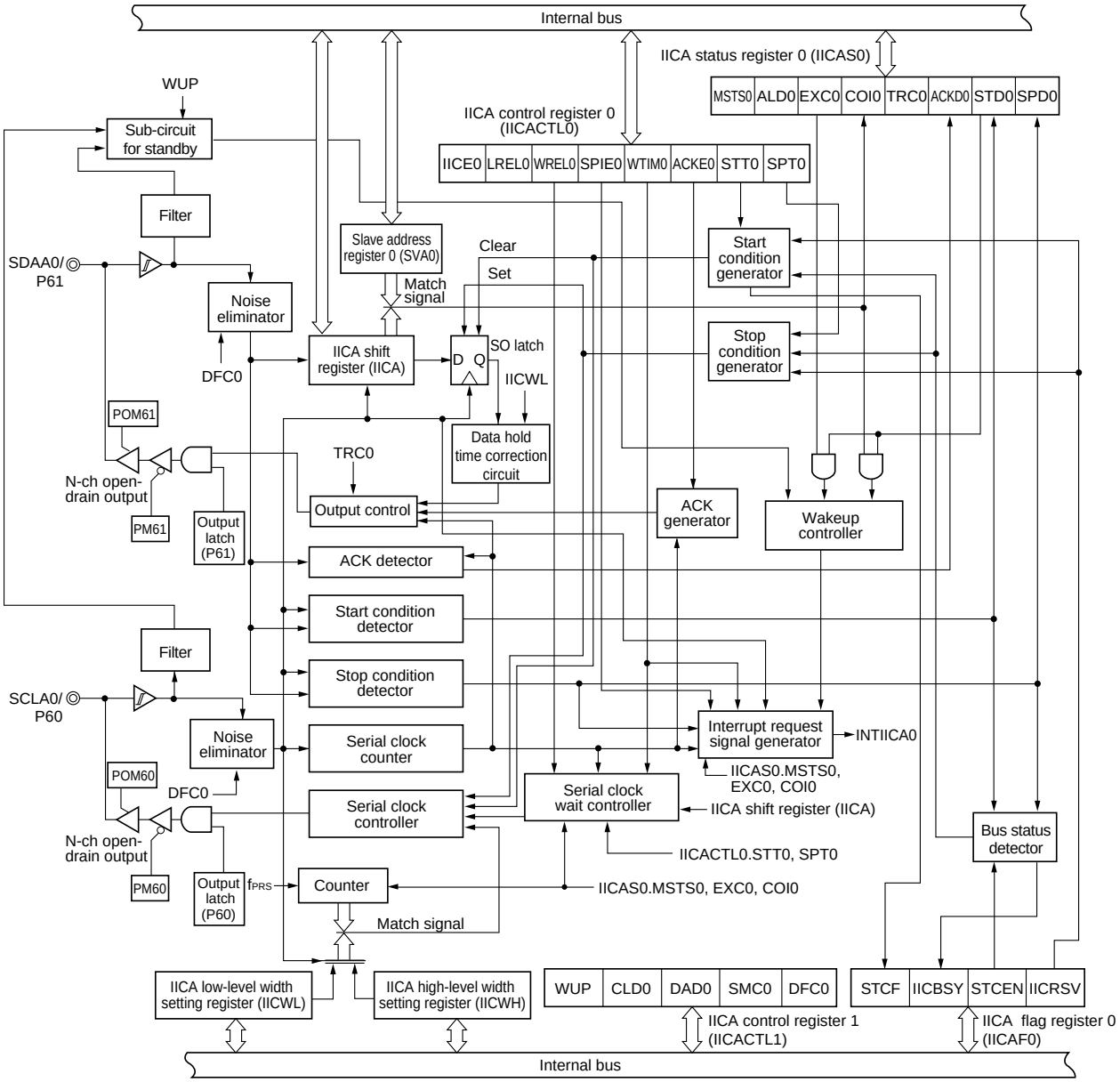
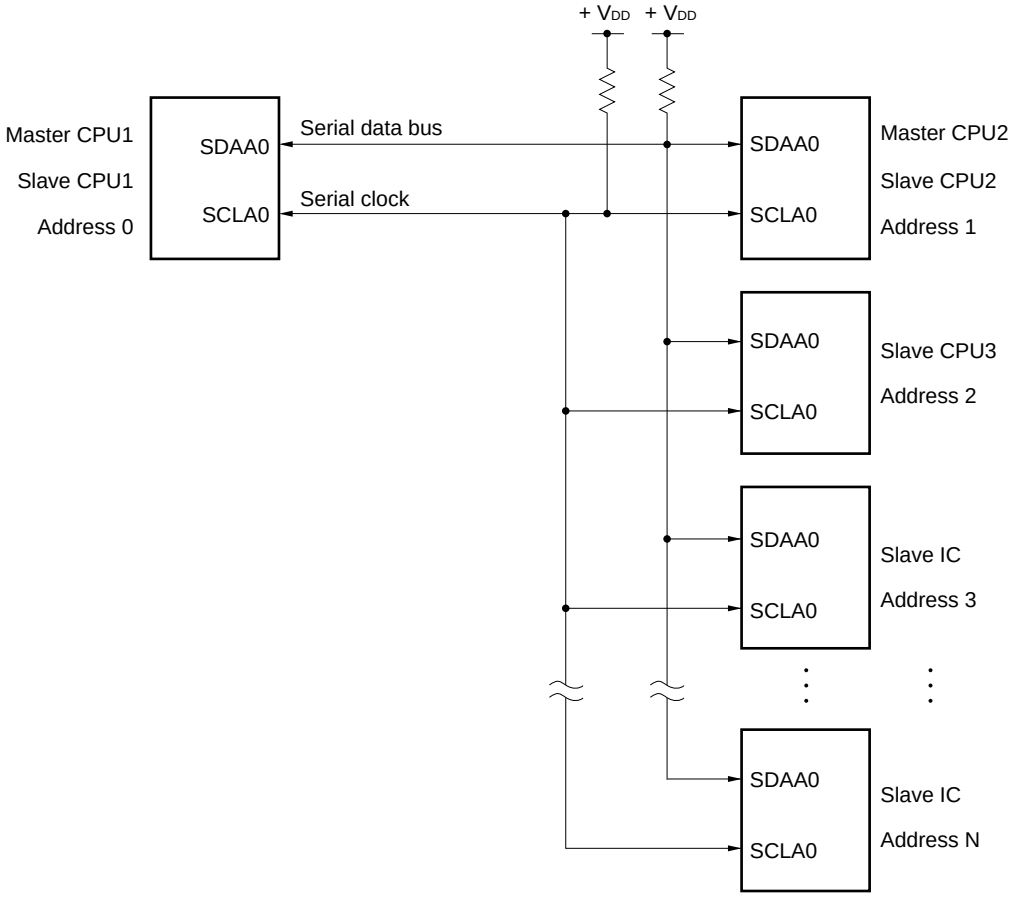


Figure 14-2 shows a serial bus configuration example.

Figure 14-2. Serial Bus Configuration Example Using I²C Bus



14.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 14-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register (IICA) Slave address register 0 (SVA0)
Control registers	IICA control register 0 (IICACTL0) IICA status register 0 (IICAS0) IICA flag register 0 (IICAF0) IICA control register 1 (IICACTL1) IICA low-level width setting register (IICWL) IICA high-level width setting register (IICWH) Port input mode register 6 (PIM6) Port output mode register 6 (POM6) Port mode register 6 (PM6) Port register 6 (P6)

(1) IICA shift register (IICA)

This register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. This register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to this register.

Cancel the wait state and start data transfer by writing data to this register during the wait period.

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

Reset signal generation clears IICA to 00H.

Figure 14-3. Format of IICA Shift Register (IICA)

Address: FFA5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA								

Cautions 1. Do not write data to the IICA register during data transfer.

2. Write or read the IICA register only during the wait period. Accessing the IICA register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, the IICA register can be written only once after the communication trigger bit (STT0) is set to 1.

3. When communication is reserved, write data to the IICA register after the interrupt triggered by a stop condition is detected.

(2) Slave address register 0 (SVA0)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears SVA0 to 00H.

Figure 14-4. Format of Slave Address Register 0 (SVA0)

Address: FFA6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0								0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA0) when the address received by this register matches the address value set to the slave address register 0 (SVA0) 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 (INTIICA0).

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 WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IICA control register 0 (IICACTL0)
SPIE0 bit: Bit 4 of IICA control register 0 (IICACTL0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLA0 pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT0 bit: Bit 1 of IICA control register 0 (IICACTL0)
SPT0 bit: Bit 0 of IICA control register 0 (IICACTL0)
IICRSV bit: Bit 0 of IICA flag register 0 (IICAF0)
IICBSY bit: Bit 6 of IICA flag register 0 (IICAF0)
STCF bit: Bit 7 of IICA flag register 0 (IICAF0)
STCEN bit: Bit 1 of IICA flag register 0 (IICAF0)

14.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following ten registers.

- IICA control register 0 (IICACTL0)
- IICA status register 0 (IICAS0)
- IICA flag register (IICAF0)
- IICA control register 1 (IICACTL1)
- IICA low-level width setting register (IICWL)
- IICA high-level width setting register (IICWH)
- Port input mode register 6 (PIM6)
- Port output mode register 6 (POM6)
- Port mode register 6 (PM6)
- Port register 6 (P6)

(1) IICA control register 0 (IICACTL0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

IICACTL0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 bit = 0 or during the wait period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 14-5. Format of IICA Control Register 0 (IICACTL0) (1/4)

Address: FFA7H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICACTL0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset the IICA status register 0 (IICAS0)Note 1. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLA0 and SDLA0 lines are at high level.	
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0Notes 2, 3	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLA0 and SDAA0 lines are set to high impedance. The following flags of IICA control register 0 (IICACTL0) and IICA status register 0 (IICAS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
The standby mode following exit from communications remains in effect until the following 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 (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0Notes 2, 3	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WREL0 is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDAA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICAS0 register, the STCF and IICBSY bits of the IICAF0 register, and the CLD0 and DAD0 bits of the IICACTL1 register are reset.

2. The signals of these bits are invalid while the IICE0 bit is 0.

3. When the LREL0 and WREL0 bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE0 = 1) when the SCLA0 line is high level, the SDAA0 line is low level, and the digital filter is turned on (DFC0 of the IICACTL1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE0 = 1).

Figure 14-5. Format of IICA Control Register 0 (IICACTL0) (2/4)

SPIE0[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUP0 of the IICA control register 1 (IICACTL1) is 1, no stop condition interrupt will be generated even if SPIE0 = 1.	
Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[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 (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0 Notes 1, 2	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAA0 line is set to low level.
Condition for clearing (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 14-5. Format of IICA Control Register 0 (IICACTL0) (3/4)

STT0[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV = 1) Even if this bit is set (1), the STT0 bit is cleared and the STT0 clear flag (STCF) is set (1). No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE0 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 stop condition trigger (SPT0).
- Setting the STT0 bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Cleared by setting STT0 bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note The signal of this bit is invalid while IICE0 is 0.

Remarks 1. Bit 1 (STT0) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IICA flag register 0 (IICAF0)
STCF: Bit 7 of IICA flag register 0 (IICAF0)

Figure 14-5. Format of IICA Control Register 0 (IICACTL0) (4/4)

SPT0	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE0 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 start condition trigger (STT0).
- The SPT0 bit can be set to 1 only when in master mode.
- When the WTIM0 bit has been cleared to 0, if the SPT0 bit 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. The WTIM0 bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting the SPT0 bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Caution When bit 3 (TRC0) of the IICA status register 0 (IICAS0) is set to 1 (transmission status), bit 5 (WREL0) of the IICACTL0 register is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

(2) IICA status register 0 (IICAS0)

This register indicates the status of I^2C.

IICAS0 is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICAS0 register while the address match wakeup function is enabled (WUP = 1) in STOP mode is prohibited. When the WUP bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 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 (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICAS0 register after the interrupt has been detected.

Remark STT0: Bit 1 of IICA control register 0 (IICACTL0)
WUP: Bit 7 of IICA control register 1 (IICACTL1)

Figure 14-6. Format of IICA Status Register 0 (IICAS0) (1/3)

Address: FFAAH After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICAS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	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”. The MSTS0 bit is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after the IICAS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the ALD0 bit of the IICAS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

Figure 14-6. Format of IICA Status Register 0 (IICAS0) (2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 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).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDAA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1[Note] (wait cancel) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS0, EXC0, COI0 = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRC0) of the IICA status register 0 (IICAS0) is set to 1 (transmission status), bit 5 (WREL0) of the IICA control register 0 (IICACTL0) is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while TRC0 bit is 1 (transmission status) by writing to the IICA shift register.

Remark LREL0: Bit 6 of IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

Figure 14-6. Format of IICA Status Register 0 (IICAS0) (3/3)

ACKD0	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• After the SDAA0 line is set to low level at the rising edge of SCLA0's ninth clock

STD0	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 (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	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 (SPD0 = 0)	Condition for setting (SPD0 = 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 IICE0 changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL0: Bit 6 of IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

(3) IICA flag register 0 (IICAF0)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

This register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT0 clear flag (STCF) and I^2C bus status flag (IICBSY) are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

The STCEN bit can be used to set the initial value of the IICBSY bit.

The IICRSV and STCEN bits can be written only when the operation of I^2C is disabled (bit 7 (IICE0) of the IICA control register 0 (IICACTL0) = 0). When operation is enabled, the IICAF0 register can be read.

Reset signal generation clears this register to 00H.

Figure 14-7. Format of IICA Flag Register 0 (IICAF0)

Address: FFA9H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICAF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT0 flag

Condition for clearing (STCF = 0)	Condition for setting (STCF = 1)
• Cleared by STT0 = 1 • When IICE0 = 0 (operation stop) • Reset	• Generating start condition unsuccessful and STT0 bit cleared to 0 when communication reservation is disabled (IICRSV = 1).

IICBSY	I^2C bus status flag
0	Bus release status (communication initial status when STCEN = 1)
1	Bus communication status (communication initial status when STCEN = 0)

Condition for clearing (IICBSY = 0)	Condition for setting (IICBSY = 1)
• Detection of stop condition • When IICE0 = 0 (operation stop) • Reset	• Detection of start condition • Setting of IICE0 when STCEN = 0

STCEN	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN = 0)	Condition for setting (STCEN = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV = 0)	Condition for setting (IICRSV = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to STCEN bit only when the operation is stopped (IICE0 = 0).

2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV bit only when the operation is stopped (IICE0 = 0).

Remark STT0: Bit 1 of IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

(4) IICA control register 1 (IICACTL1)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLA0 and SDAA0 pins.

IICACTL1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICACTL1 register, except the WUP bit, while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 0 (IICACTL0) is 0).

Reset signal generation clears this register to 00H.

Figure 14-8. Format of IICA Control Register 1 (IICACTL1) (1/2)

Address: FFA8H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICACTL1	WUP	0	CLD0	DAD0	SMC0	DFC0	0	0

WUP	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.
To shift to STOP mode when WUP = 1, execute the STOP instruction at least three clocks after setting (1) WUP bit (see **Figure 14-23 Flow When Setting WUP = 1**). Clear (0) the WUP bit after the address has matched or an extension code has been received. The subsequent communication can be entered by clearing (0) the WUP bit (The wait must be released and transmit data must be written after the WUP bit has been cleared (0).). The interrupt timing when the address has matched or when an extension code has been received, while WUP = 1, is identical to the interrupt timing when WUP = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1.	
Condition for clearing (WUP = 0)	Condition for setting (WUP = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when MSTS0, EXC0, and COI0 are “0”, and STD0 also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of IICAS0 must be checked and WUP must be set during the period shown below.

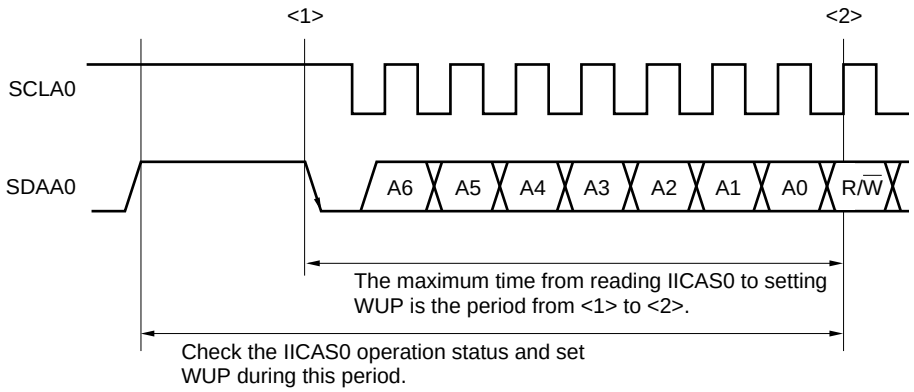


Figure 14-8. Format of IICA Control Register 1 (IICACTL1) (2/2)

CLD0	Detection of SCLA0 pin level (valid only when IICE0 = 1)
0	The SCLA0 pin was detected at low level.
1	The SCLA0 pin was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCLA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCLA0 pin is at high level

DAD0	Detection of SDAA0 pin level (valid only when IICE0 = 1)
0	The SDAA0 pin was detected at low level.
1	The SDAA0 pin was detected at high level.

Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDAA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDAA0 pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC0	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 DFC0 bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

Remark IICE0: Bit 7 of IICA control register 0 (IICACTL0)

(5) IICA low-level width setting register (IICWL)

This register is used to set the low-level width of the SCLA0 pin signal that is output by serial interface IICA being in master mode.

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

Set this register while operation of I^2C is disabled (bit 7 (IICE0) of the IICA control register 0 (IICACTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 14-9. Format of IICA Low-Level Width Setting Register (IICWL)

Address: FFADH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL								

(6) IICA high-level width setting register (IICWH)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA being in master mode.

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

Set this register while operation of I^2C is disabled (bit 7 (IICE0) of the IICA control register 0 (IICACTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 14-10. Format of IICA High-Level Width Setting Register (IICWH)

Address: FFAEH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH								

Remark For how to set the transfer clock by using the IICWL and IICWH registers, see **15.4.2 Setting transfer clock by using IICWL and IICWH registers**.

(7) Port input mode register 6 (PIM6)

This register sets the input buffer of P60 and P61 in 1-bit units. When using an input compliant with the SMBus specifications in I^2C communication, set PIM60 and PIM61 to 1.

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

Reset signal generation clears this register to 00H.

Figure 14-11. Format of Port Input Mode Register 6 (PIM6)

Address: FF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM6	0	0	0	0	0	0	PIM61	PIM60

PIM6n	P6n pin input buffer selection (n = 0, 1)
0	Normal input (Schmitt) buffer
1	SMBus input buffer

(8) Port output mode register 6 (POM6)

This register sets the output mode of P60 and P61 in 1-bit units. During I^2C communication, set POM60 and POM61 to 1.

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

Reset signal generation clears this register to 00H.

Figure 14-12. Format of Port Output Mode Register 6 (POM6)

Address: FF2AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM6	0	0	0	0	0	0	POM61	POM60

POM6n	P6n pin output mode selection (n = 0, 1)
0	Normal output (CMOS output) mode
1	N-ch open drain output (V_{DD} tolerance) mode

(9) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCLA0/TxD6 pin as clock I/O and the P61/SDAA0/RxD6 pin as serial data I/O, clear PM60 and PM61 to 0, and set the output latches of P60 and P61 to 1.

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

Reset signal generation sets this register to FFH.

Figure 14-13. Format of Port Mode Register 6 (PM6)

Address: FF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 I^2C Bus Mode Functions

14.4.1 Pin configuration

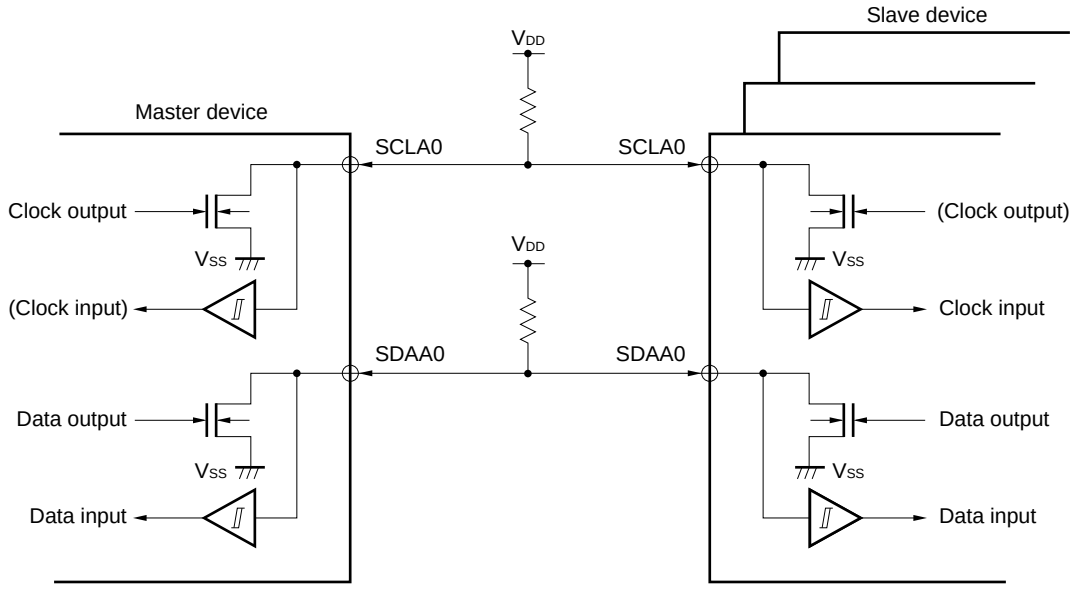
The serial clock pin (SCLA0) and serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0 .... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDAA0 .... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave 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 14-14. Pin Configuration Diagram



14.4.2 Setting transfer clock by using IICWL and IICWH registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{PRS}}{IICWL + IICWH + f_{PRS}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL and IICWH registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL = \frac{0.52}{\text{Transfer clock}} \times f_{PRS}$$
$$IICWH = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{PRS}$$

- When the normal mode

$$IICWL = \frac{0.47}{\text{Transfer clock}} \times f_{PRS}$$
$$IICWH = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{PRS}$$

(2) Setting IICWL and IICWH on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL = 1.3\ \mu s \times f_{PRS}$$
$$IICWH = (1.2\ \mu s - t_R - t_F) \times f_{PRS}$$

- When the normal mode

$$IICWL = 4.7\ \mu s \times f_{PRS}$$
$$IICWH = (5.3\ \mu s - t_R - t_F) \times f_{PRS}$$

Caution Note the minimum f_{PRS} operation frequency when setting the transfer clock. The minimum f_{PRS} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{PRS} = 3.5 MHz (min.)

Normal mode: f_{PRS} = 1 MHz (min.)

Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register

t_F: SDAA0 and SCLA0 signal falling times
(refer to **CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)** and **CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**)

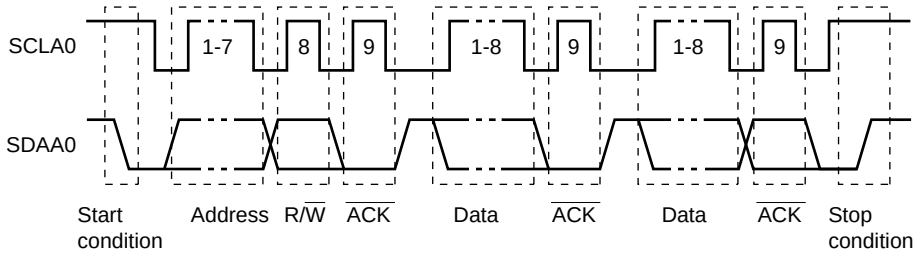
t_R: SDAA0 and SCLA0 signal rising times
(refer to **CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)** and **CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**)

f_{PRS}: Peripheral hardware clock frequency

14.5 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 14-15 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 14-15. I^2C Bus Serial Data Transfer Timing



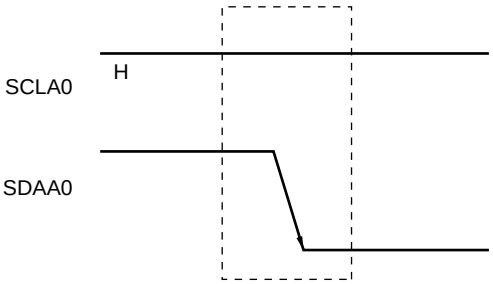
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLA0) is continuously output by the master device. However, in the slave device, the SCLA0's low level period can be extended and a wait can be inserted.

14.5.1 Start conditions

A start condition is met when the SCLA0 pin is at high level and the SDAA0 pin changes from high level to low level. The start conditions for the SCLA0 pin and SDAA0 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 14-16. Start Conditions



A start condition is output when bit 1 (STT0) of IICA control register 0 (IICACTL0) is set (1) after a stop condition has been detected (SPD0: Bit 0 of the IICA status register 0 (IICAS0) = 1). When a start condition is detected, bit 1 (STD0) of the IICAS0 register is set (1).

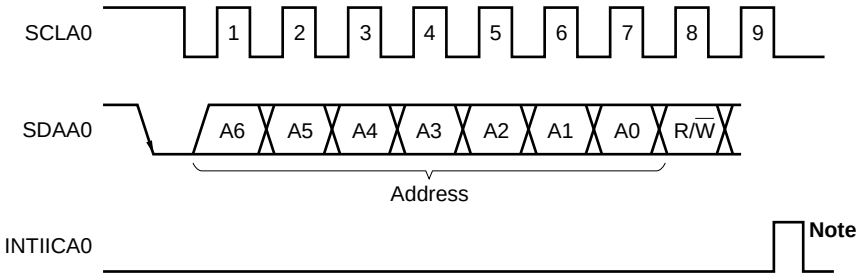
14.5.2 Addresses

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 0 (SVA0). If the address data matches the SVA0 register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 14-17. Address



Note INTIICA0 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 **14.5.3 Transfer direction specification** are written to the IICA shift register (IICA). The received addresses are written to IICA register .

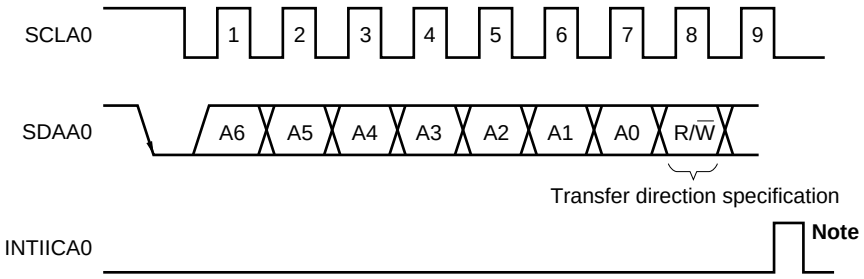
The slave address is assigned to the higher 7 bits of IICA register .

14.5.3 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 14-18. Transfer Direction Specification



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

14.5.4 Acknowledge ($\overline{ACK}$)

$\overline{ACK}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{ACK}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{ACK}$ after transmitting 8-bit data. When $\overline{ACK}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{ACK}$ has been detected can be checked by using bit 2 (ACKD0) of the IICA status register 0 (IICAS0).

When the master receives the last data item, it does not return $\overline{ACK}$ and instead generates a stop condition. If a slave does not return $\overline{ACK}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{ACK}$ is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

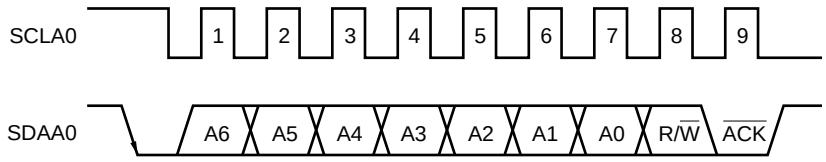
To generate $\overline{ACK}$, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE0) of IICA control register 0 (IICACTL0) to 1. Bit 3 (TRC0) of the IICAS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set ACKE0 bit to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing ACKE0 bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear ACKE0 bit to 0 so that $\overline{ACK}$ is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 14-19. $\overline{ACK}$



When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of ACKE0 bit . When an address other than that of the local address is received, $\overline{ACK}$ is not generated (NACK).

When an extension code is received, $\overline{ACK}$ is generated if ACKE0 bit is set to 1 in advance.

How $\overline{ACK}$ is generated when data is received differs as follows depending on the setting of the wait timing.

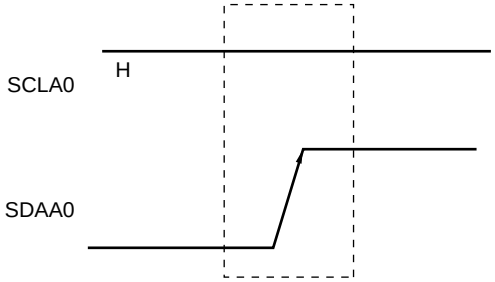
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICACTL0 register = 0):
  By setting ACKE0 bit to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICACTL0 register = 1):
  $\overline{ACK}$ is generated by setting ACKE0 bit to 1 in advance.

14.5.5 Stop condition

When the SCLA0 pin is at high level, changing the SDAA0 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 14-20. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IICA control register 0 (IICACTL0) is set to 1. When the stop condition is detected, bit 0 (SPD0) of the IICA status register 0 (IICAS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of IICACTL0 register is set to 1.

14.5.6 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 SCLA0 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 14-21. 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 ACKE0 = 1)

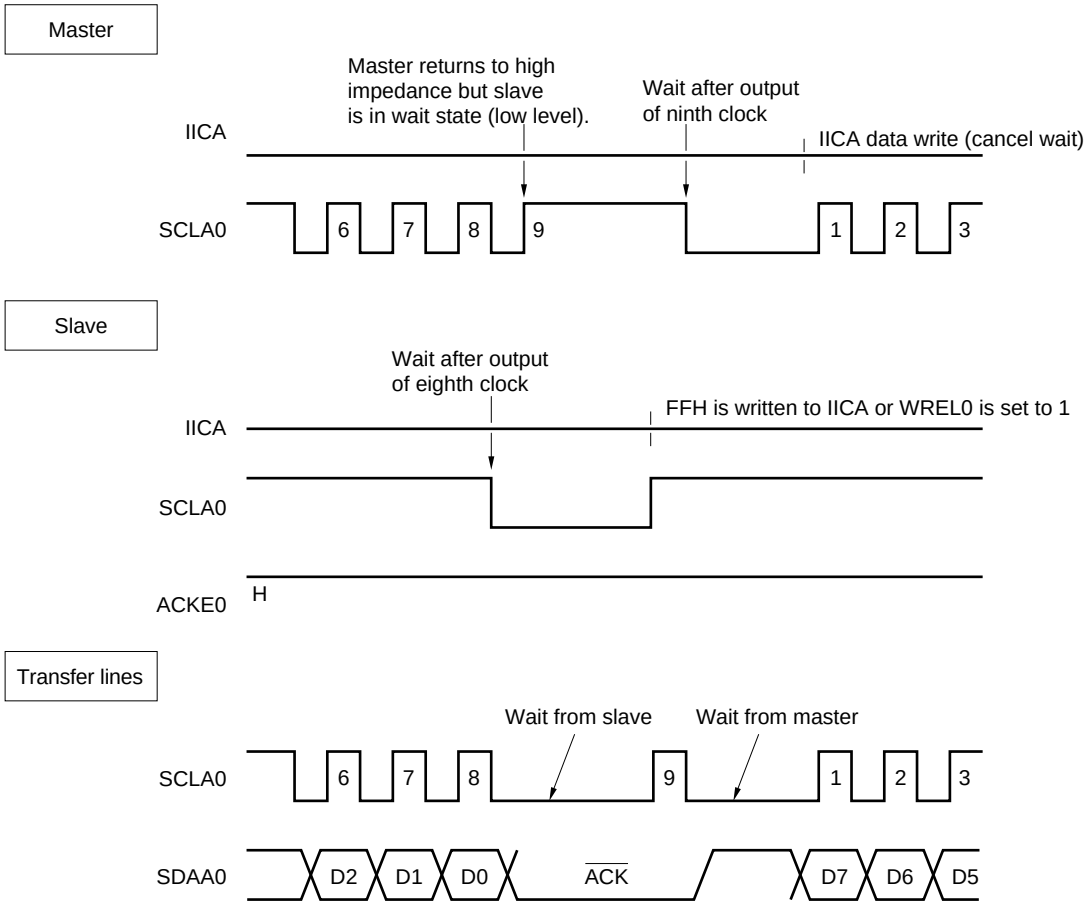
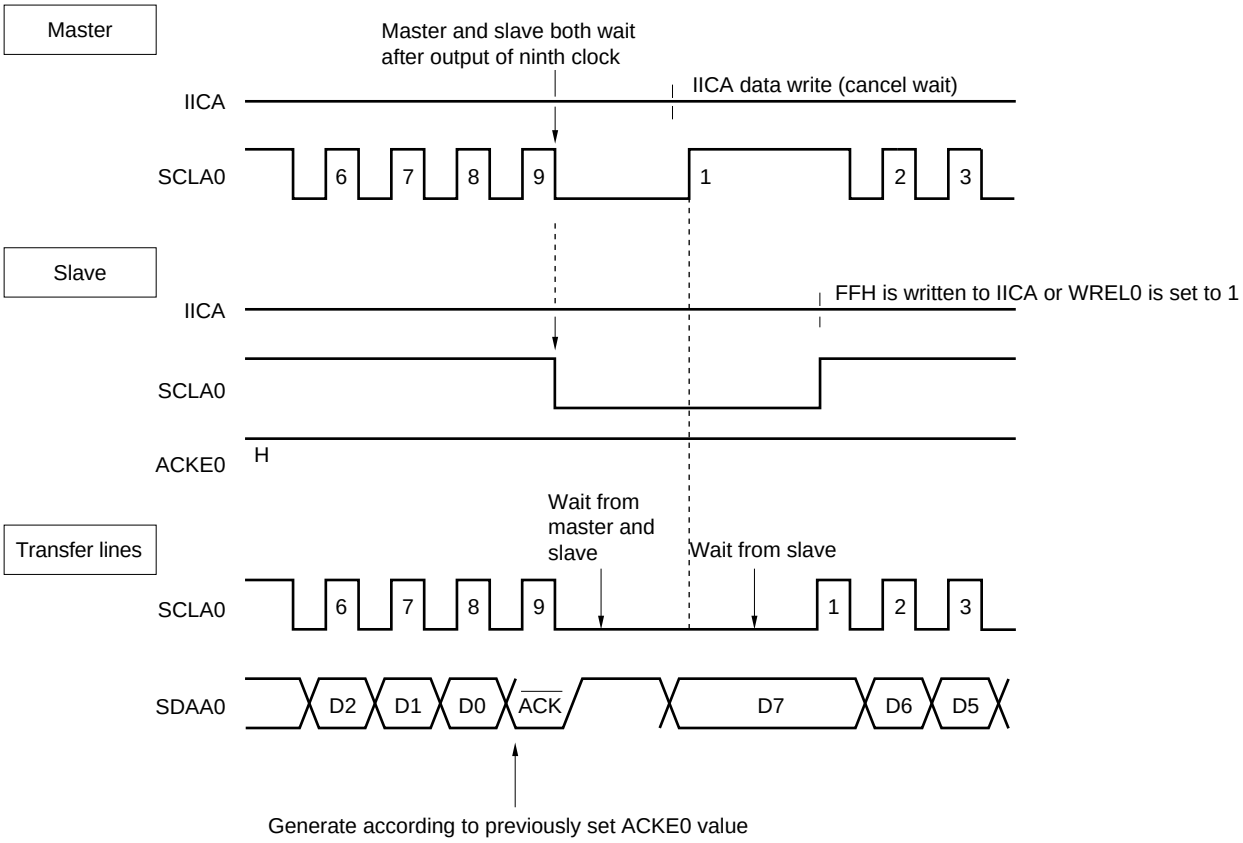


Figure 14-21. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IICA control register 0 (IICACTL0)
WREL0: Bit 5 of IICA control register 0 (IICACTL0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 0 (IICACTL0).

Normally, the receiving side cancels the wait state when bit 5 (WREL0) of the IICACTL0 register is set to 1 or when FFH is written to the IICA shift register (IICA), and the transmitting side cancels the wait state when data is written to the IICA register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of IICACTL0 register to 1
- By setting bit 0 (SPT0) of IICACTL0 register to 1

14.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL0) of IICA control register 0 (IICACTL0) (canceling wait)
- Setting bit 1 (STT0) of IICACTL0 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICACTL0 register (generating stop condition)Note

Note Master only

When the above wait canceling processing is executed, the I^2C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to IICA register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of IICA control register 0 (IICACTL0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of IICACTL0 register to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of IICACTL0 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IICA after canceling a wait state by setting WREL0 bit to 1, an incorrect value may be output to SDAA0 because the timing for changing the SDAA0 line conflicts with the timing for writing IICA register.

In addition to the above, communication is stopped if IICE0 bit 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 (LREL0) of IICACTL0 register, so that the wait state can be canceled.

Caution If a processing to cancel a wait state executed when WUP (bit 7 of IICA control register 1 (IICACTL1)) = 1, the wait state will not be canceled.

14.5.8 Interrupt request (INTIICA0) generation timing and wait control

The setting of bit 3 (WTIM0) of IICA control register 0 (IICACTL0) determines the timing by which INTIICA0 is generated and the corresponding wait control, as shown in Table 14-2.

Table 14-2. INTIICA0 Generation Timing and Wait Control

WTIM0	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 INTIICA0 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 0 (SVA0).
At this point, $\overline{ACK}$ is generated regardless of the value set to IICACTL0 register's bit 2 (ACKE0). For a slave device that has received an extension code, INTIICA0 occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA0 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 0 (SVA0) and extension code is not received, neither INTIICA0 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 WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL0) of IICA control register 0 (IICACTL0) (canceling wait)
- Setting bit 1 (STT0) of IICACTL0 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICACTL0 register (generating stop condition)Note

Note Master only.

When an 8-clock wait has been selected (WTIM0 = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 1).

14.5.9 Address match detection method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICA0) occurs when a local address has been set to the slave address register 0 (SVA0) and when the address set to SVA0 matches the slave address sent by the master device, or when an extension code has been received.

14.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDAA0) during data transmission is captured by the IICA shift register (IICA) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

14.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIICA0) is issued at the falling edge of the eighth clock. The local address stored in the slave address register 0 (SVA0) is not affected.

(2) If “11110××0” is set to the SVA0 register by a 10-bit address transfer and “11110××0” is transferred from the master device, the results are as follows. Note that INTIICA0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IICA status register 0 (IICAS0)
COI0: Bit 4 of IICA status register 0 (IICAS0)

(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 (LREL0) of the IICA control register 0 (IICACTL0) to 1 to set the standby mode for the next communication operation.

Table 14-3. 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)

Remark For extension codes other than the above, refer to THE I^2C-BUS SPECIFICATION published by NXP.

14.5.12 Arbitration

When several master devices simultaneously generate a start condition (when STT0 is set to 1 before STD0 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 (ALD0) in the IICA status register 0 (IICAS0) is set (1) via the timing by which the arbitration loss occurred, and the SCLA0 and SDAA0 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 ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, refer to **14.5.8 Interrupt request (INTIICA0) generation timing and wait control**.

Remark STD0: Bit 1 of IICA status register 0 (IICAS0)
STT0: Bit 1 of IICA control register 0 (IICACTL0)

Figure 14-22. Arbitration Timing Example

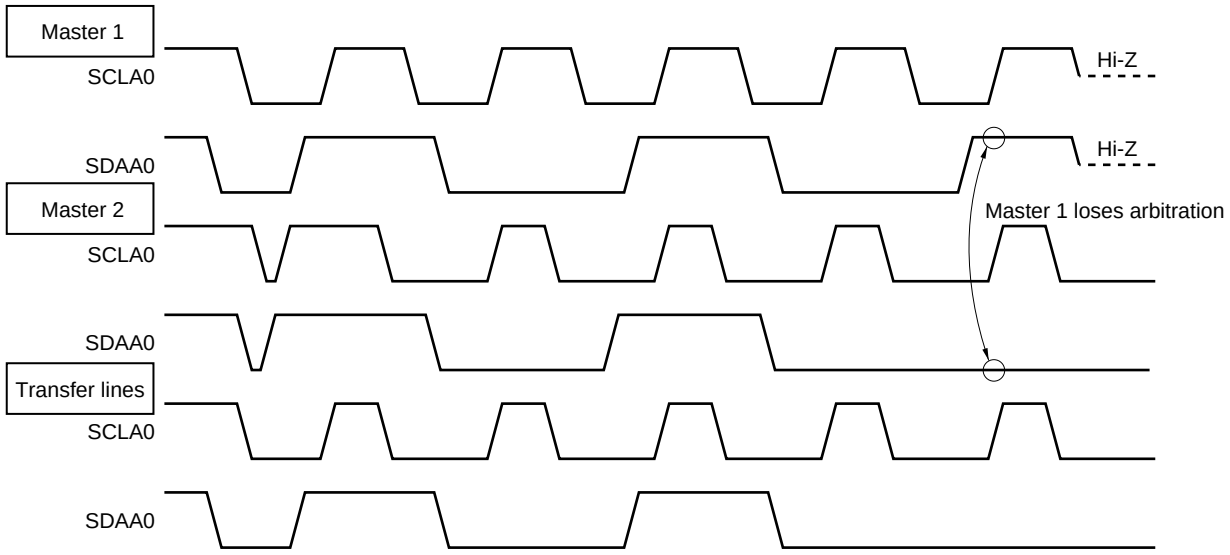


Table 14-4. Status During Arbitration and Interrupt Request Generation Timing

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 SPIE0 = 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 SPIE0 = 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 SCLA0 is at low level while attempting to generate a restart condition	

Notes 1. When the WTIM0 bit (bit 3 of the IICA control register 0 (IICACTL0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 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 SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 4 of IICA control register 0 (IICACTL0)

14.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, bit 4 (SPIE0) of IICA control register 0 (IICACTL0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

To use the wakeup function in the STOP mode, set WUP to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP bit after this interrupt has been generated.

Figure 14-23 shows the flow for setting WUP = 1 and Figure 14-24 shows the flow for setting WUP = 0 upon an address match.

Figure 14-23. Flow When Setting WUP = 1

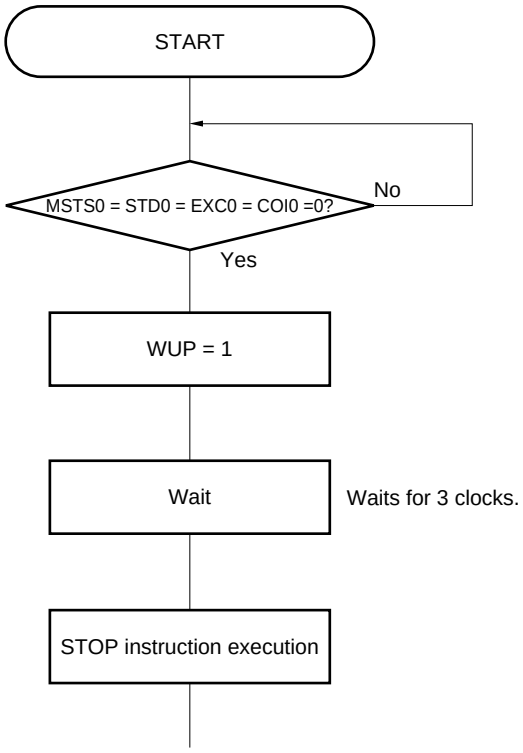
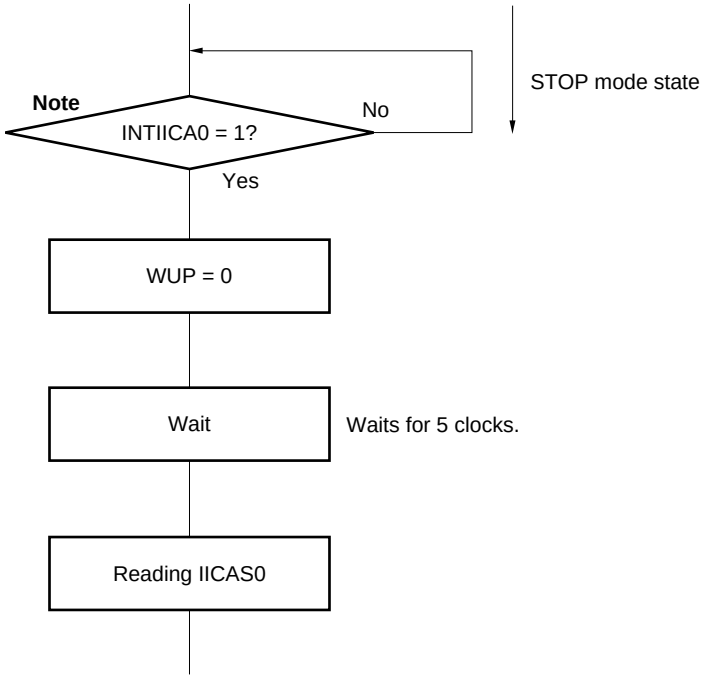


Figure 14-24. Flow When Setting WUP = 0 upon Address Match (Including Extension Code Reception)

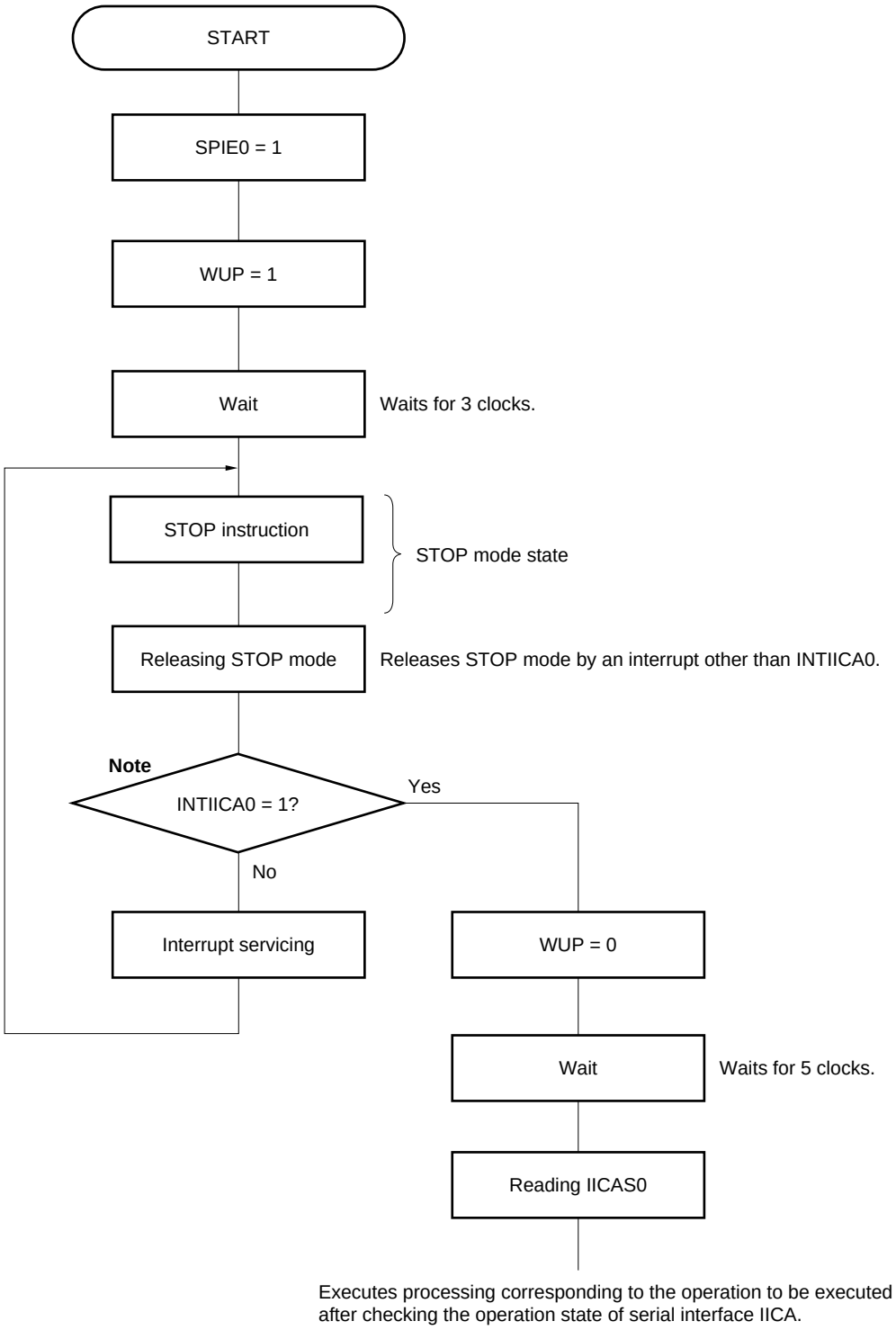


Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Note Perform the processing after “INTIICA0 = 1?” also when an INTIICA0 vector interrupt occurs.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA0) generated from serial interface IICA.

Figure 14-25. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0



Note INTIICA0 also becomes 1 when a STOP condition is issued.

14.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register 0 (IICAF0) = 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 (LREL0) of IICA control register 0 (IICACTL0) to 1 and saving communication).

If bit 1 (STT0) of the IICACTL0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register (IICA) after bit 4 (SPIE0) of the IICACTL0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICA register before the stop condition is detected is invalid.
When the STT0 bit 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

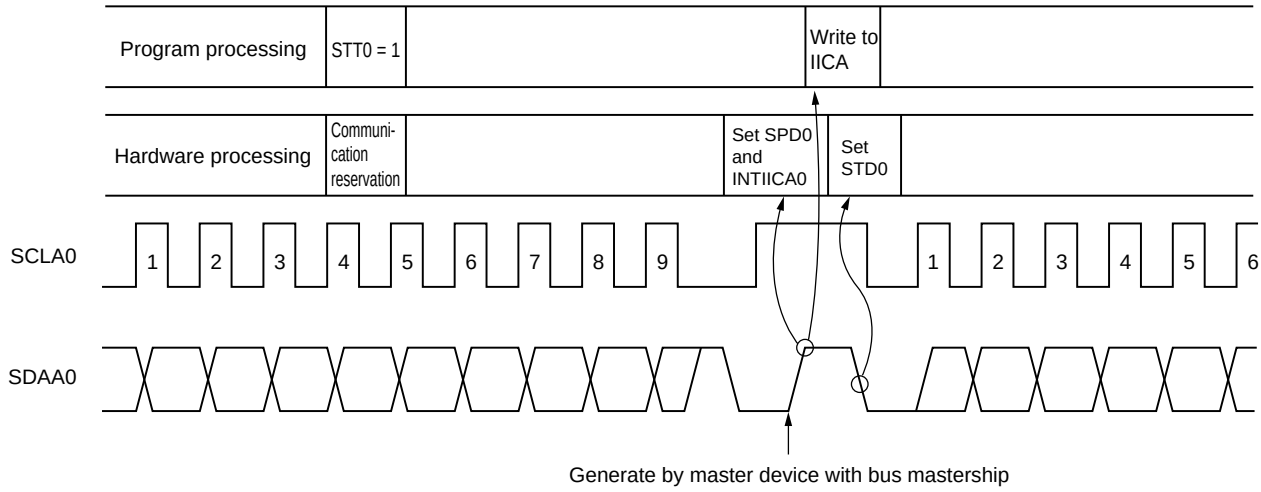
Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICAS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT0 = 1 to checking the MSTS0 flag:
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{PRS}$ (clocks)

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDAA0 and SCLA0 signal falling times
(refer to **CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)** and **CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**)
f_{PRS}: Peripheral hardware clock frequency

Figure 14-26 shows the communication reservation timing.

Figure 14-26. Communication Reservation Timing



Remark IICA: IICA shift register
STT0: Bit 1 of IICA control register 0 (IICACTL0)
STD0: Bit 1 of IICA status register 0 (IICAS0)
SPD0: Bit 0 of IICA status register 0 (IICAS0)

Communication reservations are accepted via the timing shown in Figure 14-27. After bit 1 (STD0) of the IICA status register 0 (IICAS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 0 (IICACTL0) to 1 before a stop condition is detected.

Figure 14-27. Timing for Accepting Communication Reservations

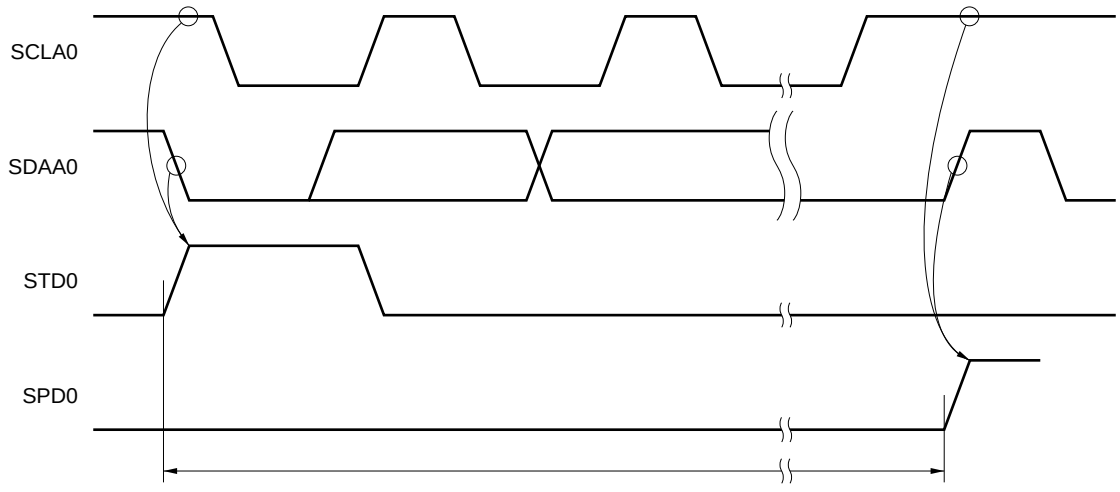
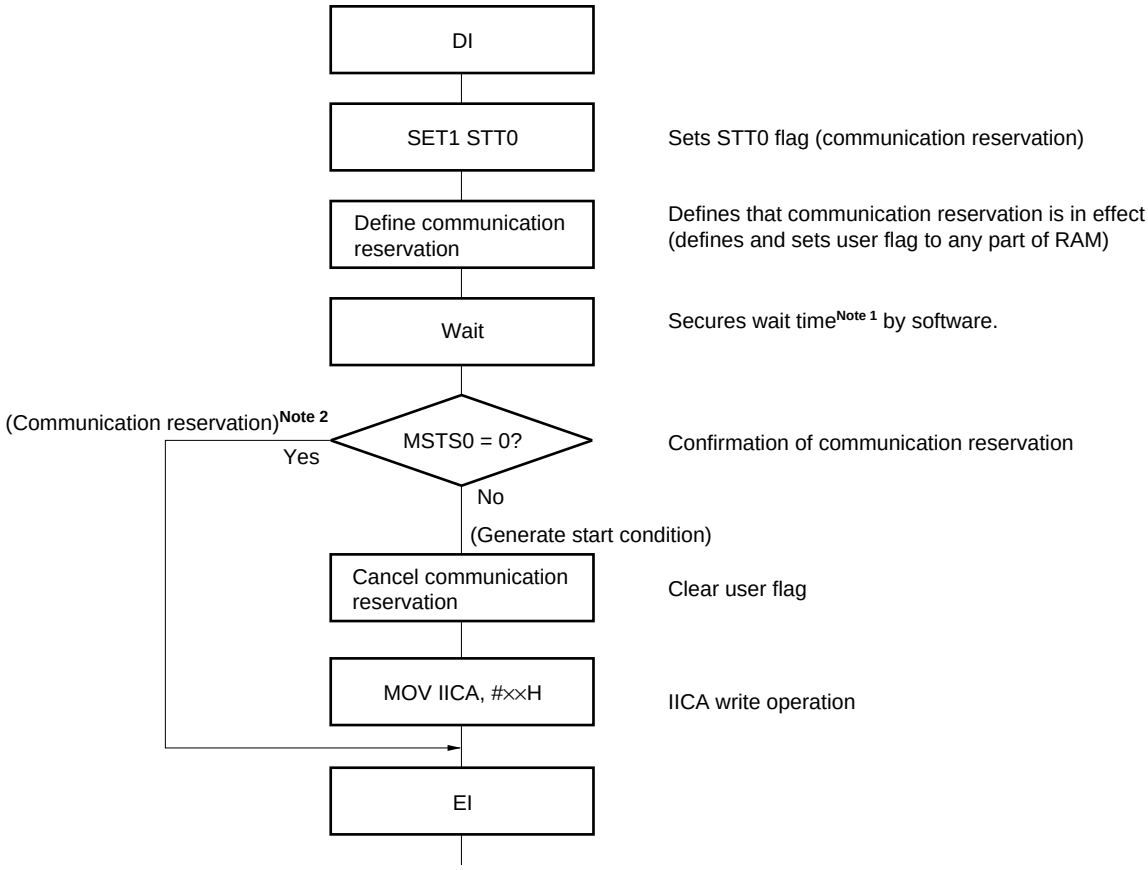


Figure 14-28 shows the communication reservation protocol.

Figure 14-28. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{PRS}$ (clocks)

2. The communication reservation operation executes a write to the IICA shift register (IICA) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IICA control register 0 (IICACTL0)
MSTS0: Bit 7 of IICA status register 0 (IICAS0)
IICA: IICA shift register
IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDAA0 and SCLA0 signal falling times
(refer to **CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)** and **CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**)
f_{PRS}: Peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register 0 (IICAF0) = 1)

When bit 1 (STT0) of the IICA control register 0 (IICACTL0) 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 (LREL0) of the IICACTL0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check the STCF flag (bit 7 of IICF0 register). It takes up to 5 clocks until the STCF flag is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

14.5.15 Cautions

(1) When STCEN (bit 1 of IICA flag register 0 (IICAF0)) = 0

Immediately after I^2C operation is enabled (IICE0 = 1), the bus communication status (the IICBSY flag (bit 6 of the IICAF0 register) = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 1 (IICACTL1).
<2> Set bit 7 (IICE0) of IICA control register 0 (IICACTL0) to 1.
<3> Set bit 0 (SPT0) of IICACTL0 to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 (bit 1 of the IICA control register 0 (IICACTL0)) = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDAA0 pin is low and the SCLA0 pin is high, the macro of I^2C recognizes that the SDAA0 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 (SPIE0) of the IICACTL0 register to 0 to disable generation of an interrupt request signal (INTIICA0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of the IICACTL0 register to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of the IICACTL0 register to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting the IICE0 bit to 1), to forcibly disable detection.

(4) Setting the STT0 and SPT0 bits (bits 1 and 0 of the IICACTL0 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set SPIE0 (bit 4 of the IICACTL0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register (IICA) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIE0 bit to 1 when the MSTS0 bit (bit 7 of the IICA status register (IICAS0)) is detected by software.

14.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0/Fx2-L 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/Fx2-L 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/Fx2-L 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

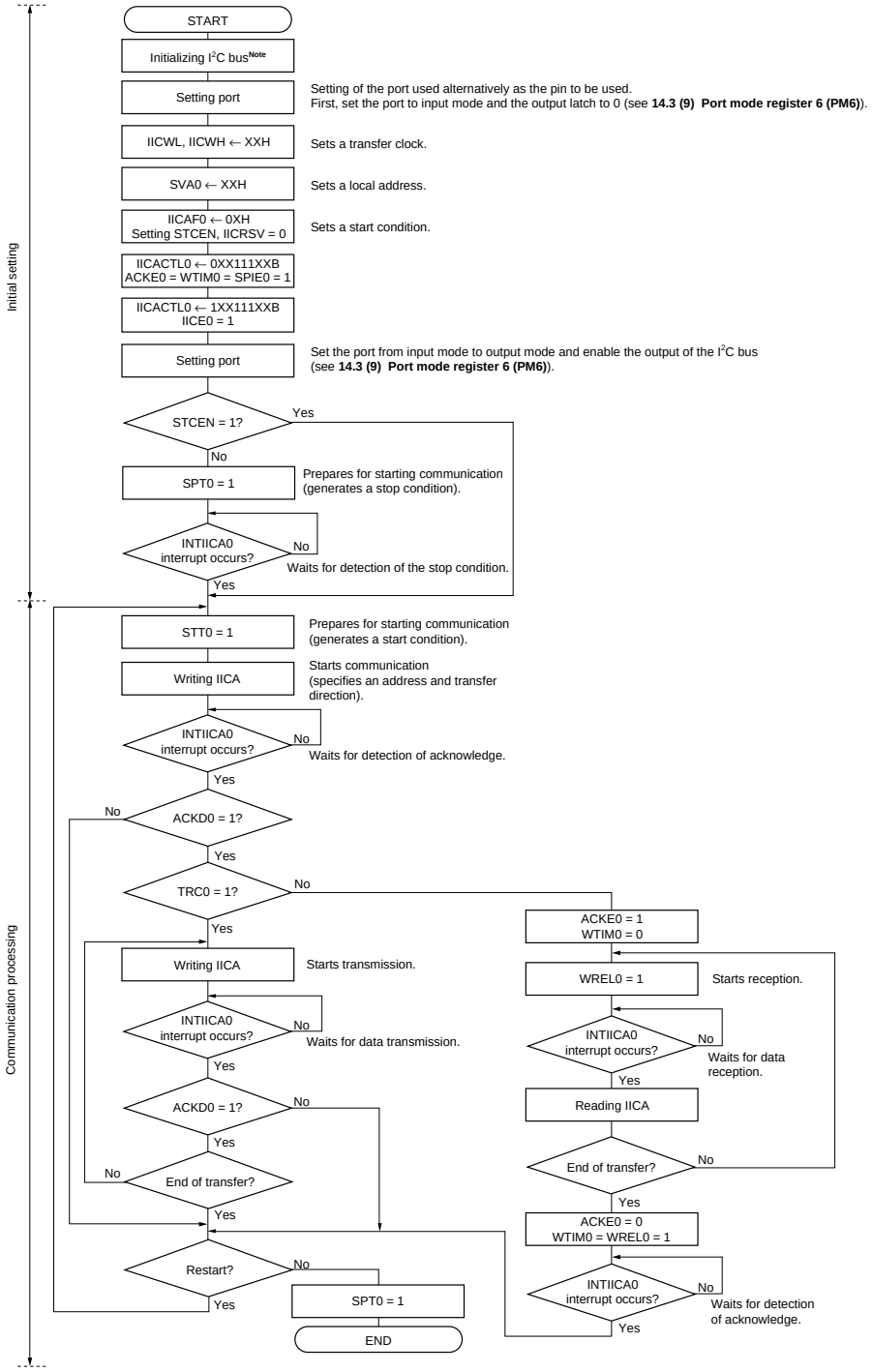
An example of when the 78K0/Fx2-L microcontrollers are used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 14-29. Master Operation in Single-Master System

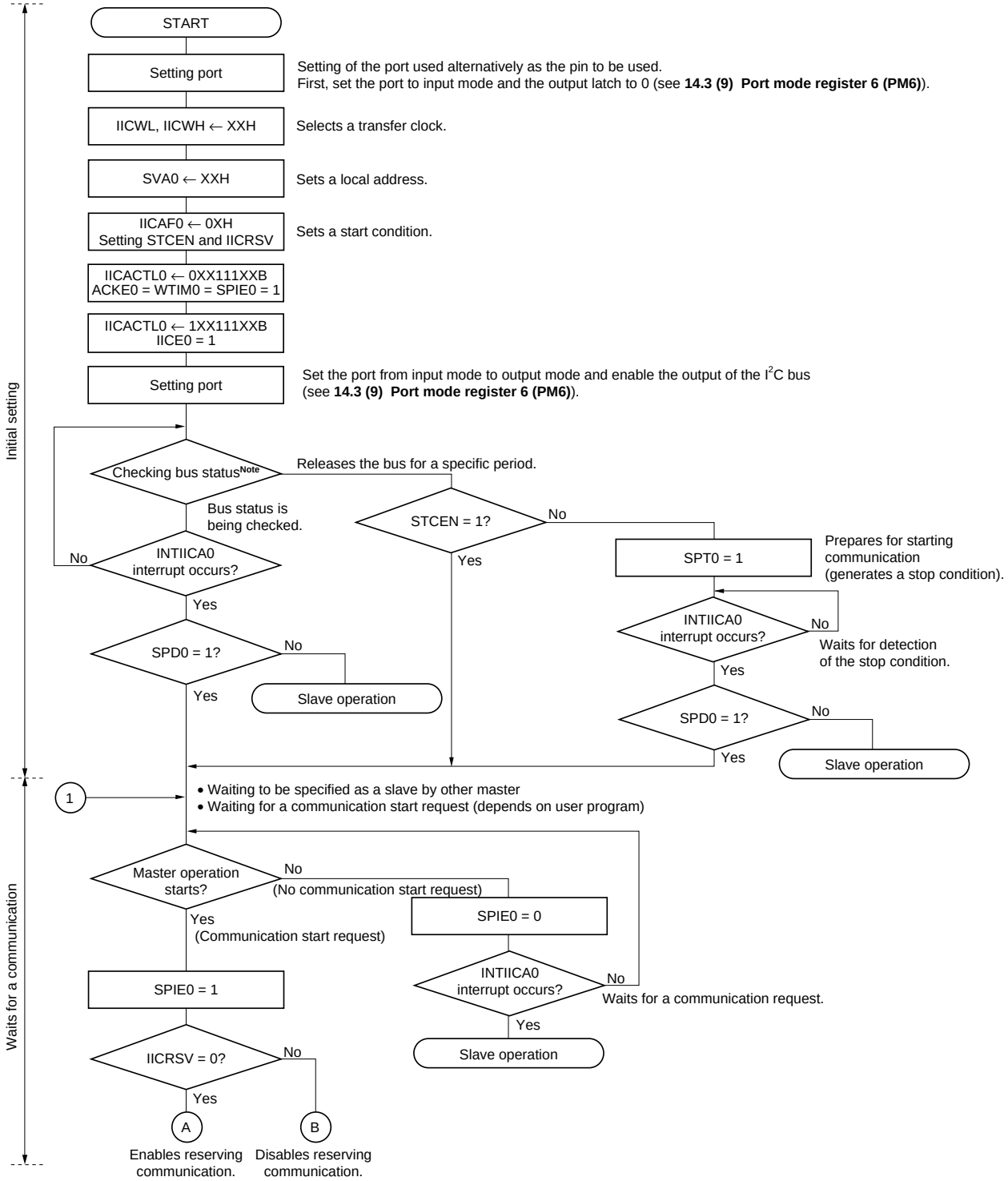


Note Release (SCLA0 and SDAA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the SDAA0 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.

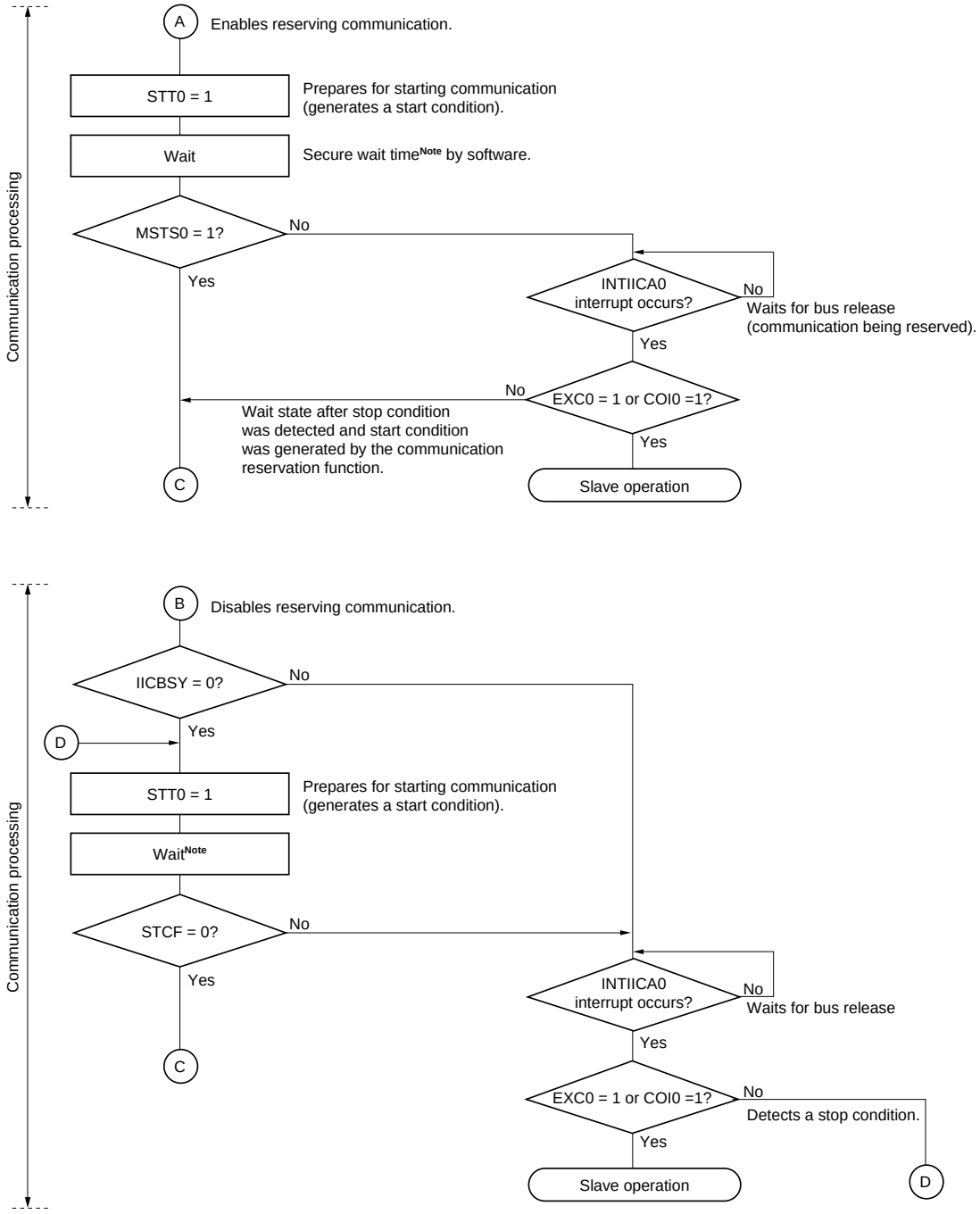
(2) Master operation in multi-master system

Figure 14-30. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDAA0 pin is constantly at low level, decide whether to release the I^2C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

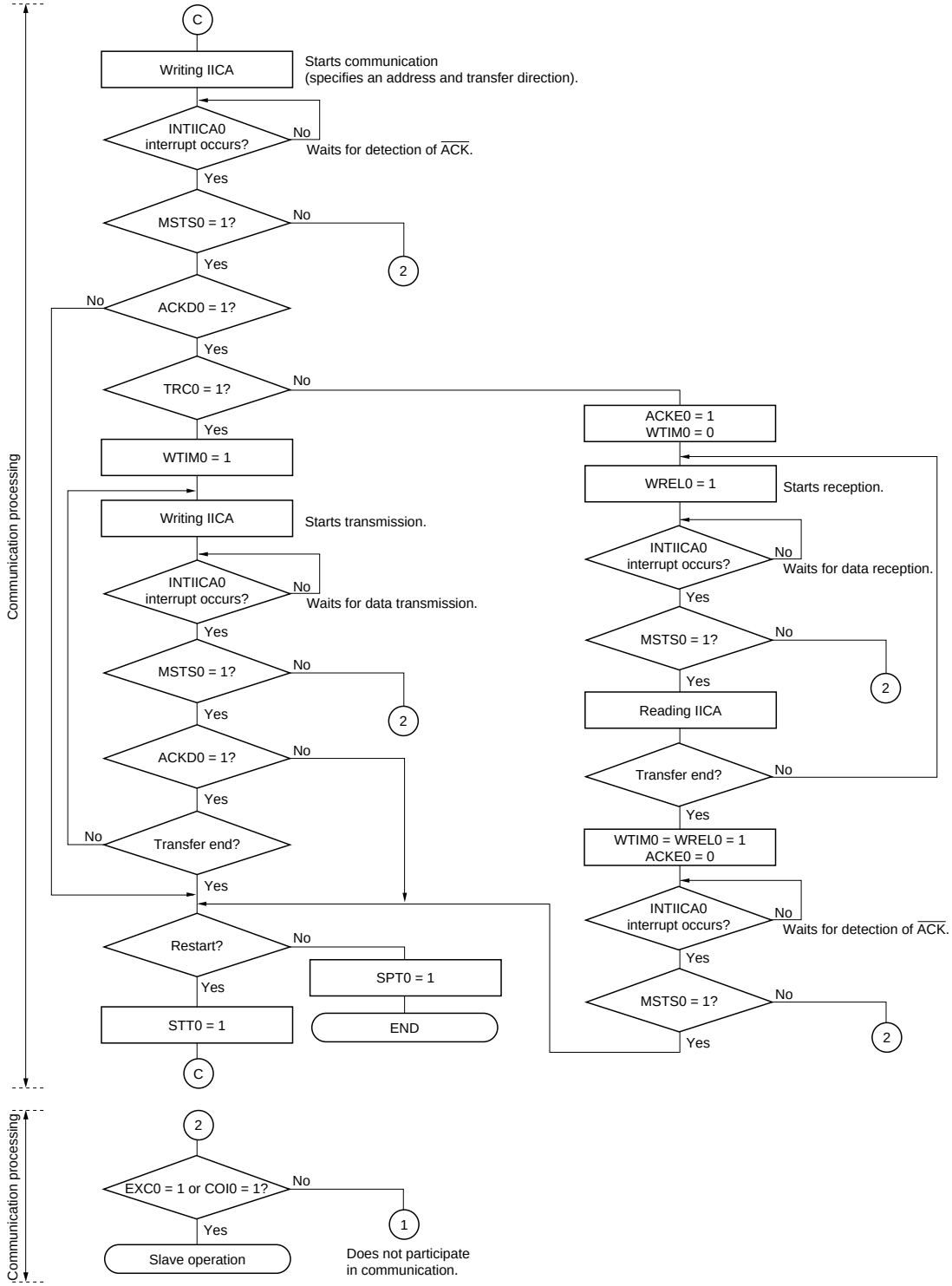
Figure 14-30. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{PRS}$ (clocks)

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDAA0 and SCLA0 signal falling times (refer to **CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)** and **CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**)
f_{PRS}: Peripheral hardware clock frequency

Figure 14-30. 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 MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

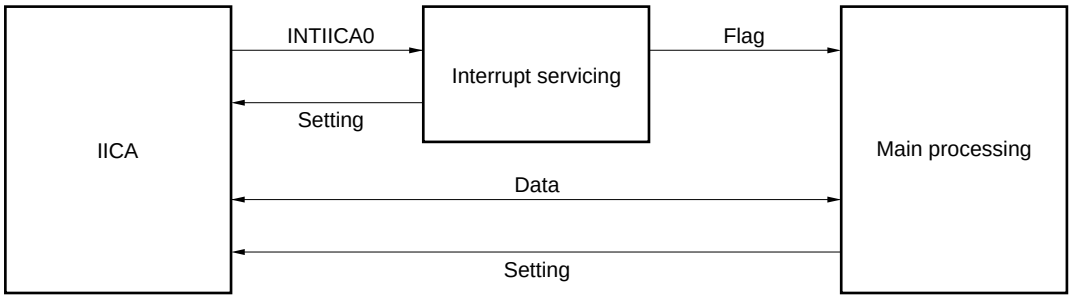
3. To use the device as a slave in a multi-master system, check the status by using the IICAS0 and IICAF0 registers each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA0 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 INTIICA0 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 INTIICA0.

<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 INTIICA0 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 TRC0.

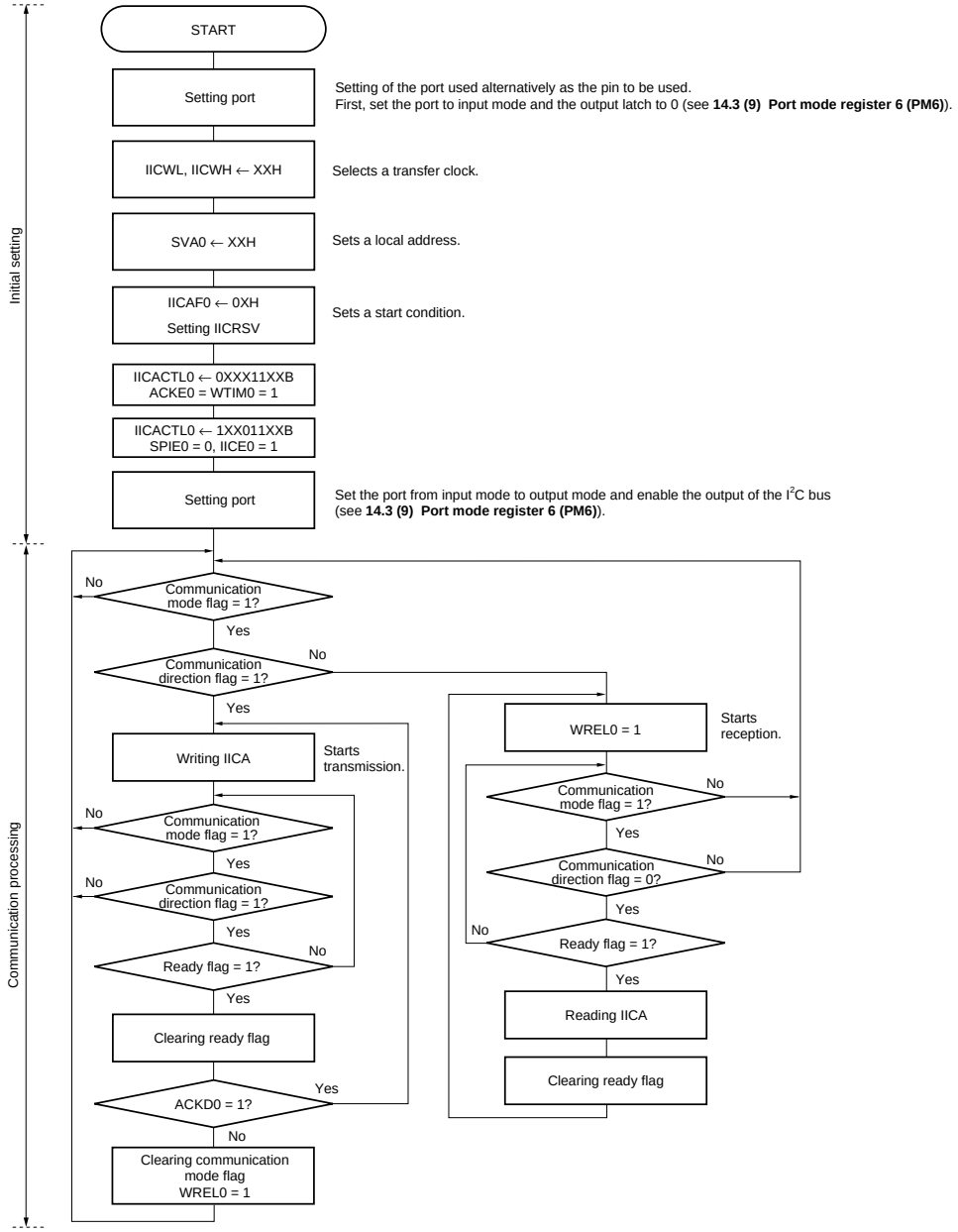
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{\text{ACK}}$. If $\overline{\text{ACK}}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{\text{ACK}}$ 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 14-31. Slave Operation Flowchart (1)



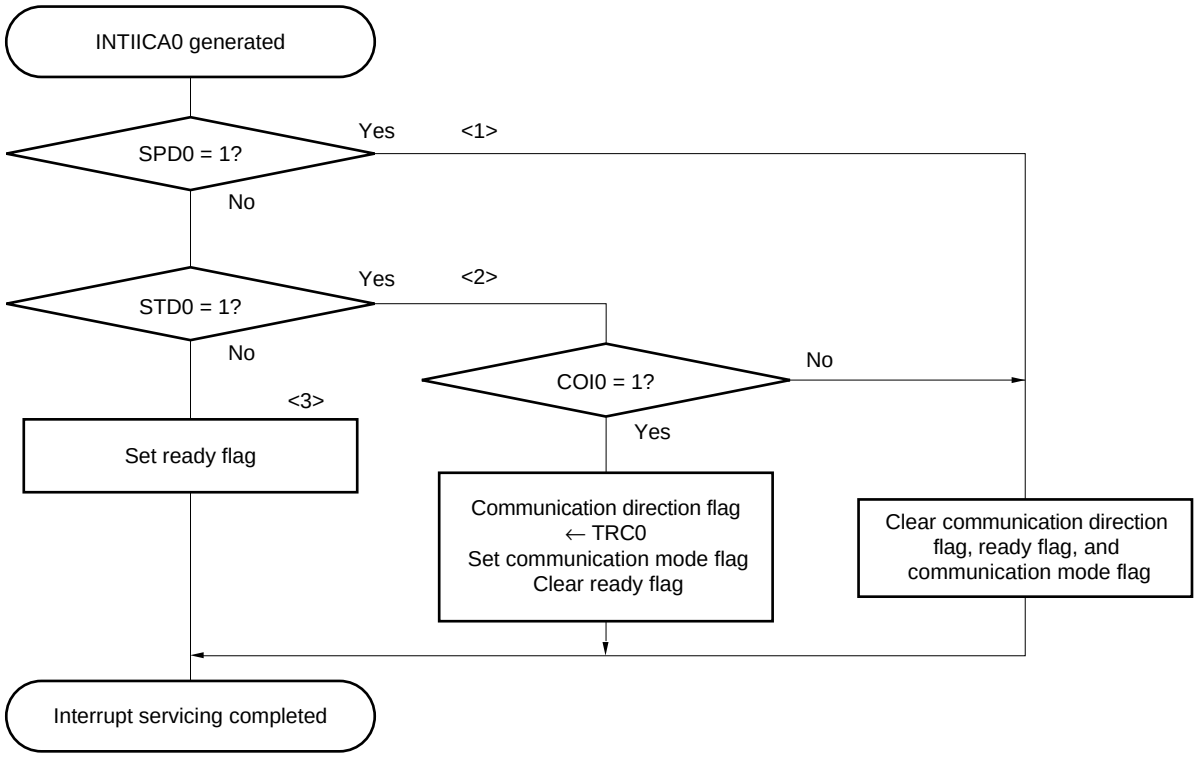
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address 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 14-32 Slave Operation Flowchart (2).

Figure 14-32. Slave Operation Flowchart (2)



14.5.17 Timing of I^2C interrupt request (INTIICA0) occurrence

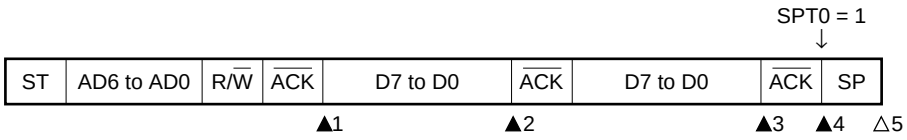
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICAS0 register when the INTIICA0 signal is generated are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0

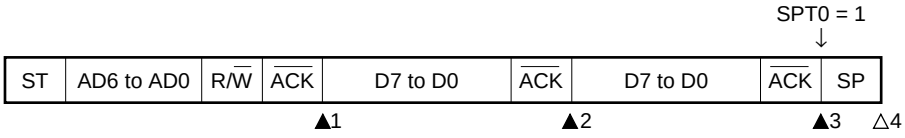


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B
▲3: IICAS0 = 1000×000B (Sets WTIM0 to 1)[Note]
▲4: IICAS0 = 1000××00B (Sets SPT0 to 1)[Note]
△5: IICAS0 = 00000001B

Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

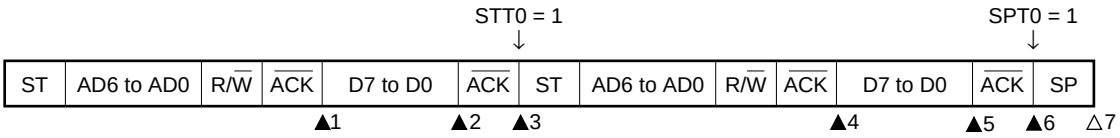


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×100B
▲3: IICAS0 = 1000××00B (Sets SPT0 to 1)
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

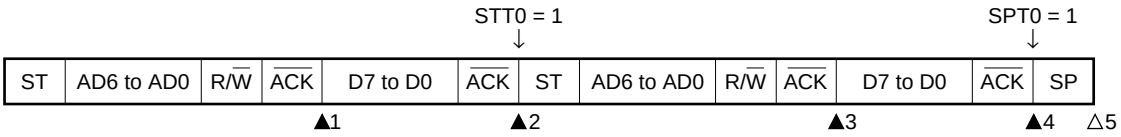


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)[Note 1]
▲3: IICAS0 = 1000××00B (Clears WTIM0 to 0[Note 2], sets STT0 to 1)
▲4: IICAS0 = 1000×110B
▲5: IICAS0 = 1000×000B (Sets WTIM0 to 1)[Note 3]
▲6: IICAS0 = 1000××00B (Sets SPT0 to 1)
△7: IICAS0 = 00000001B

Notes 1. To generate a start condition, set WTIM0 to 1 and change the timing for generating the INTIICA0 interrupt request signal.
2. Clear WTIM0 to 0 to restore the original setting.
3. To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

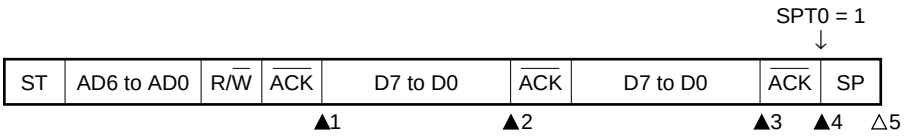


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000××00B (Sets STT0 to 1)
▲3: IICAS0 = 1000×110B
▲4: IICAS0 = 1000××00B (Sets SPT0 to 1)
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

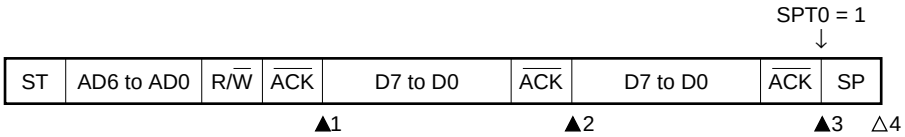


▲1: IICAS0 = 1010×110B
▲2: IICAS0 = 1010×000B
▲3: IICAS0 = 1010×000B (Sets WTIM0 to 1)[Note]
▲4: IICAS0 = 1010××00B (Sets SPT0 to 1)
△5: IICAS0 = 00000001B

Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



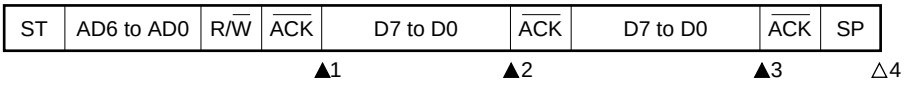
▲1: IICAS0 = 1010×110B
▲2: IICAS0 = 1010×100B
▲3: IICAS0 = 1010××00B (Sets SPT0 to 1)
△4: IICAS0 = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

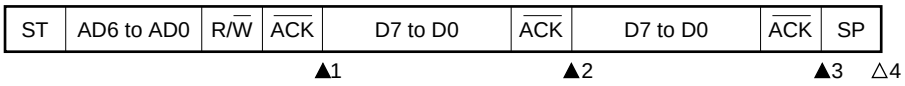
(i) When WTIM0 = 0



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0001×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

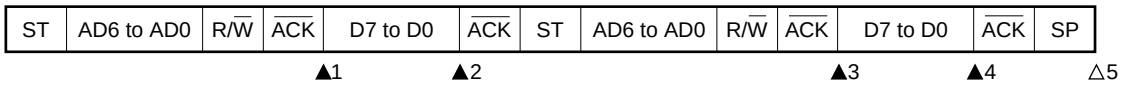


▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×100B
▲3: IICAS0 = 0001××00B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

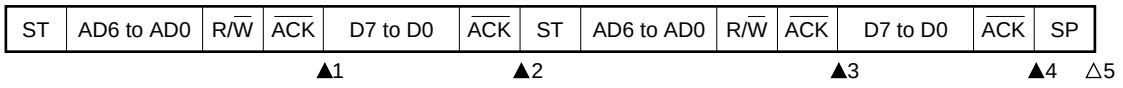
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0001×110B
▲4: IICAS0 = 0001×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001××00B
▲3: IICAS0 = 0001×110B
▲4: IICAS0 = 0001××00B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

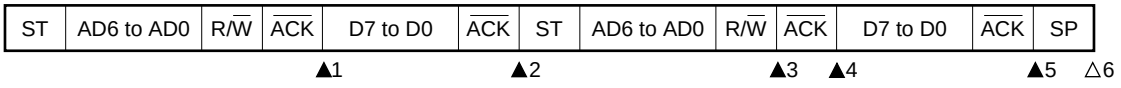
(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= extension code))

▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0010×010B
▲4: IICAS0 = 0010×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))



▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001××00B
▲3: IICAS0 = 0010×010B
▲4: IICAS0 = 0010×110B
▲5: IICAS0 = 0010××00B
△6: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

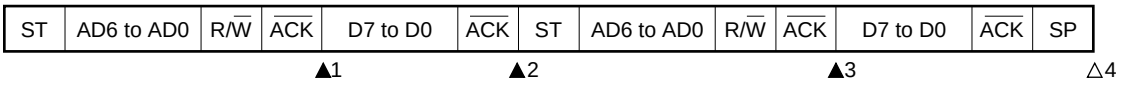
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

▲1 ▲2 ▲3 △4

▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))



▲1 ▲2 ▲3 △4

▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001××00B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 00000001B

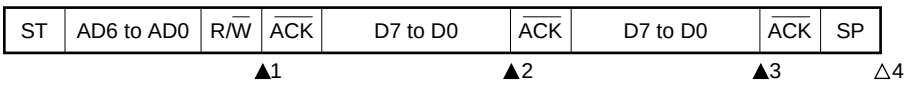
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(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

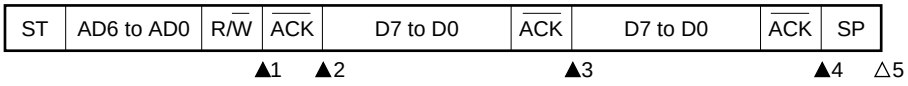
(i) When WTIM0 = 0



▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0010×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

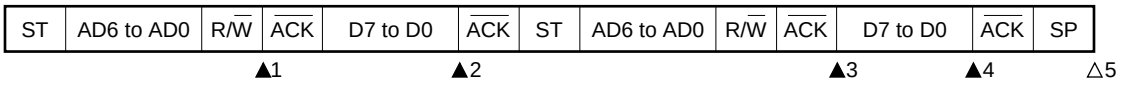


▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010×100B
▲4: IICAS0 = 0010××00B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

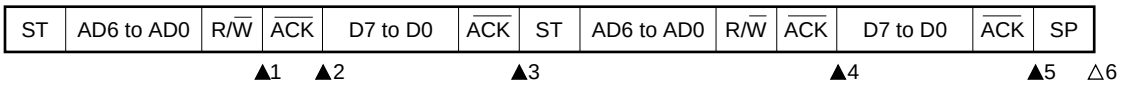
(i) When WTIM0 = 0 (after restart, matches SVA0)



▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0001×110B
▲4: IICAS0 = 0001×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)

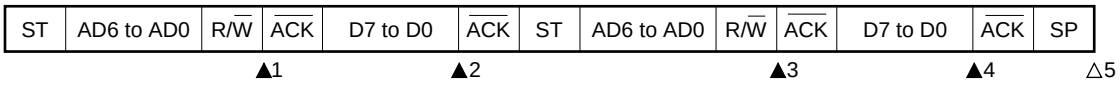


▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010××00B
▲4: IICAS0 = 0001×110B
▲5: IICAS0 = 0001××00B
△6: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

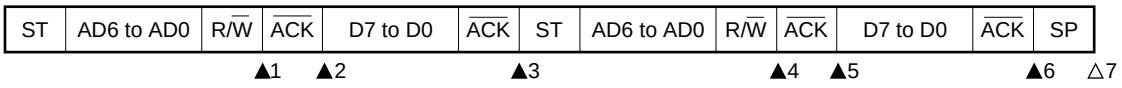
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0010×010B
▲4: IICAS0 = 0010×000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)



▲1: IICAS0 = 0010×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010××00B
▲4: IICAS0 = 0010×010B
▲5: IICAS0 = 0010×110B
▲6: IICAS0 = 0010××00B
△7: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

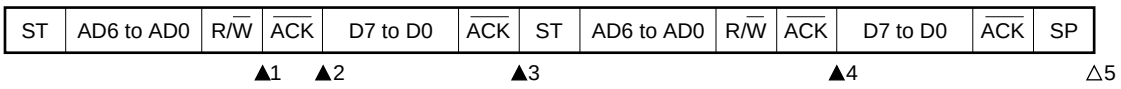
(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

▲1: IICAS0 = 00100010B
▲2: IICAS0 = 00100000B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

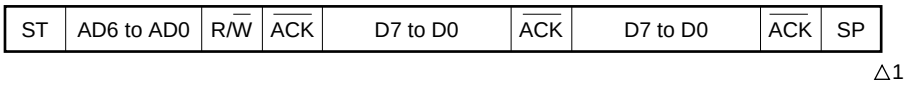


▲1: IICAS0 = 00100010B
▲2: IICAS0 = 00100110B
▲3: IICAS0 = 00100×00B
▲4: IICAS0 = 00000110B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICAS0 = 00000001B

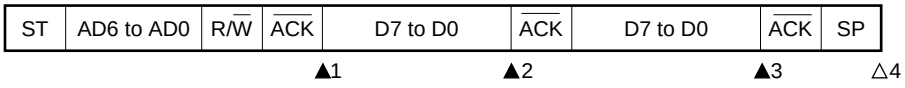
Remark △: Generated only when SPIE0 = 1

(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 MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

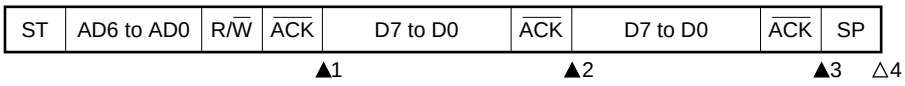
(i) When WTIM0 = 0



▲1: IICAS0 = 0101×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 0001×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



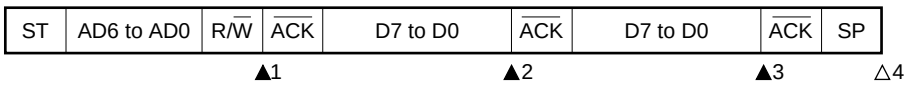
▲1 ▲2 ▲3 △4

▲1: IICAS0 = 0101×110B
▲2: IICAS0 = 0001×100B
▲3: IICAS0 = 0001××00B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIM0 = 0

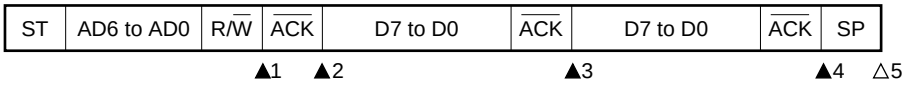


▲1 ▲2 ▲3 △4

▲1: IICAS0 = 0110×010B
▲2: IICAS0 = 0010×000B
▲3: IICAS0 = 0010×000B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



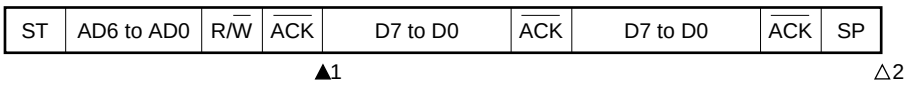
▲1: IICAS0 = 0110×010B
▲2: IICAS0 = 0010×110B
▲3: IICAS0 = 0010×100B
▲4: IICAS0 = 0010××00B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 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 MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

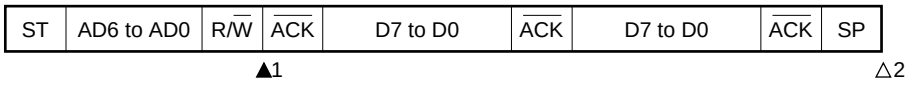
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICAS0 = 01000110B
△2: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

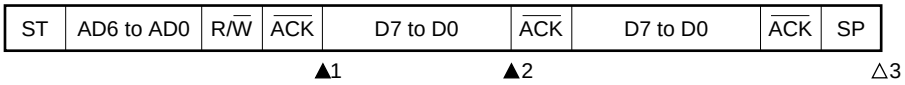


▲1: IICAS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

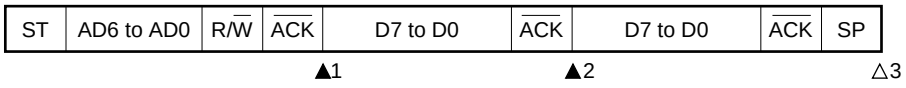
(i) When WTIM0 = 0



▲1: IICAS0 = 10001110B
▲2: IICAS0 = 01000000B
△3: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

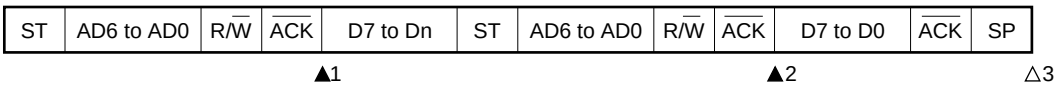


▲1: IICAS0 = 10001110B
▲2: IICAS0 = 01000100B
△3: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

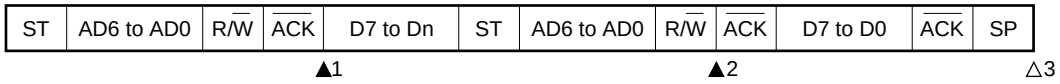
(i) Not extension code (Example: unmatches with SVA0)



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 01000110B
△3: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICAS0 = 1000×110B

▲2: IICAS0 = 01100010B

Sets LREL0 = 1 by software

△3: IICAS0 = 00000001B

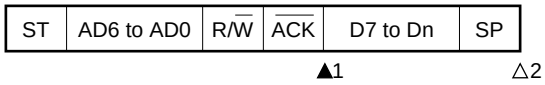
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICAS0 = 10000110B

△2: IICAS0 = 01000001B

Remark ▲: Always generated

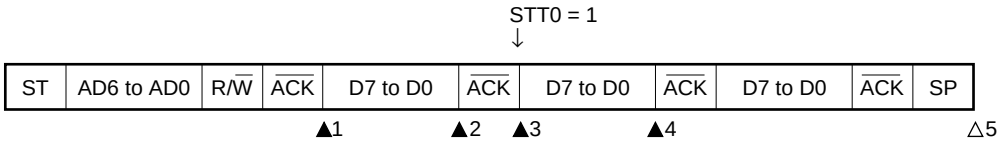
△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

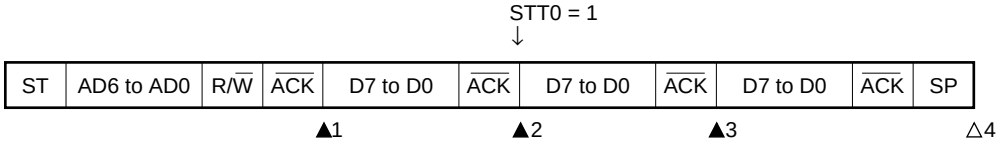
(i) When WTIM0 = 0



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICAS0 = 1000×100B (Clears WTIM0 to 0)
▲4: IICAS0 = 01000000B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

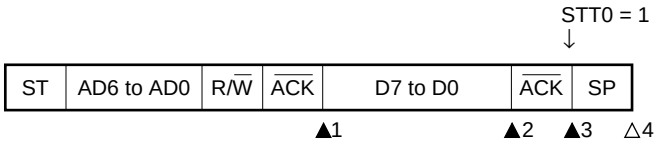


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×100B (Sets STT0 to 1)
▲3: IICAS0 = 01000100B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

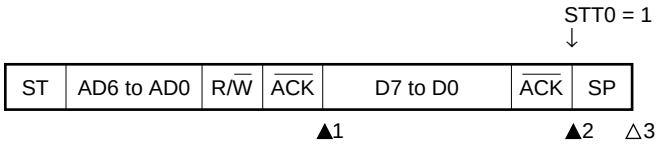
(i) When WTIM0 = 0



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICAS0 = 1000××00B (Sets STT0 to 1)
△4: IICAS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

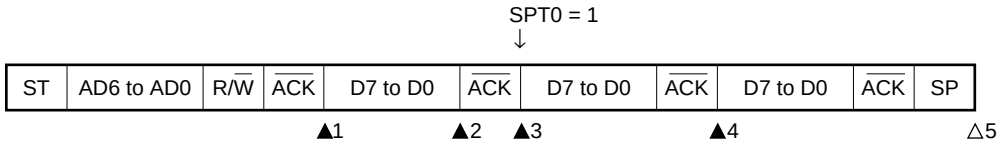


▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000××00B (Sets STT0 to 1)
△3: IICAS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

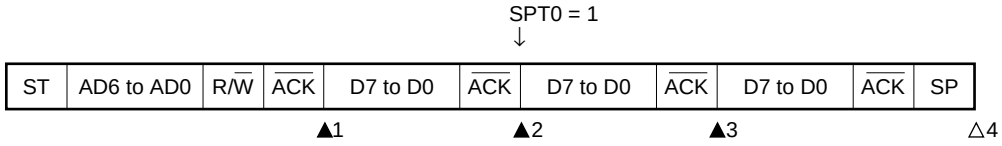
(i) When WTIM0 = 0



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICAS0 = 1000×100B (Clears WTIM0 to 0)
▲4: IICAS0 = 01000100B
△5: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1



▲1: IICAS0 = 1000×110B
▲2: IICAS0 = 1000×100B (Sets SPT0 to 1)
▲3: IICAS0 = 01000100B
△4: IICAS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

14.6 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 TRC0 bit (bit 3 of the IICA status register 0 (IICAS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

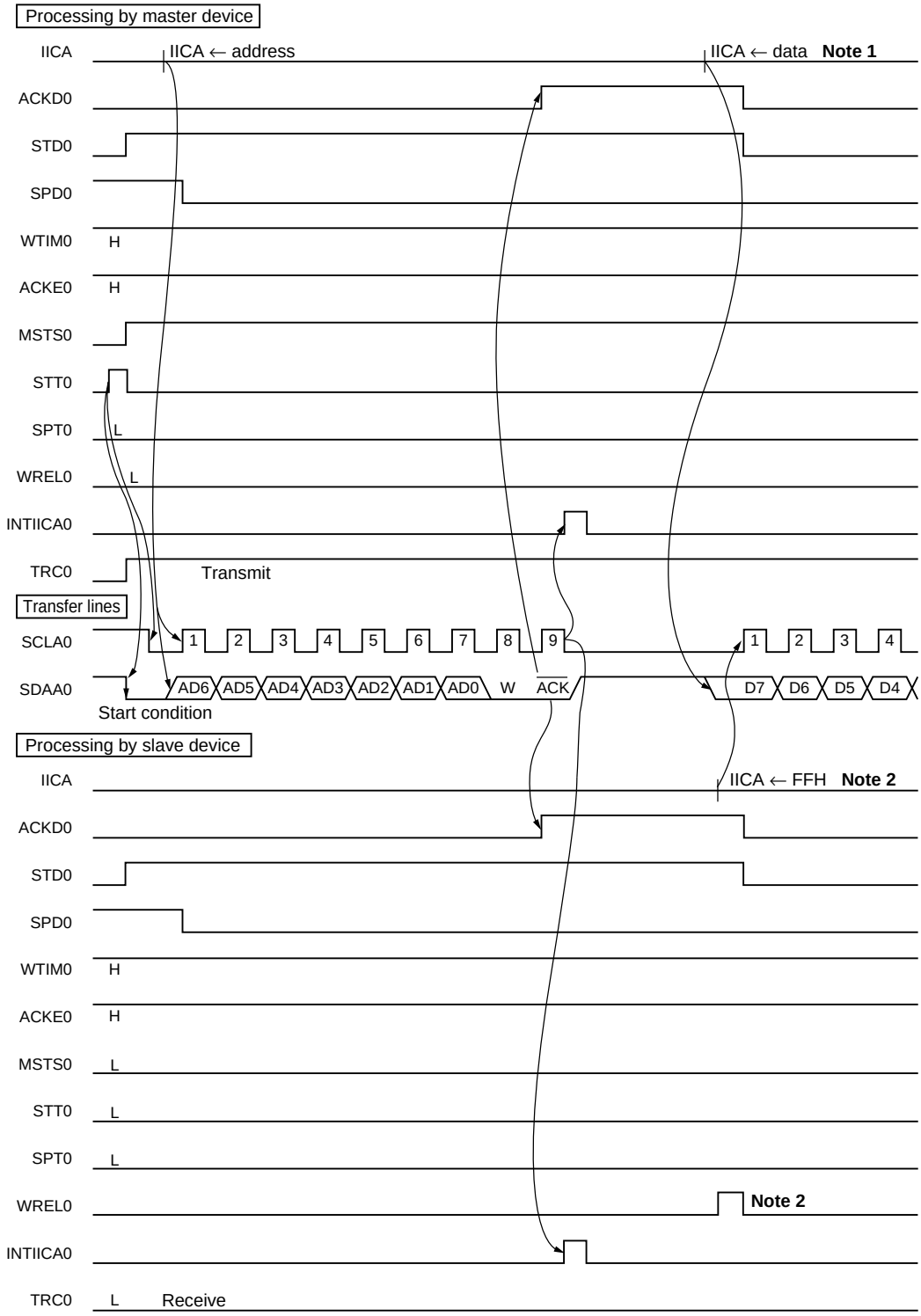
Figures 14-33 and 14-34 show timing charts of the data communication.

The IICA shift register (IICA)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA at the rising edge of SCLA0.

Figure 14-33. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(1) Start condition ~ address

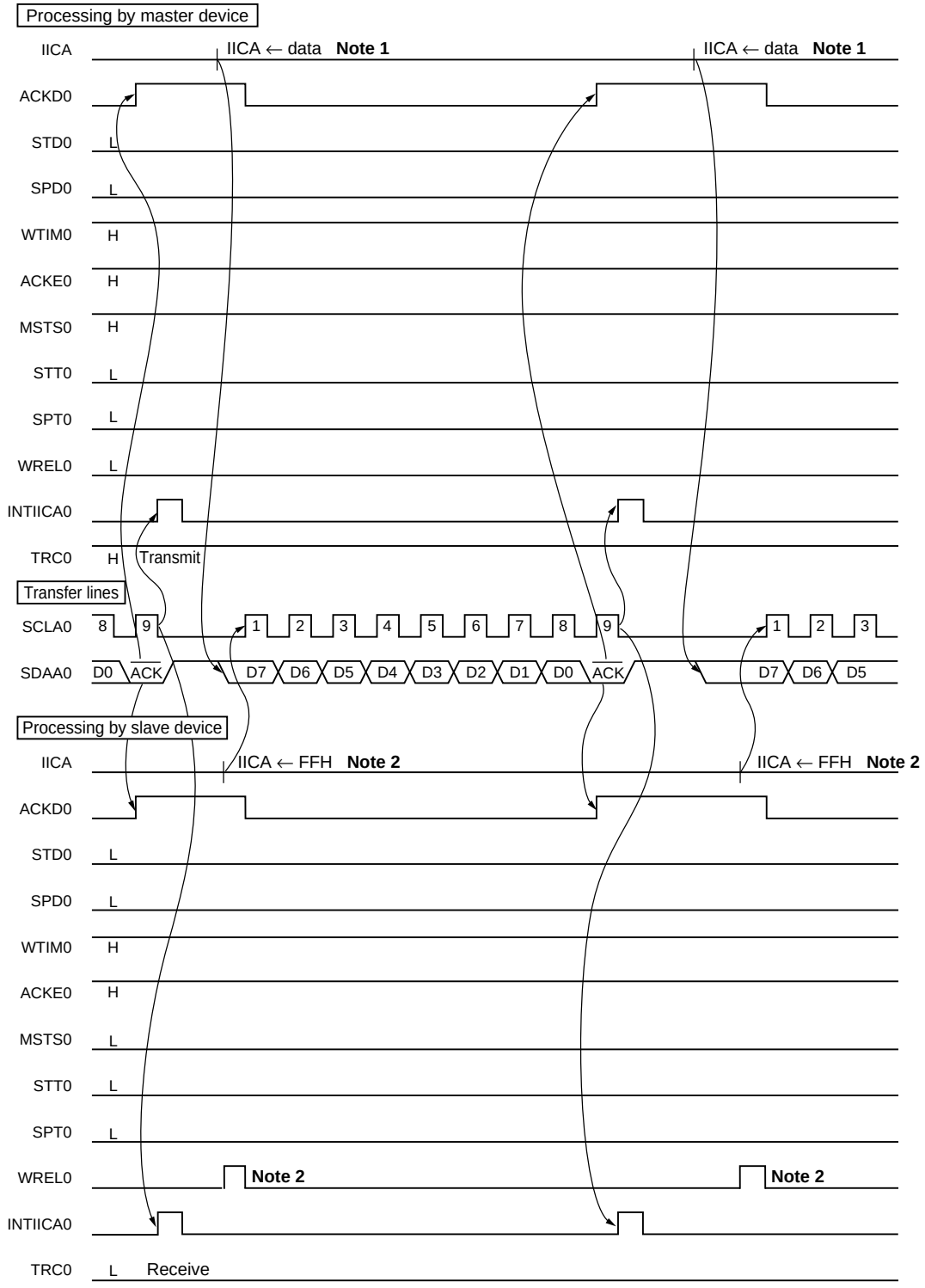


Notes 1. Write data to IICA, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL0.

Figure 14-33. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(2) Data

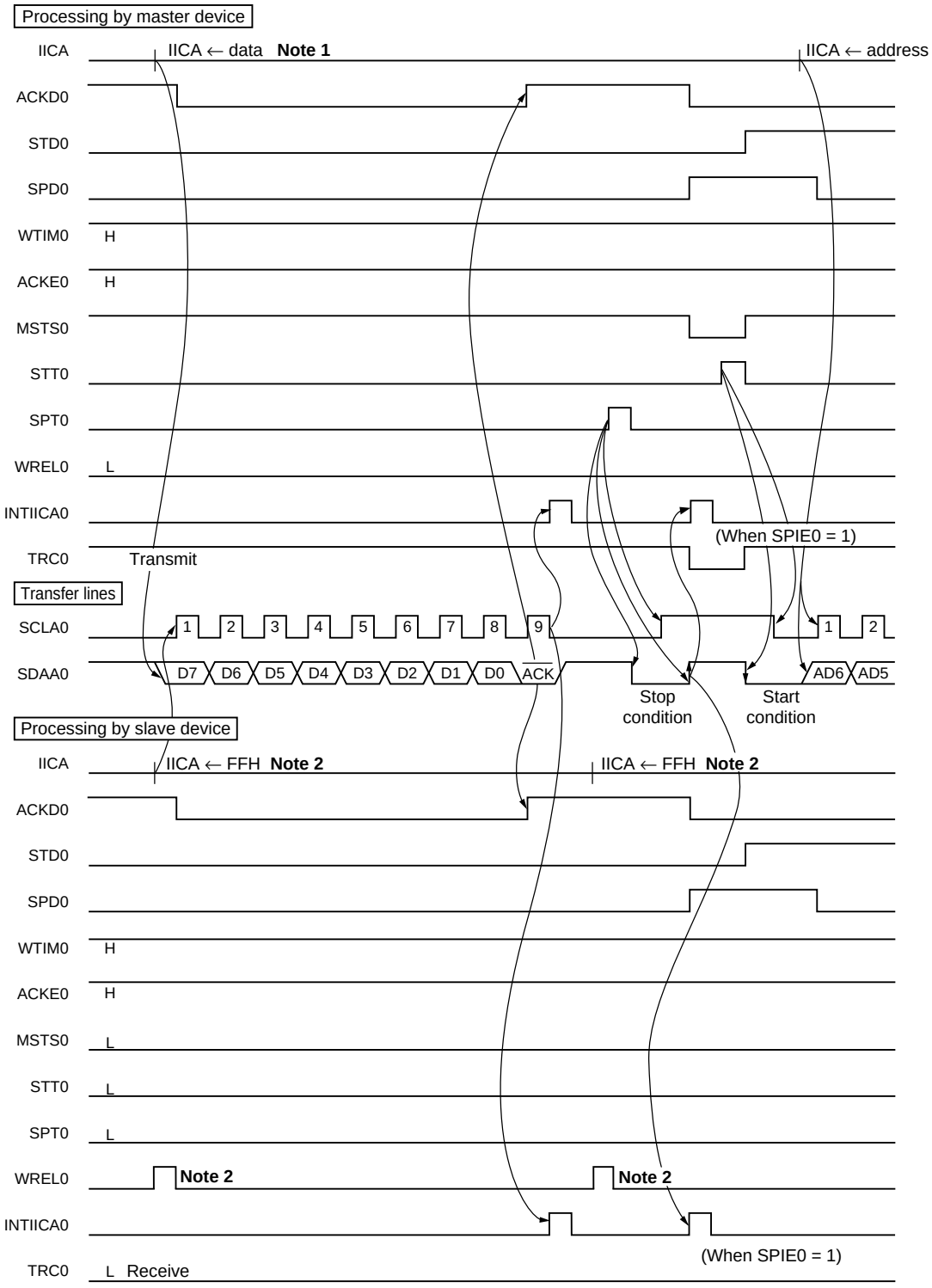


Notes 1. Write data to IICA, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL0.

Figure 14-33. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition

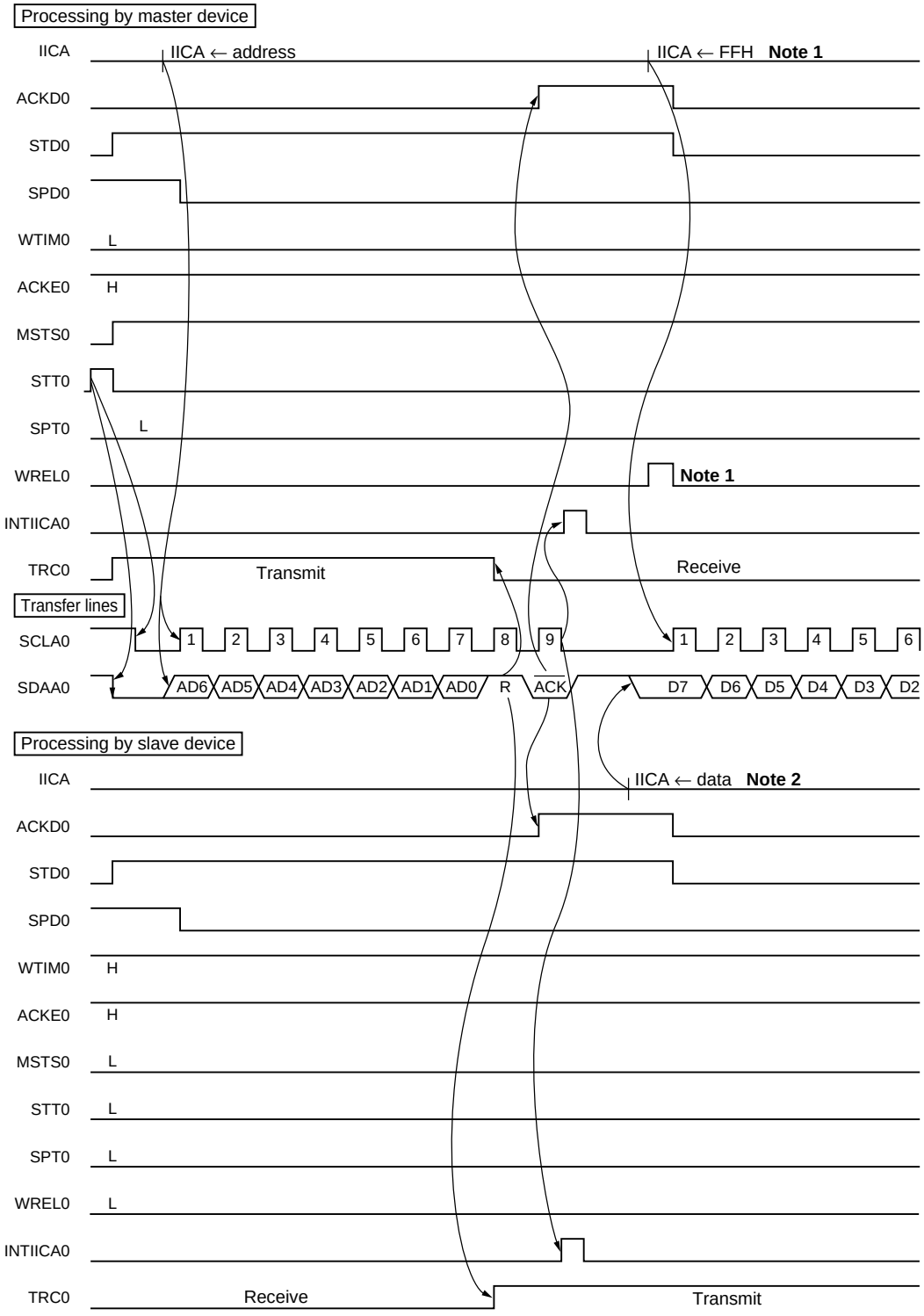


Notes 1. Write data to IICA, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set WREL0.

Figure 14-34. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address

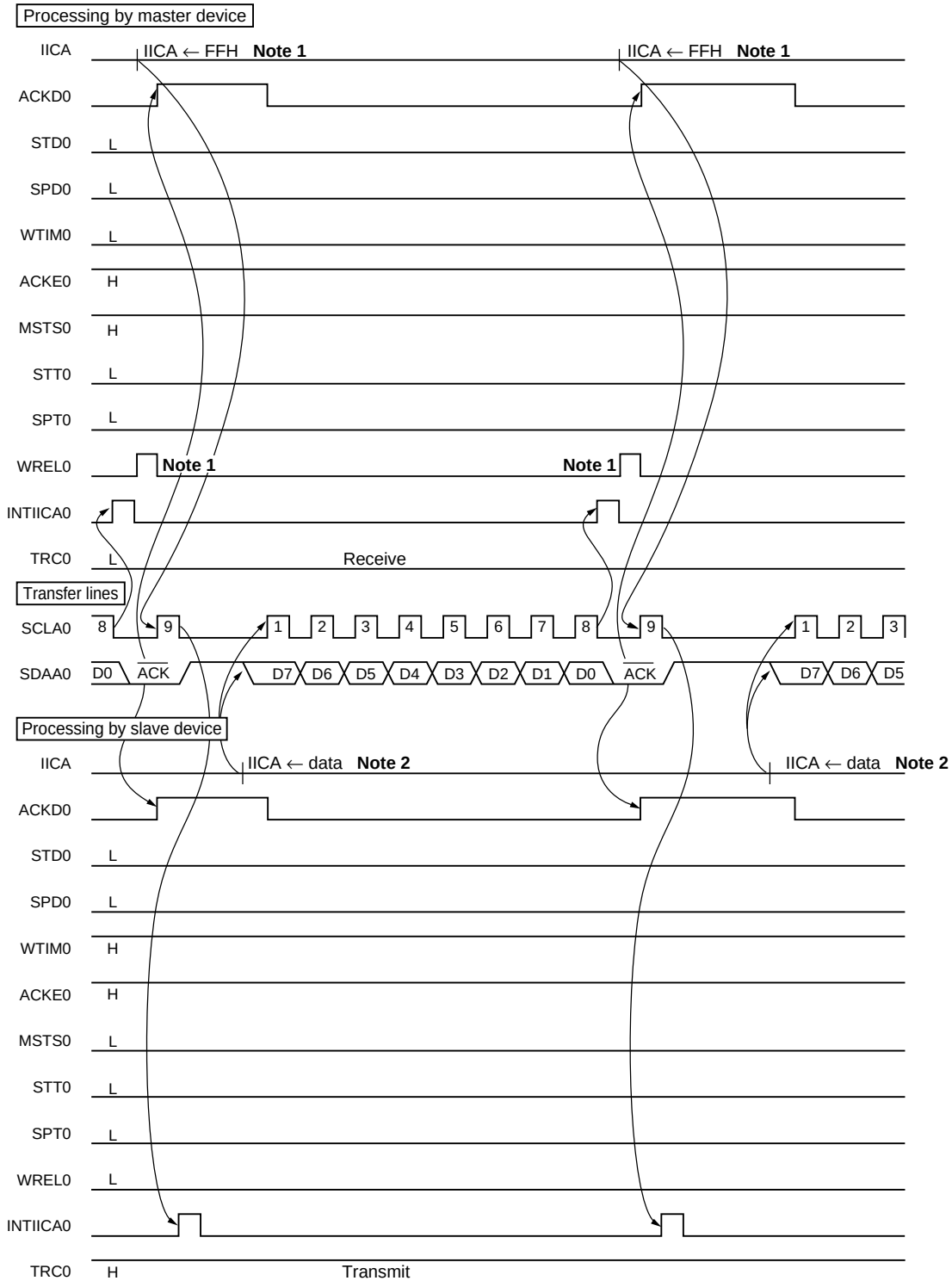


Notes 1. To cancel master wait, write “FFH” to IICA or set WREL0.

2. Write data to IICA, not setting WREL0, in order to cancel a wait state during slave transmission.

Figure 14-34. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Data

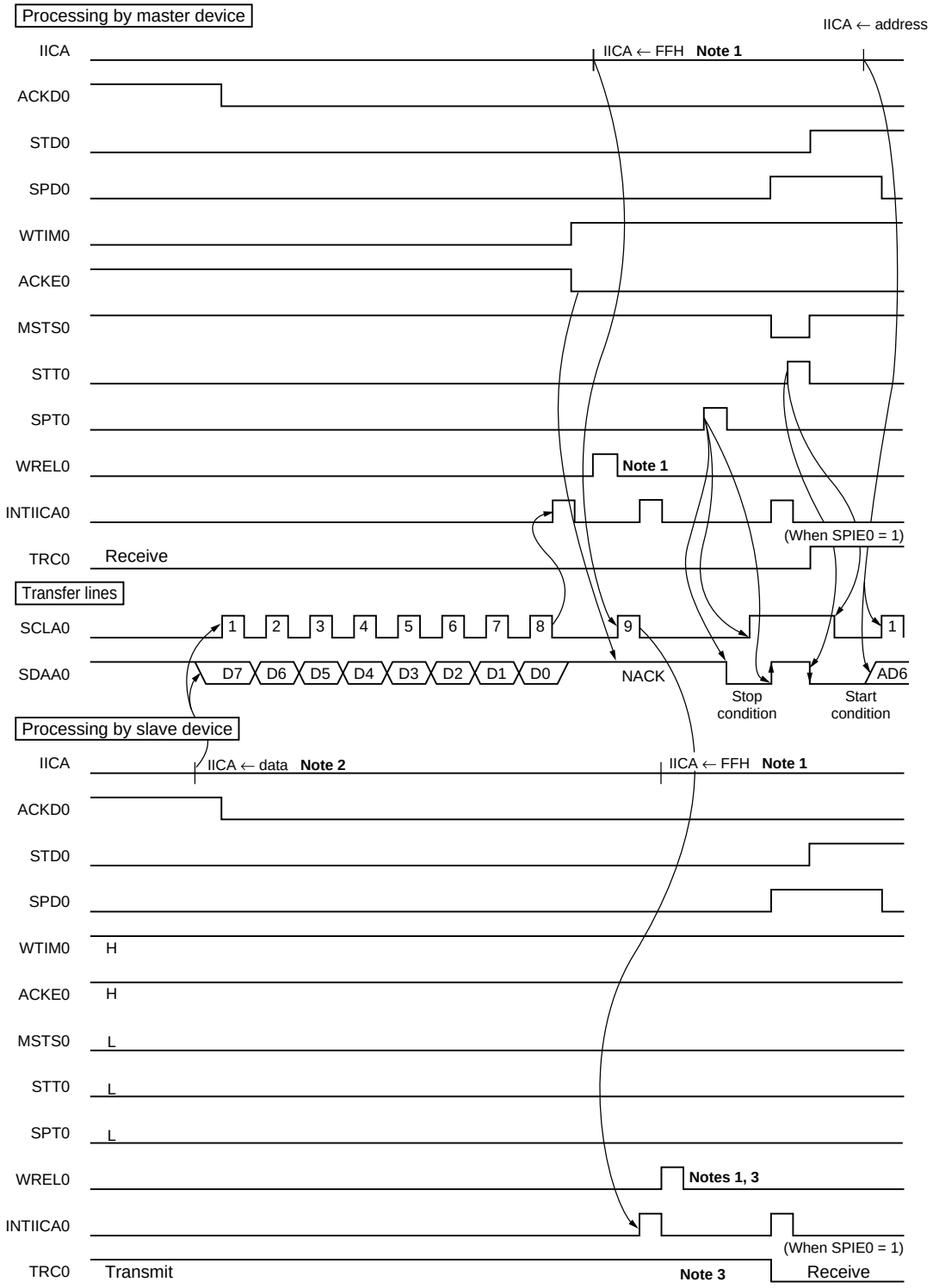


Notes 1. To cancel master wait, write “FFH” to IICA or set WREL0.

2. Write data to IICA, not setting WREL0, in order to cancel a wait state during slave transmission.

Figure 14-34. Example of Slave to Master Communication
(When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

(3) Stop condition



Notes 1. To cancel wait, write “FFH” to IICA or set WREL0.

2. Write data to IICA, not setting WREL0, in order to cancel a wait state during slave transmission.

3. If a wait state during slave transmission is canceled by setting WREL0, TRC0 will be cleared.

CHAPTER 15 SERIAL INTERFACE CSI11

Item	78K0/FY2-L (16 Pins)	78K0/FA2-L (20 Pins)	78K0/FB2-L (30 Pins)
Serial interface CSI11	Not mounted		Mounted

15.1 Functions of Serial Interface CSI11

Serial interface CSI11 has 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, refer to **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 ($\overline{SCK11}$) and two serial data lines (SI11 and SO11).

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, refer to **15.4.2 3-wire serial I/O mode**.

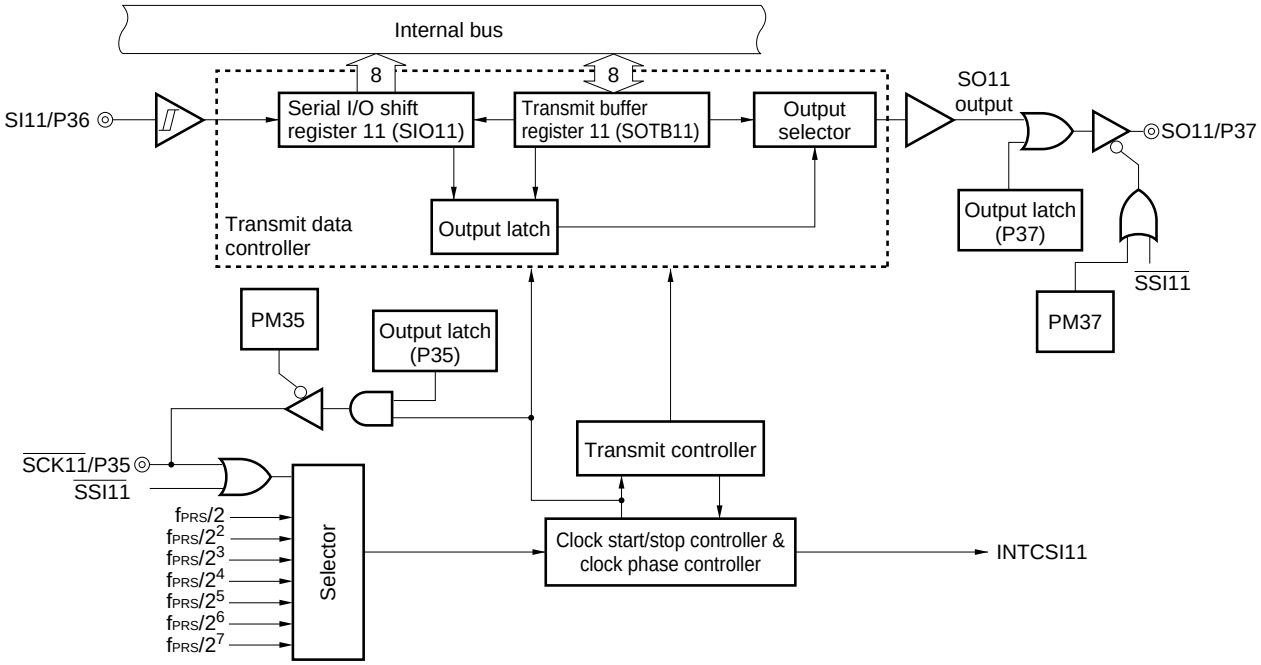
15.2 Configuration of Serial Interface CSI11

Serial interface CSI11 includes the following hardware.

Table 15-1. Configuration of Serial Interface CSI11

Item	Configuration
Controller	Transmit controller Clock start/stop controller & clock phase controller
Registers	Transmit buffer register 11 (SOTB11) Serial I/O shift register 11 (SIO11)
Control registers	Serial operation mode register 11 (CSIM11) Serial clock selection register 11 (CSIC11) Port mode registers 0 and 3 (PM0, PM3) Port register 3 (P3)

Figure 15-1. Block Diagram of Serial Interface CSI11



(1) Transmit buffer register 11 (SOTB11)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB11 when bit 7 (CSIE11) and bit 6 (TRMD11) of serial operation mode register 11 (CSIM11) is 1.

The data written to SOTB11 is converted from parallel data into serial data by serial I/O shift register 11, and output to the serial output pin (SO11).

SOTB11 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 SOTB11 when CSOT11 = 1 (during serial communication).

2. In the slave mode, 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, refer to 15.4.2 (2) Communication operation.

(2) Serial I/O shift register 11 (SIO11)

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 SIO11 if bit 6 (TRMD11) of serial operation mode register 11 (CSIM11) is 0.

During reception, the data is read from the serial input pin (SI11) to SIO11.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SIO11 when CSOT11 = 1 (during serial communication).

2. In the slave mode, 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, refer to 15.4.2 (2) Communication operation.

15.3 Registers Controlling Serial Interface CSI11

Serial interface CSI11 is controlled by the following four registers.

- Serial operation mode register 11 (CSIM11)
- Serial clock selection register 11 (CSIC11)
- Port mode registers 0 and 3 (PM0, PM3)
- Port register 3 (P3)

(1) Serial operation mode register 11 (CSIM11)

CSIM11 is used to select the operation mode and enable or disable operation.

CSIM11 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-2. 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	DIR11	0	0	0	CSOT11

CSIE11	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

TRMD11[Note 4]	Transmit/receive mode control
0[Note 5]	Receive mode (transmission disabled).
1	Transmit/receive mode

SSE11[Notes 6, 7]	$\overline{\text{SSI11}}$ pin use selection
0	$\overline{\text{SSI11}}$ pin is not used
1	$\overline{\text{SSI11}}$ pin is used

DIR11[Note 8]	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. To use P37/SO11, P35/$\overline{\text{SCK11}}$, and P20/$\overline{\text{SSI11}}$/INTP5 as general-purpose ports, set CSIM11 in the default status (00H).

3. Bit 0 (CSOT11) of CSIM11 and serial I/O shift register 11 (SIO11) are reset.

4. Do not rewrite TRMD11 when CSOT11 = 1 (during serial communication).

5. The SO11 output (refer to **Figure 15-1**) is fixed to the low level when TRMD11 is 0. Reception is started when data is read from SIO11.

6. Do not rewrite SSE11 when CSOT11 = 1 (during serial communication).

7. Before setting this bit to 1, fix the $\overline{\text{SSI11}}$ pin input level to 0 or 1.

8. Do not rewrite DIR11 when CSOT11 = 1 (during serial communication).

(2) Serial clock selection register 11 (CSIC11)

This register specifies the timing of the data transmission/reception and sets the serial clock.

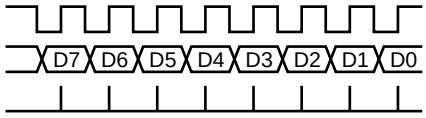
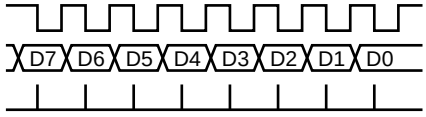
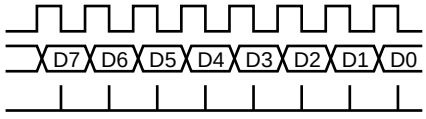
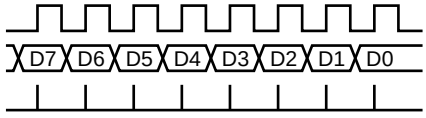
CSIC11 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-3. 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					Mode
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz (when using PLL)	
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	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					Slave mode

Note Do not start communication with the external clock from the $\overline{SCK11}$ pin when in the STOP mode.

Cautions 1. Do not write to CSIC11 while CSIE11 = 1 (operation enabled).

2. To use P37/SO11 and P35/$\overline{SCK11}$ as general-purpose ports, set CSIC11 in the default status (00H).

3. The phase type of the data clock is type 1 after reset.

Remark f_{PRS}: Peripheral hardware clock frequency

(3) Port mode registers 0 and 3 (PM0, PM3)

These registers set input/output of ports 0 and 3 in 1-bit units.

When using P35/$\overline{SCK11}$ as the clock output pin of the serial interface, clear PM35 to 0, and set the output latches of P35 to 1.

When using P37/SO11 as the data output pin of the serial interface, clear PM37 and the output latches of P37 to 0.

When using P35/$\overline{SCK11}$ as the clock input pin of the serial interface, P36/SI11 as the data input pin of the serial interface, and P02/$\overline{SSI11}$/INTP5 as the chip select input pin of the serial interface, set PM35, PM36, and PM02 to 1. At this time, the output latches of P35, P36, and P02 may be 0 or 1.

PM0 and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 15-4. Format of Port Mode Register 0 (PM0) (78K0/FB2-L)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	PM02	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 15-5. Format of Port Mode Register 3 (PM3) (78K0/FB2-L)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

15.4 Operation of Serial Interface CSI11

Serial interface 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 $\overline{SCK11}$, SI11, SO11, and $\overline{SSI11}$ pins can be used as ordinary I/O port pins in this mode.

(1) Register used

The operation stop mode is set by serial operation mode register 11 (CSIM11).

To set the operation stop mode, clear bit 7 (CSIE11) of CSIM11 to 0.

(a) Serial operation mode register 11 (CSIM11)

CSIM11 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CSIM11 to 00H.

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 P37/SO11, P35/$\overline{SCK11}$, and P02/$\overline{SSI11}$/INTP5 as general-purpose ports set CSIM11 in the default status (00H).

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 ($\overline{SCK11}$), serial output (SO11), and serial input (SI11) lines.

(1) Registers used

- Serial operation mode register 11 (CSIM11)
- Serial clock selection register 11 (CSIC11)
- Port mode registers 0 and 3 (PM0, PM3)
- Port register 3 (P3)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC11 register (refer to **Figure 15-3**).
<2> Set bits 4 to 6 (DIR11, SSE11, and TRMD11) of the CSIM11 register (refer to **Figure 15-2**).
<3> Set bit 7 (CSIE11) of the CSIM11 register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 11 (SOTB11). → Data transmission/reception is started.
Read data from serial I/O shift register 11 (SIO11). → Data reception 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 15-2. Relationship Between Register Settings and Pins

CSIE11	TRMD11	SSE11	PM36	P36	PM37	P37	PM35	P35	PM02	P02	CSI11 Operation	Pin Function			
												SI11/P36	SO11/P37	$\overline{\text{SCK11}}$/P35	$\overline{\text{SSI11}}$/P02/INTP5
0	0	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 2	×Note 2	Stop	P36	P37	P35Note 4	P02/INTP5
1	0	0	1	×	×Note 1	×Note 1	1	×	×Note 2	×Note 2	Slave reception Note 3	SI11	P37	$\overline{\text{SCK11}}$ (input) Note 3	P02/INTP5
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	×Note 1	×Note 1	0	0	1	×	×Note 2	×Note 2	Slave transmission Note 3	P36	SO11	$\overline{\text{SCK11}}$ (input) Note 3	P02/INTP5
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	1	×	0	0	1	×	×Note 2	×Note 2	Slave transmission/ reception Note 3	SI11	SO11	$\overline{\text{SCK11}}$ (input) Note 3	P02/INTP5
		1							1	×					$\overline{\text{SSI11}}$
1	0	0	1	×	×Note 1	×Note 1	0	1	×Note 2	×Note 2	Master reception	SI11	P37	$\overline{\text{SCK11}}$ (output)	P02/INTP5
1	1	0	×Note 1	×Note 1	0	0	0	1	×Note 2	×Note 2	Master transmission	P36	SO11	$\overline{\text{SCK11}}$ (output)	P02/INTP5
1	1	0	1	×	0	0	0	1	×Note 2	×Note 2	Master transmission/ reception	SI11	SO11	$\overline{\text{SCK11}}$ (output)	P02/INTP5

Notes 1. Can be set as port function.

2. Can be set as port function or external interrupt function.

3. To use the slave mode, set CKS112, CKS111, and CKS110 to 1, 1, 1.

4. To use P35/$\overline{\text{SCK11}}$ as port pin, clear CKP11 to 0.

Remark ×: don't care

CSIE11: Bit 7 of serial operation mode register 11 (CSIM11)

TRMD11: Bit 6 of CSIM11

SSE11: Bit 5 of CSIM11

CKP11: Bit 4 of serial clock selection register 11 (CSIC11)

CKS112, CKS111, CKS110: Bits 2 to 0 of CSIC11

PM0, PM3: Port mode registers 0 and 3

P0, P3: Output latch of ports 0 and 3

(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 (TRMD11) of serial operation mode register 11 (CSIM11) is 1. Transmission/reception is started when a value is written to transmit buffer register 11 (SOTB11). In addition, data can be received when bit 6 (TRMD11) of serial operation mode register 11 (CSIM11) is 0.

Reception is started when data is read from serial I/O shift register 11 (SIO11).

However, communication is performed as follows if bit 5 (SSE11) of CSIM11 is 1 when serial interface CSI11 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.

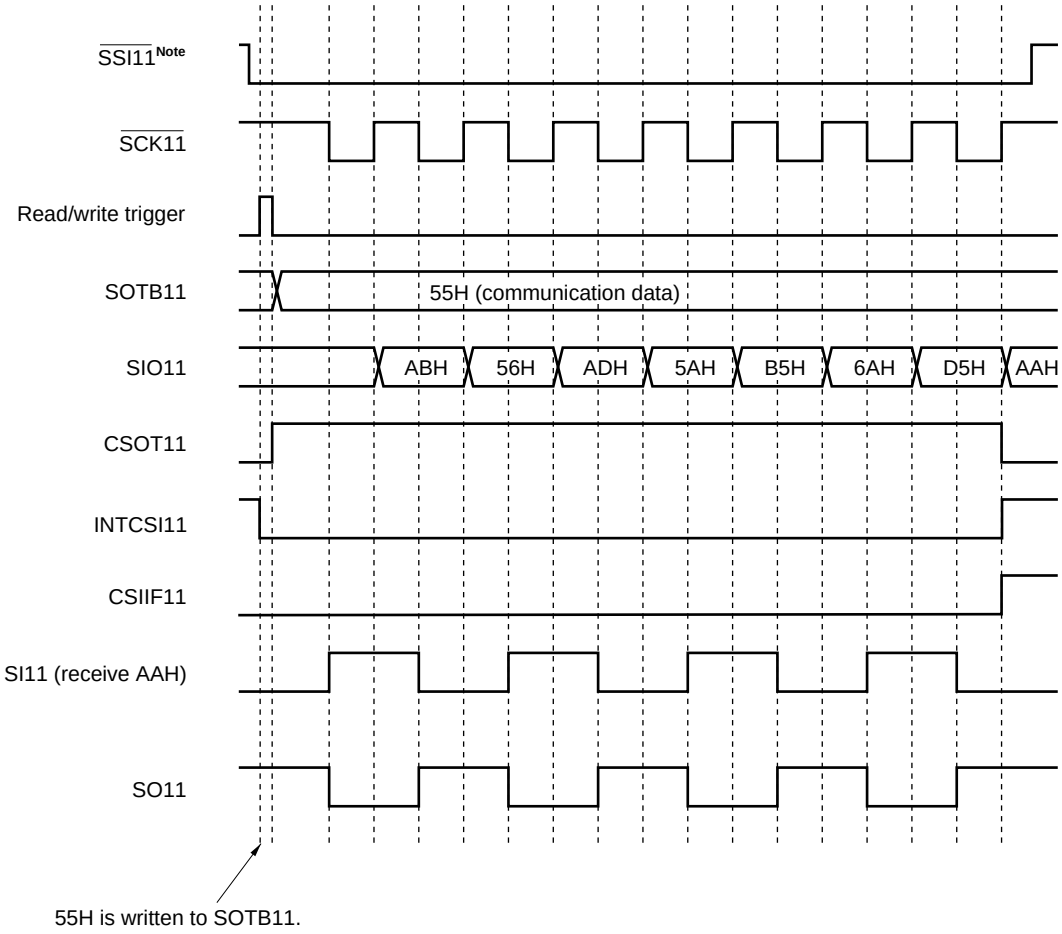
After communication has been started, bit 0 (CSOT11) of CSIM11 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF11) is set, and CSOT11 is cleared to 0. Then the next communication is enabled.

Cautions 1. Do not access the control register and data register when CSOT11 = 1 (during serial communication).

2. 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.

Figure 15-6. Timing in 3-Wire Serial I/O Mode (1/2)

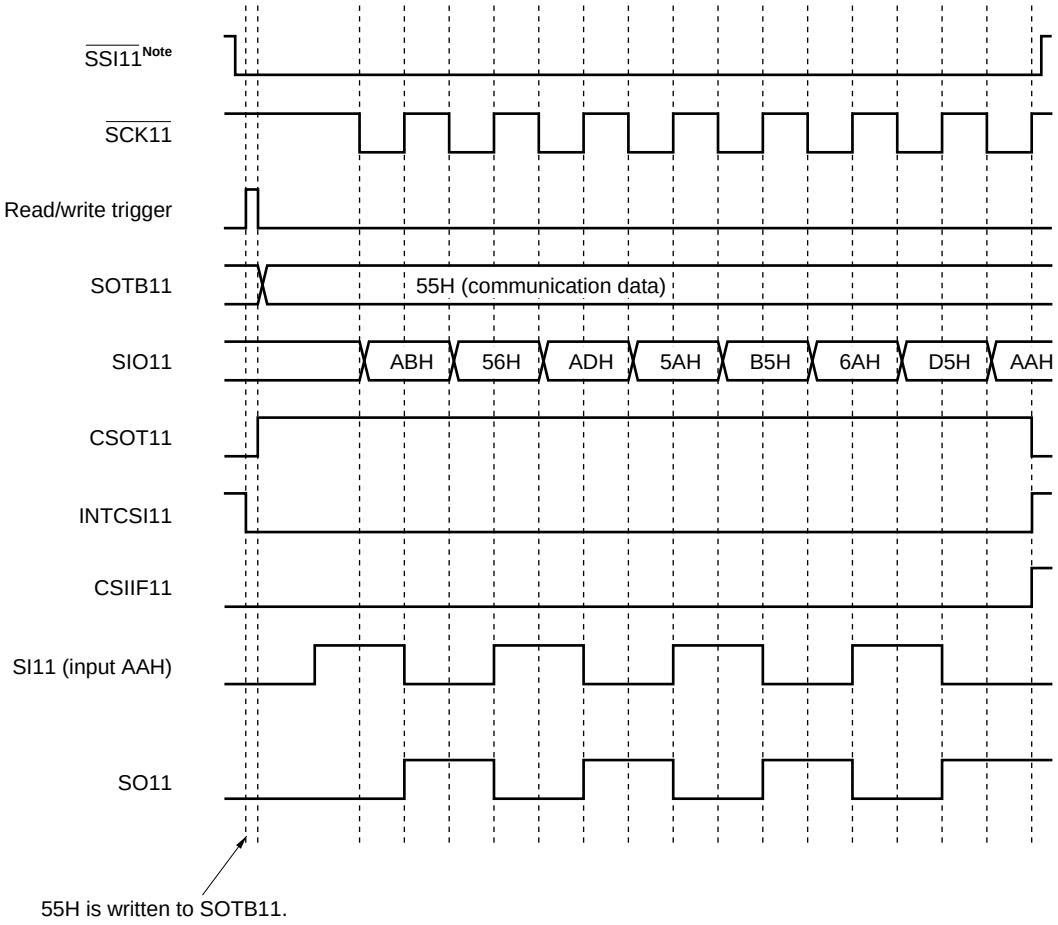
(a) Transmission/reception timing (Type 1: TRMD11 = 1, DIR11 = 0, CKP11 = 0, DAP11 = 0, SSE11 = 1[Note])



Note The SSE11 flag and SSI11 pins are used in the slave mode.

Figure 15-6. Timing in 3-Wire Serial I/O Mode (2/2)

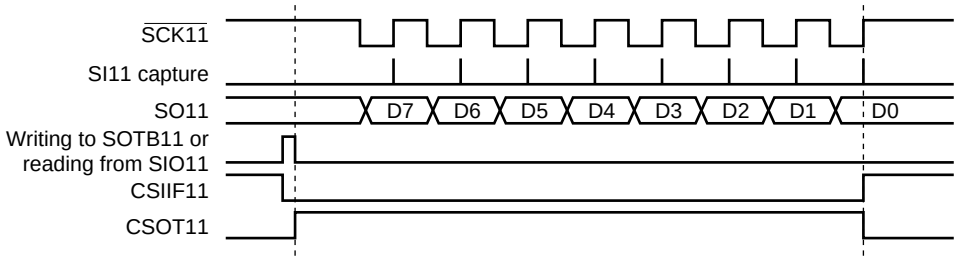
(b) Transmission/reception timing (Type 2: TRMD11 = 1, DIR11 = 0, CKP11 = 0, DAP11 = 1, SSE11 = 1[Note])



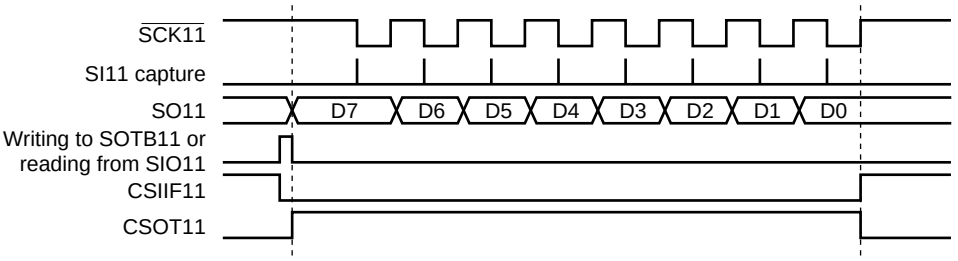
Note The SSE11 flag and $\overline{\text{SSI11}}$ pins are used in the slave mode.

Figure 15-7. Timing of Clock/Data Phase

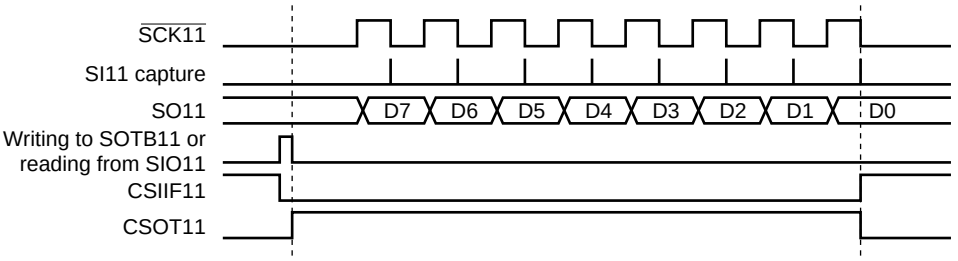
(a) Type 1: CKP11 = 0, DAP11 = 0, DIR11 = 0



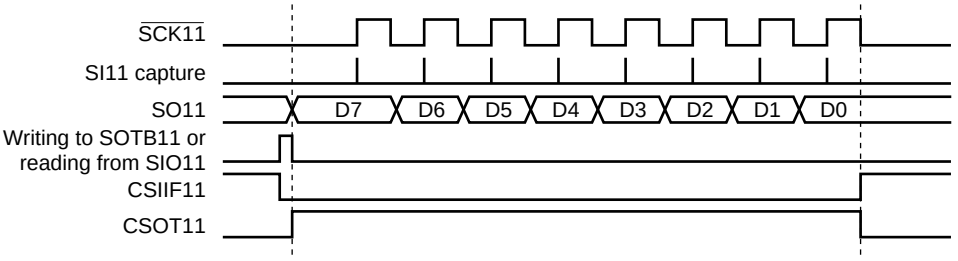
(b) Type 2: CKP11 = 0, DAP11 = 1, DIR11 = 0



(c) Type 3: CKP11 = 1, DAP11 = 0, DIR11 = 0



(d) Type 4: CKP11 = 1, DAP11 = 1, DIR11 = 0



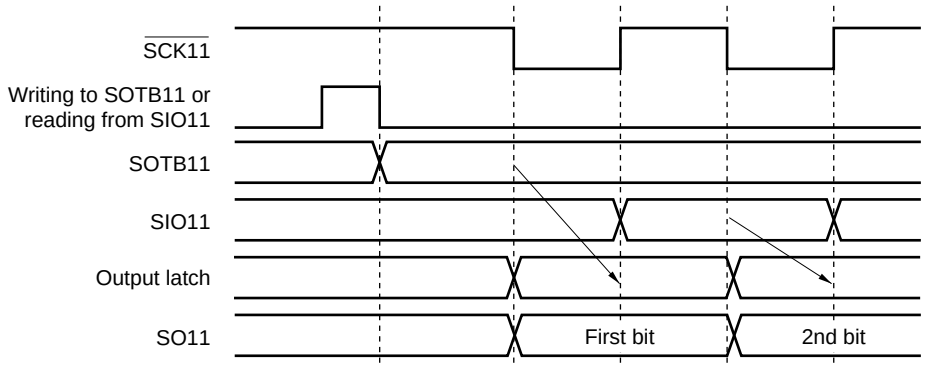
Remark The above figure illustrates a communication operation where data is transmitted with the MSB first.

(3) Timing of output to SO11 pin (first bit)

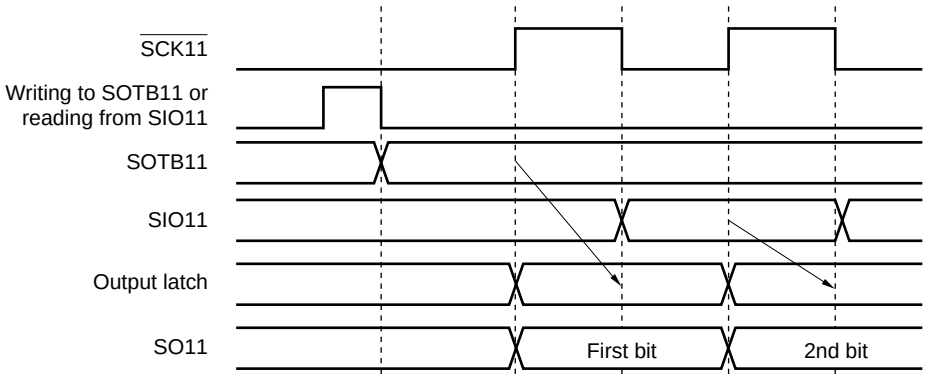
When communication is started, the value of transmit buffer register 11 (SOTB11) is output from the SO11 pin. The output operation of the first bit at this time is described below.

Figure 15-8. Output Operation of First Bit (1/2)

(a) Type 1: CKP11 = 0, DAP11 = 0



(b) Type 3: CKP11 = 1, DAP11 = 0

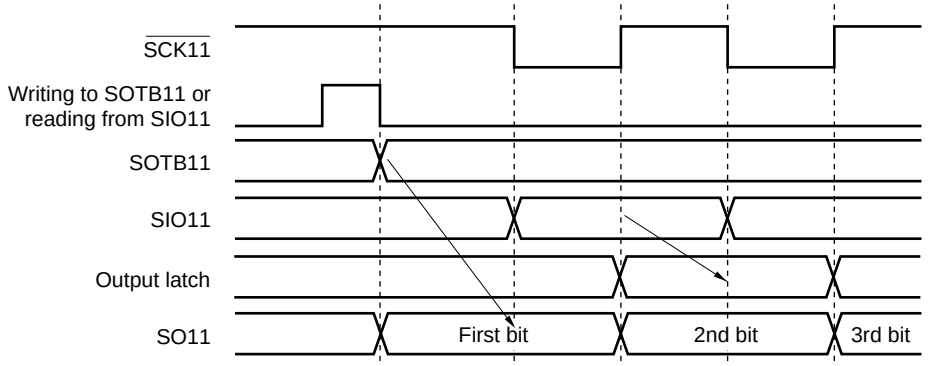


The first bit is directly latched by the SOTB11 register to the output latch at the falling (or rising) edge of $\overline{\text{SCK11}}$, and output from the SO11 pin via an output selector. Then, the value of the SOTB11 register is transferred to the SIO11 register at the next rising (or falling) edge of $\overline{\text{SCK11}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO11 register via the SI11 pin.

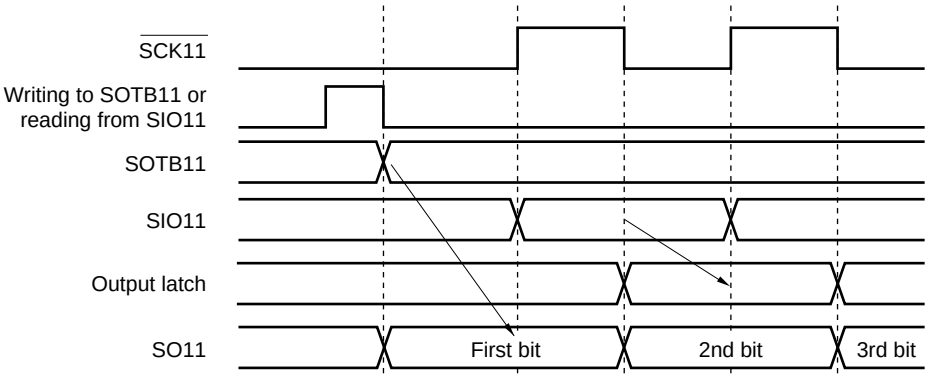
The second and subsequent bits are latched by the SIO11 register to the output latch at the next falling (or rising) edge of $\overline{\text{SCK11}}$, and the data is output from the SO11 pin.

Figure 15-8. Output Operation of First Bit (2/2)

(c) Type 2: CKP11 = 0, DAP11 = 1



(d) Type 4: CKP11 = 1, DAP11 = 1



The first bit is directly latched by the SOTB11 register at the falling edge of the write signal of the SOTB11 register or the read signal of the SIO11 register, and output from the SO11 pin via an output selector. Then, the value of the SOTB11 register is transferred to the SIO11 register at the next falling (or rising) edge of $\overline{\text{SCK11}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO11 register via the SI11 pin.

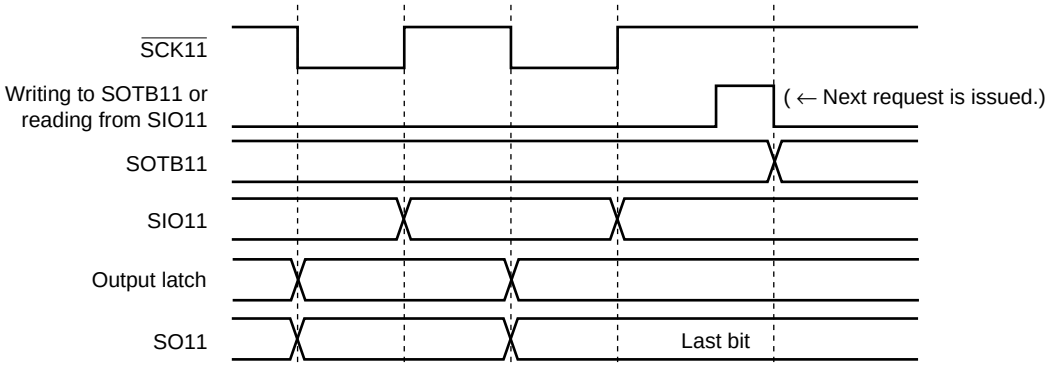
The second and subsequent bits are latched by the SIO11 register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK11}}$, and the data is output from the SO11 pin.

(4) Output value of SO11 pin (last bit)

After communication has been completed, the SO11 pin holds the output value of the last bit.

Figure 15-9. Output Value of SO11 Pin (Last Bit) (1/2)

(a) Type 1: CKP11 = 0, DAP11 = 0



(b) Type 3: CKP11 = 1, DAP11 = 0

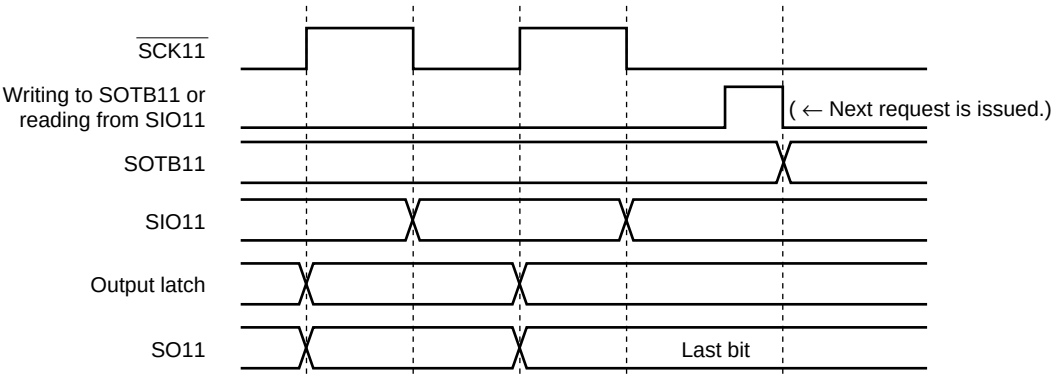
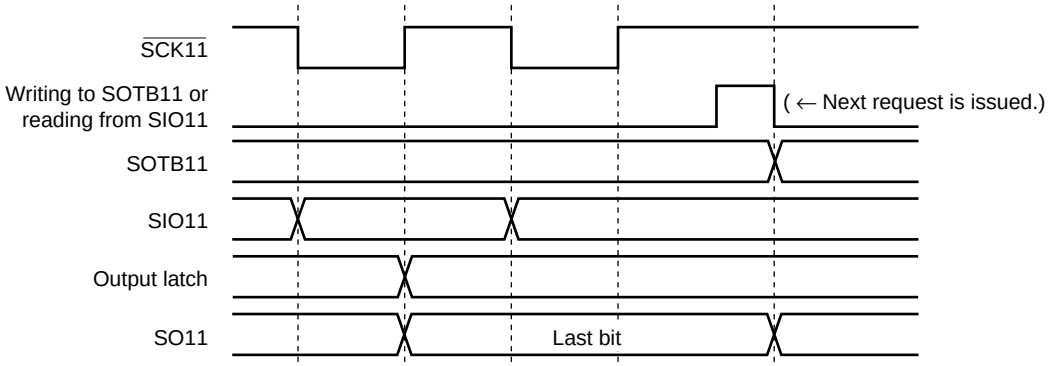
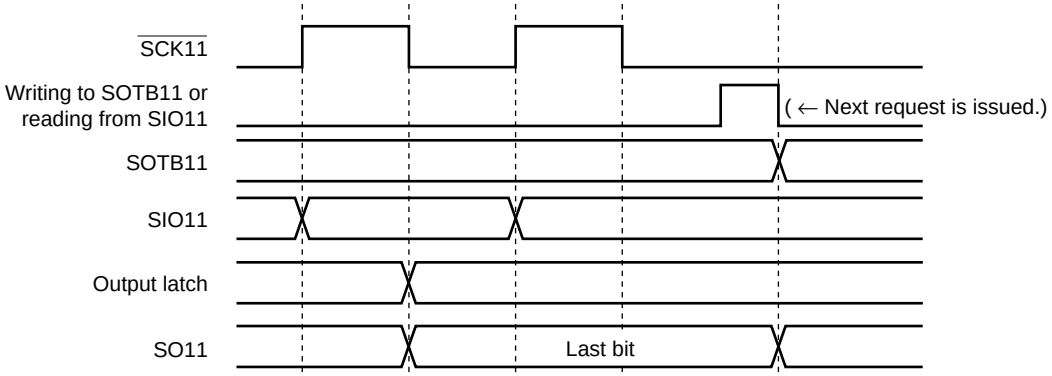


Figure 15-9. Output Value of SO11 Pin (Last Bit) (2/2)

(c) Type 2: CKP11 = 0, DAP11 = 1



(d) Type 4: CKP11 = 1, DAP11 = 1



(5) SO11 output (refer to Figure 15-1)

The status of the SO11 output is as follows depending on the setting of CSIE11, TRMD11, DAP11, and DIR11.

Table 15-3. SO11 Output Status

CSIE11	TRMD11	DAP11	DIR11	SO11 Output[Note 1]
CSIE11 = 0 [Note 2]	TRMD11 = 0[Note 2]	–	–	Low level output[Note 2]
	TRMD11 = 1 [Note 3]	DAP11 = 0	–	Low level output
		DAP11 = 1	DIR11 = 0	Value of bit 7 of SOTB11
			DIR11 = 1	Value of bit 0 of SOTB11
CSIE11 = 1	TRMD11 = 0	–	–	Low level output
	TRMD11 = 1	–	–	Transmission data [Note 4]

Notes 1. The actual output of the SO11 pin is determined according to PM37 and P37, as well as the SO11 output.

2. This is a status after reset.

3. To use SO11/P37 as general-purpose port, set CSIC11 in the default status (00H).

4. After transmission has been completed, the SO11 pin holds the output value of the last bit of transmission data.

Caution If a value is written to CSIE11, TRMD11, DAP11, and DIR11, the output value of SO11 changes.

CHAPTER 16 MULTIPLIER

16.1 Functions of Multiplier

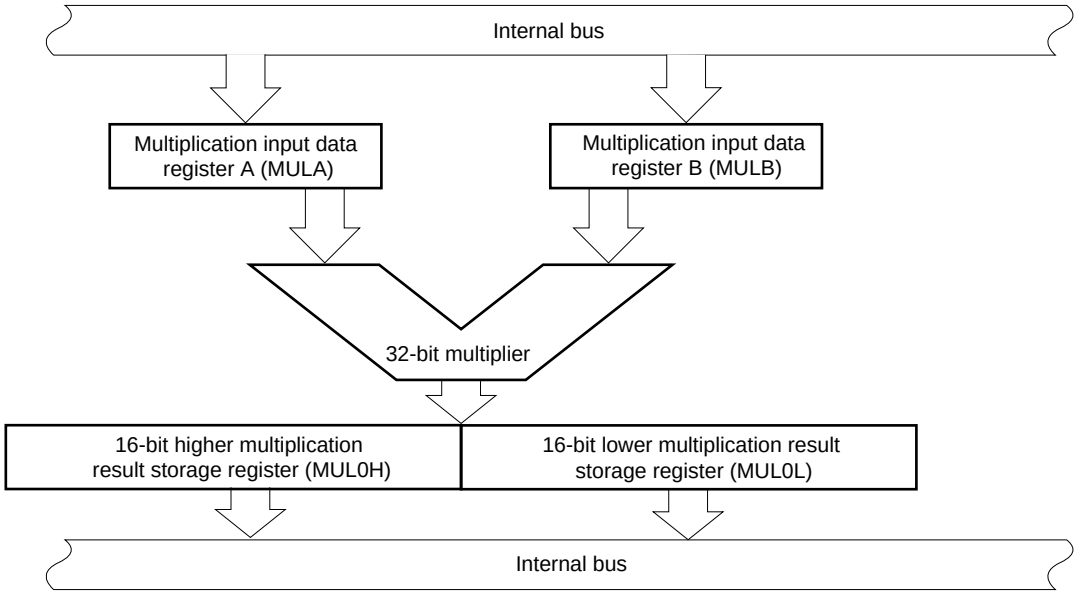
The multiplier is mounted onto all 78K0/Fx2-L microcontroller products.

The multiplier has the following functions.

- Can execute calculation of 8 bits × 8 bits = 16 bits.
- Can execute calculation of 16 bits × 16 bits = 32 bits.

Figure 16-1 shows the block diagram of the multiplier.

Figure 16-1. Block Diagram of Multiplier



16.2 Configuration of Multiplier

(1) 16-bit higher multiplication result storage register and 16-bit lower multiplication result storage register (MUL0H, MUL0L)

These two registers, MUL0H and MUL0L, are used to store a 32-bit multiplication result.

In the case of multiplication of 8 bits by 8 bits, the 16 bits of the multiplication result are stored in MUL0L.

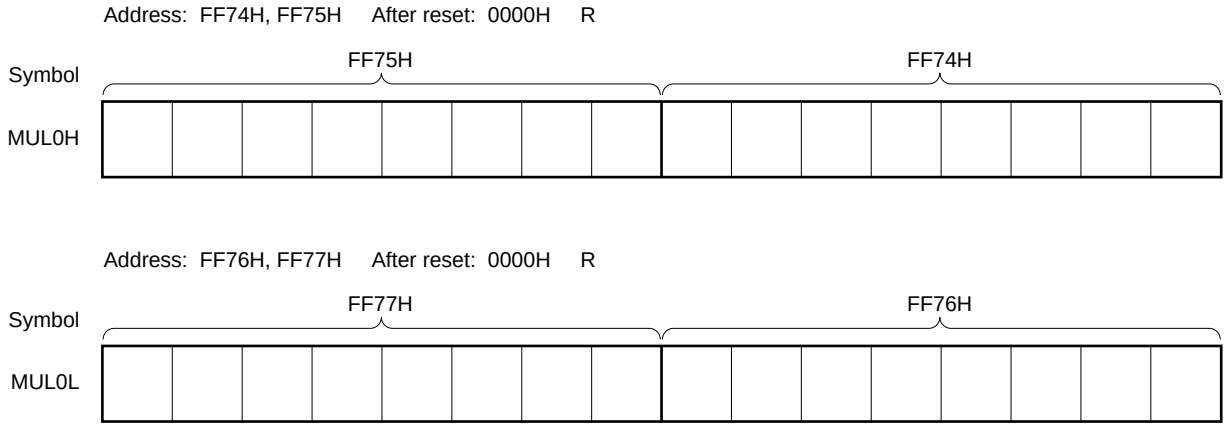
In the case of multiplication of 16 bits by 16 bits, the higher 16 bits of the multiplication result are stored in MUL0H and the lower 16 bits, in MUL0L, so that a total of 32 bits of the multiplication result can be stored.

These registers hold the result of multiplication after the lapse of one CPU clock.

MUL0H and MUL0L can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-2. Format of 16-bit Higher Multiplication Result Storage Register and 16-bit Lower Multiplication Result Storage Register (MUL0H, MUL0L)



(2) Multiplication input data registers A, B (MULA, MULB)

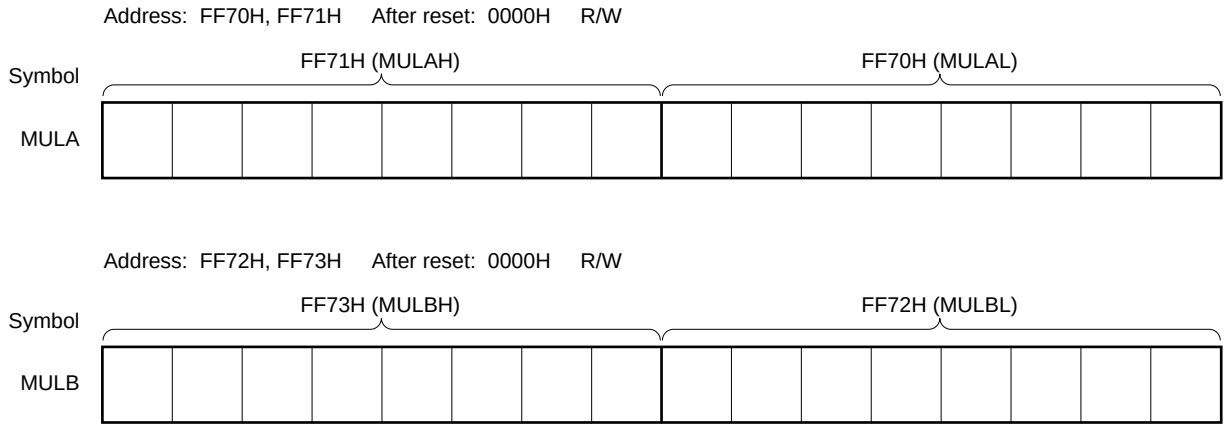
These are 16-bit registers that store data for multiplication. The multiplier multiplies the values of MULA and MULB.

MULA and MULB can be set by an 8-bit or 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

Caution In the case of multiplication of 8 bits by 8 bits, set the multiplied data to MULAL and MULBL.

Figure 16-3. Format of Multiplication Input Data Registers A, B (MULA, MULB)



16.3 Operation of Multiplier

The result of the multiplication can be obtained by storing the values in the MULA and MULB registers and then reading the MUL0H and MUL0L registers after waiting for 1 clock. The result can also be obtained after 1 clock or more has elapsed, even when fixing either of MULA or MULB and rewrite the other of these. The result can be read without problem, regardless of whether MUL0H or MUL0L is read in first.

A multiplication source example is shown below.

Example 1: Multiplication of 8 bits by 8 bits

```
MOV     MULAL, #005H
MOV     MULBL, #022H
NOP                         ; 1 clock wait.  Doesn't have to be NOP
MOVW    AX, MUL0L           ; Acquire multiplication result
```

Caution In Example 1, set the multiplied data to MULAL and MULBL.

Example 2: Multiplication of 16 bits by 16 bits (using the MOVW instruction for setting the multiplication data)

```
MOVW    MULA, #1234H
MOVW    MULB, #5678H
NOP                         ; 1 clock wait.  Doesn't have to be NOP
MOVW    AX, MUL0H           ; The result obtained on upper side
PUSH    AX
MOVW    AX, MUL0L           ; The result obtained on lower side
```

Example 3: Multiplication of 16 bits by 16 bits (using the MOV instruction for setting the multiplication data)

```
MOV     MULAL, #034H
MOV     MULAH, #012H
MOV     MULBL, #078H
MOV     MULBH, #056H
NOP                         ; 1 clock wait.  Doesn't have to be NOP
MOVW    AX, MUL0H           ; The result obtained on upper side
PUSH    AX
MOVW    AX, MUL0L           ; The result obtained on lower side
```

Caution In Example 3, set the higher 8 bits of the multiplied data to MULAH/MULBH after setting the lower 8 bits to MULAL/MULBL. When the higher 8 bits of the multiplied data are the same value and only the lower 8 bits are to be changed, set the changed value to MULAL/MULBL and re-set the same value to MULAH/MULBH.

CHAPTER 17 INTERRUPT FUNCTIONS

Item		78K0/FY2-L (16 Pins)	78K0/FA2-L (20 Pins)	78K0/FB2-L (30 Pins)
Maskable interrupts	External	3	7	9
	Internal	11	11	13

17.1 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, refer to **Table 17-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

17.2 Interrupt Sources and Configuration

The interrupt sources consist of maskable interrupts and software interrupts. In addition, they also have up to four reset sources (refer to **Table 17-1**).

Table 17-1. Interrupt Source List (1/2)

Interrupt Type	Internal/ External	Basic Configuration Type Note 1	Default Priority Note 2	Interrupt Source		Vector Table Address	FY2-L	FA2-L	FB2-L
				Name	Trigger		16 Pins	20 Pins	30 Pins
Maskable	Internal	(A)	0	INTLVI	Low-voltage detection Note 3	0004H	√	√	√
	External	(B)	1	INTP0	Pin input edge detection	0006H	√	√	√
			2	INTP1		0008H	√	√	√
			3	INTP2		000AH	–	√	√
			4	INTP3		000CH	–	√	√
			5	INTP4		000EH	–	–	√
			6	INTP5		0010H	–	–	√
	Internal	(A)	7	INTSRE6	UART6 reception error generation	0012H	√	√	√
			8	INTSR6	End of UART6 reception	0014H	√	√	√
			9	INTST6	End of UART6 transmission	0016H	√	√	√
			10	INTCSI11	End of CSI11 communication	0018H	–	–	√
			11	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)	001AH	√	√	√
			12	INTTMX0	Match between TMX0 counter and TX0CR1 or TX0CR3 (when compare register is specified)	001CH	√	√	√
			13	INTTMX1	Match between TMX1 counter and TX1CR1 or TX1CR3 (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	INTTM51 Note 4	Match between TM51 and CR51 (when compare register is specified)	002AH	√	√	√
	External	(B)	18	INTCMP0	Comparator 0 edge detection	002CH	–	√	√
			19	INTCMP1	Comparator 1 edge detection	002EH	–	√	√
			20	INTCMP2	Comparator 2 edge detection	0030H	√	√	√
	Internal	(A)	21	INTIICA0	End of IICA communication	0034H	√	√	√

Notes 1. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 17-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 21 indicates the lowest priority.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

4. When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (refer to **Figure 9-11 Transfer Timing**).

Table 17-1. Interrupt Source List (2/2)

Interrupt Type	Internal/ External	Basic Configuration Type Note 1	Default Priority Note 2	Interrupt Source		Vector Table Address	FY2-L	FA2-L	FB2-L
				Name	Trigger		16 Pins	20 Pins	30 Pins
Software	–	(C)	–	BRK	BRK instruction execution	003EH	√	√	√
Reset	–	–	–	RESET	Reset input	0000H	√	√	√
				POC	Power-on clear				
				LVI	Low-voltage detection Note 3				
				WDT	WDT overflow				

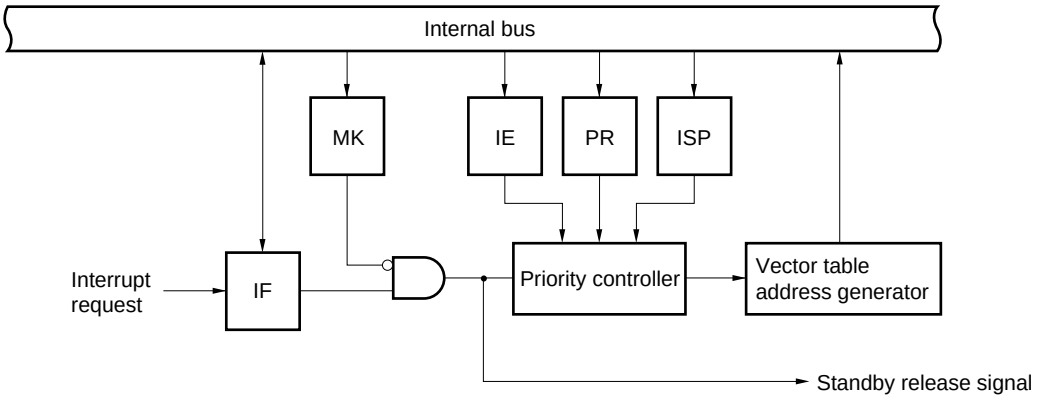
Notes 1. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 17-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 21 indicates the lowest priority.

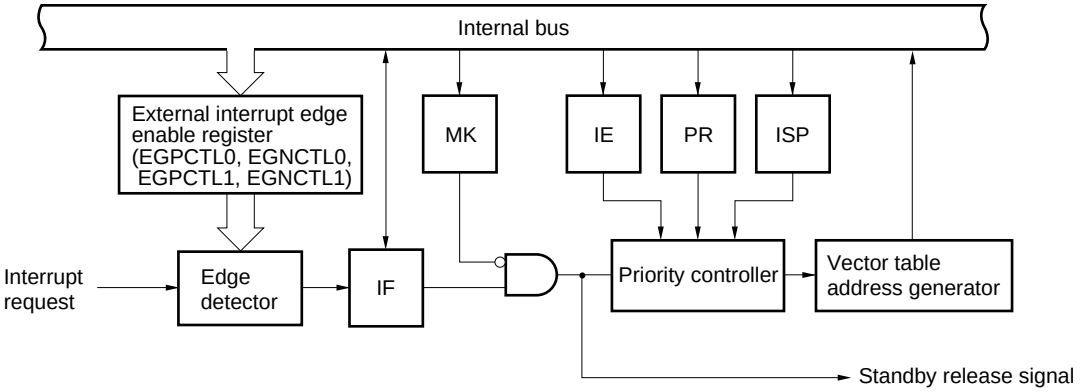
3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

Figure 17-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPm, INTCMP0 to INTCMP2)

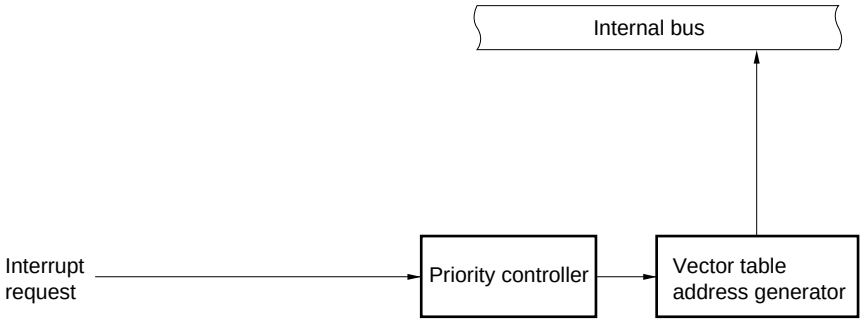


Remark m = 0, 1: 78K0/FY2-L
m = 0 to 3: 78K0/FA2-L
m = 0 to 5: 78K0/FB2-L

IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

Figure 17-1. Basic Configuration of Interrupt Function (2/2)

(C) Software interrupt



17.3 Registers Controlling Interrupt Functions

The following 7 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H)
- Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)
- Port alternate switch control registers (MUXSEL)
- External interrupt rising edge enable registers (EGPCTL0, EGPCTL1)
- External interrupt falling edge enable registers (EGNCTL0, EGNCTL1)
- Program status word (PSW)

Table 17-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 17-2. Flags Corresponding to Interrupt Request Sources

FY2-L	FA2-L	FB2-L	Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
					Register		Register		Register
16 Pins	20 Pins	30 Pins							
√	√	√	INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
√	√	√	INTP0	PIF0		PMK0		PPR0	
√	√	√	INTP1	PIF1		PMK1		PPR1	
–	√	√	INTP2	PIF2		PMK2		PPR2	
–	√	√	INTP3	PIF3		PMK3		PPR3	
–	–	√	INTP4	PIF4		PMK4		PPR4	
–	–	√	INTP5	PIF5		PMK5		PPR5	
√	√	√	INTSRE6	SREIF6		SREMK6		SREPR6	
√	√	√	INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
√	√	√	INTST6	STIF6		STMK6		STPR6	
–	–	√	INTCSI11	CSIIF11		CSIMK11		CSIPR11	
√	√	√	INTTMH1	TMIFH1		TMMKH1		TMPRH1	
√	√	√	INTTMX0	TMIFX0		TMMKX0		TMPRX0	
–	–	√	INTTMX1	TMIFX1		TMMKX1		TMPRX1	
√	√	√	INTTM000	TMIF000		TMMK000		TMPR000	
√	√	√	INTTM010	TMIF010		TMMK010		TMPR010	
√	√	√	INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
√	√	√	INTTM51[Note]	TMIF51		TMMK51		TMPR51	
–	√	√	INTCMP0	CMPIF0		CMPMK0		CMPPR0	
–	√	√	INTCMP1	CMPIF1		CMPMK1		CMPPR1	
√	√	√	INTCMP2	CMPIF2		CMPMK2		CMPPR2	
√	√	√	INTIICA0	IICAIF0	IF1H	IICAMK0	MK1H	IICAPR0	PR1H

Note When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (refer to **Figure 9-11 Transfer Timing**).

(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.

Cautions 1. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H) (78K0/FY2-L)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
IF0L	SREIF6	0	0	0	0	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
IF0H	TMIF010	TMIF000	0	TMIFX0	TMIFH1	0	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	7	<6>	5	4	<3>	2	1	<0>
IF1L	0	CMPIF2	0	0	TMIF51	0	0	ADIF

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF1H	0	0	0	0	0	0	0	IICAIF0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bits 3 to 6 of IF0L, bits 2 and 5 of IF0H, bits 1, 2, 4, 5 and 7 of IF1L, and bits 1 to 7 of IF1H to 0.

Figure 17-3. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H) (78K0/FA2-L)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	0	0	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
IF0H	TMIF010	TMIF000	0	TMIFX0	TMIFH1	0	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
IF1L	0	CMPIF2	CMPIF1	CMPIF0	TMIF51	0	0	ADIF

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF1H	0	0	0	0	0	0	0	IICAIF0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bits 5 and 6 of IF0L, bits 2 and 5 of IF0H, bits 1, 2, and 7 of IF1L, and bits 1 to 7 of IF1H to 0.

Figure 17-4. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H) (78K0/FB2-L)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIFX1	TMIFX0	TMIFH1	CSIIF11	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
IF1L	0	CMPIF2	CMPIF1	CMPIF0	TMIF51	0	0	ADIF

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF1H	0	0	0	0	0	0	0	IICAIF0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bits 1, 2, and 7 of IF1L, and bits 1 to 7 of IF1H to 0.

(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 17-5. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H) (78K0/FY2-L)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
MK0L	SREMK6	1	1	1	1	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
MK0H	TMMK010	TMMK000	1	TMMKX0	TMMKH1	1	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	7	<6>	5	4	<3>	2	1	<0>
MK1L	1	CMPMK2	1	1	TMMK51	1	1	ADMK

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
MK1H	1	1	1	1	1	1	1	IICAMK0

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 3 to 6 of MK0L, bits 2 and 5 of MK0H, bits 1, 2, 4, 5 and 7 of MK1L, and bits 1 to 7 of MK1H to 1.

Figure 17-6. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H) (78K0/FA2-L)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	1	1	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
MK0H	TMMK010	TMMK000	1	TMMKX0	TMMKH1	1	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
MK1L	1	CMPMK2	CMPMK1	CMPMK0	TMMK51	1	1	ADMK

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
MK1H	1	1	1	1	1	1	1	IICAMK0

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 5 and 6 of MK0L, bits 2 and 5 of MK0H, bits 1, 2, and 7 of MK1L, and bits 1 to 7 of MK1H to 1.

Figure 17-7. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H) (78K0/FB2-L)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMKX1	TMMKX0	TMMKH1	CSIMK11	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
MK1L	1	CMPMK2	CMPMK1	CMPMK0	TMMK51	1	1	ADMK

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
MK1H	1	1	1	1	1	1	1	IICAMK0

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 1, 2, and 7 of MK1L, and bits 1 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 17-8. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H) (78K0/FY2-L)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
PR0L	SREPR6	1	1	1	1	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
PR0H	TMPR010	TMPR000	1	TMPRX0	TMPRH1	1	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	7	<6>	5	4	<3>	2	1	<0>
PR1L	1	CMPPR2	1	1	TMPR51	1	1	ADPR

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR1H	1	1	1	1	1	1	1	IICAPR0

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bits 3 to 6 of PR0L, bits 2 and 5 of PR0H, bits 1, 2, 4, 5 and 7 of PR1L, and bits 1 to 7 of PR1H to 1.

Figure 17-9. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H) (78K0/FA2-L)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	1	1	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
PR0H	TMPR010	TMPR000	1	TMPRX0	TMPRH1	1	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
PR1L	1	CMPPR2	CMPPR1	CMPPR0	TMPR51	1	1	ADPR

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR1H	1	1	1	1	1	1	1	IICAPR0

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bits 5 and 6 of PR0L, bits 2 and 5 of PR0H, bits 1, 2, and 7 of PR1L, and bits 1 to 7 of PR1H to 1.

Figure 17-10. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H) (78K0/FB2-L)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPRX1	TMPRX0	TMPRH1	CSIPR11	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
PR1L	1	CMPPR2	CMPPR1	CMPPR0	TMPR51	1	1	ADPR

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR1H	1	1	1	1	1	1	1	IICAPR0

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bits 1, 2, and 7 of PR1L, and bits 1 to 7 of PR1H to 1.

(4) Port alternate switch control register (MUXSEL)

This register assigns the pin function.

The interrupt input (INTP0) function can be assigned to the P121 pin of the 78K0/FB2-L.

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

Reset signal generation clears MUXSEL to 00H.

Caution 78K0/FB2-L only

Figure 17-11. Format of Port Alternate Switch Control Register (MUXSEL)

Address: FF39H After reset: 00H R/W

Symbol	7	<6>	5	<4>	3	2	1	0
MUXSEL	0	INTP0SEL0	0	TM00SEL0	0	0	0	0

INTP0SEL0	External interrupt input (INTP0) pin assignment
0	(Default)
1	P121/INTP0

(5) External interrupt rising edge enable registers (EGPCTL0, EGPCTL1), external interrupt falling edge enable registers (EGNCTL0, EGNCTL1)

EGPCTL0, EGPCTL1, EGNCTL0, and EGNCTL1 are the registers that set the INTPm and INTCMP0 to INTCMP2 valid edges.

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

Reset signal generation clears these registers to 00H.

Remark m = 0, 1: 78K0/FY2-L
m = 0 to 3: 78K0/FA2-L
m = 0 to 5: 78K0/FB2-L

Figure 17-12. Format of External Interrupt Rising Edge Enable Registers (EGPCTL0, EGPCTL1) and External Interrupt Falling Edge Enable Registers (EGNCTL0, EGNCTL1) (1/3)

(a) 78K0/FY2-L

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL0	0	0	0	0	0	0	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL0	0	0	0	0	0	0	EGN1	EGN0

Address: FF4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL1	0	0	0	0	0	0	0	EGP8

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL1	0	0	0	0	0	0	0	EGN8

EGPn	EGNn	INTPm and INTCMP2 valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution Be sure to clear bits 2 to 7 of EGPCTL0 and EGNCTL0 to 0 in the 78K0/FY2-L.

Remark n = 0, 1, 8
m = 0, 1

Figure 17-12. Format of External Interrupt Rising Edge Enable Registers (EGPCTL0, EGPCTL1) and External Interrupt Falling Edge Enable Registers (EGNCTL0, EGNCTL1) (2/3)

(b) 78K0/FA2-L

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL0	EGP7	EGP6	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL0	EGN7	EGN6	0	0	EGN3	EGN2	EGN1	EGN0

Address: FF4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL1	0	0	0	0	0	0	0	EGP8

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL1	0	0	0	0	0	0	0	EGN8

EGPn	EGNn	INTPm and INTCMP0 to INTCMP2 valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution Be sure to clear bits 4 and 5 of EGPCTL0 and EGNCTL0 to 0 in the 78K0/FA2-L.

Remark n = 0-3, 6-8
m = 0-3

Figure 17-12. Format of External Interrupt Rising Edge Enable Registers (EGPCTL0, EGPCTL1) and External Interrupt Falling Edge Enable Registers (EGNCTL0, EGNCTL1) (3/3)

(c) 78K0/FB2-L

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FF4AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGPCTL1	0	0	0	0	0	0	0	EGP8

Address: FF4BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGNCTL1	0	0	0	0	0	0	0	EGN8

EGPn	EGNn	INTPm and INTCMP0 to INTCMP2 valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution Be sure to clear bits 1 to 7 of EGPCTL1 and EGNCTL1 to 0 in the 78K0/FB2-L.

Remark n = 0 to 8
m = 0 to 5

Table 17-3 shows the ports corresponding to EGPn and EGNn.

Table 17-3. Ports Corresponding to EGPn and EGNn

(a) 78K0/FY2-L

Detection Enable Bit		Edge Detection Signal	Interrupt Request Signal
EGP0	EGN0	P00 pin input	INTP0
EGP1	EGN1	P30 pin input	INTP1
EGP8	EGN8	CMP2 output	INTCMP2

(b) 78K0/FA2-L

Detection Enable Bit		Edge Detection Signal	Interrupt Request Signal
EGP0	EGN0	P00 pin input	INTP0
EGP1	EGN1	P30 pin input	INTP1
EGP2	EGN2	P31 pin input	INTP2
EGP3	EGN3	P32 pin input	INTP3
EGP6	EGN6	CMP0 output	INTCMP0
EGP7	EGN7	CMP1 output	INTCMP1
EGP8	EGN8	CMP2 output	INTCMP2

(c) 78K0/FB2-L

Detection Enable Bit		Edge Detection Signal	Interrupt Request Signal
EGP0	EGN0	P00 or P121 pin input**Note**	INTP0
EGP1	EGN1	P30 pin input	INTP1
EGP2	EGN2	P31 pin input	INTP2
EGP3	EGN3	P32 pin input	INTP3
EGP4	EGN4	P34 pin input	INTP4
EGP5	EGN5	P02 pin input	INTP5
EGP6	EGN6	CMP0 output	INTCMP0
EGP7	EGN7	CMP1 output	INTCMP1
EGP8	EGN8	CMP2 output	INTCMP2

Note The pin functions can be assigned by setting the port alternate switch control register (MUXSEL).

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, 1, 8: 78K0/FY2-L
n = 0 to 3, 6 to 8: 78K0/FA2-L
n = 0 to 8: 78K0/FB2-L

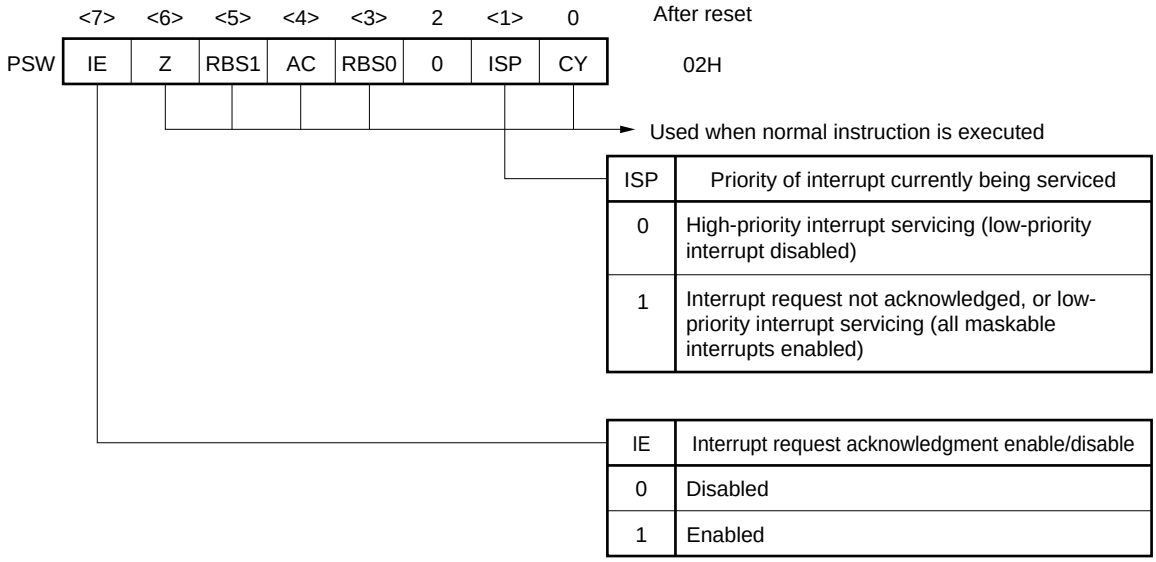
(5) Program status word (PSW)

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 17-13. 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

17.4 Interrupt Servicing Operations

17.4.1 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 17-4 below.

For the interrupt request acknowledgment timing, refer to **Figures 17-15** and **17-16**.

Table 17-4. Time from Generation of Maskable Interrupt Request 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.

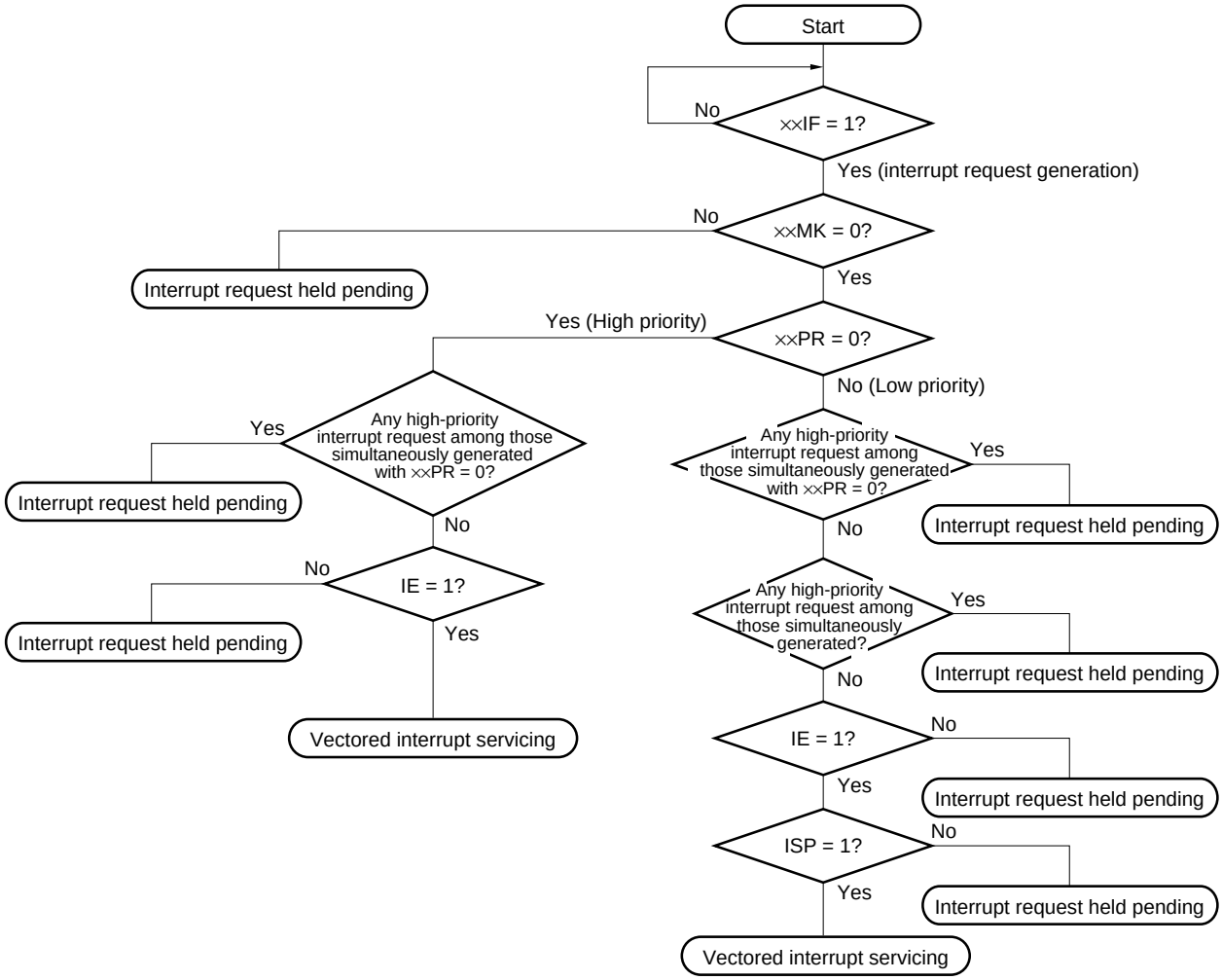
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 17-14 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.

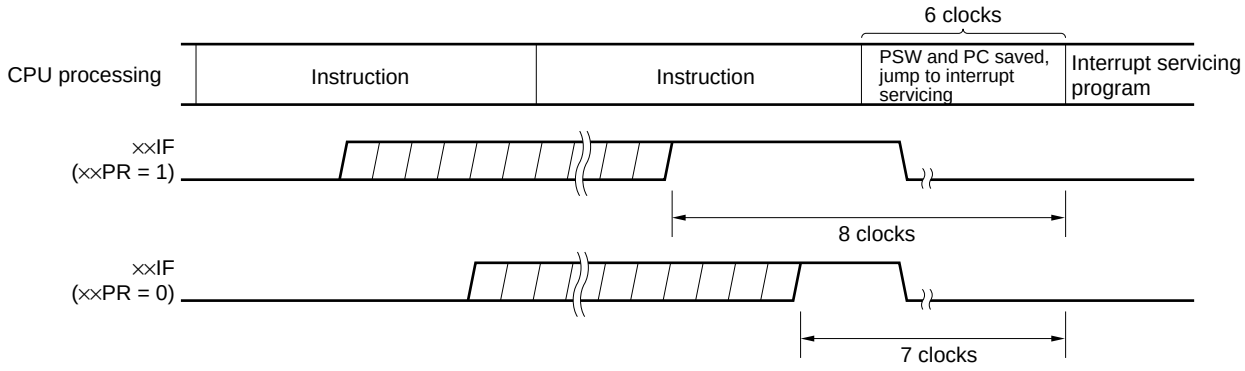
Restoring from an interrupt is possible by using the RETI instruction.

Figure 17-14. Interrupt Request Acknowledgment Processing Algorithm



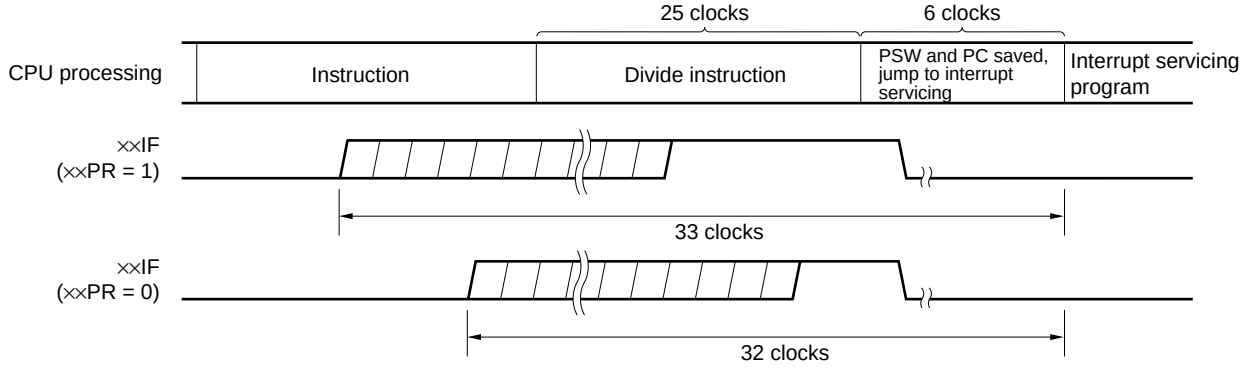
××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR: Priority specification flag
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 17-15. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 17-16. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

17.4.2 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.

17.4.3 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 17-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 17-17 shows multiple interrupt servicing examples.

Table 17-5. 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.

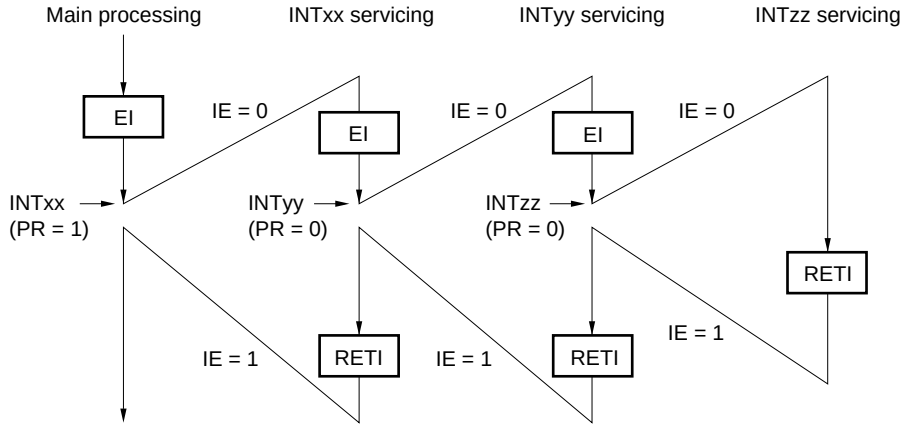
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

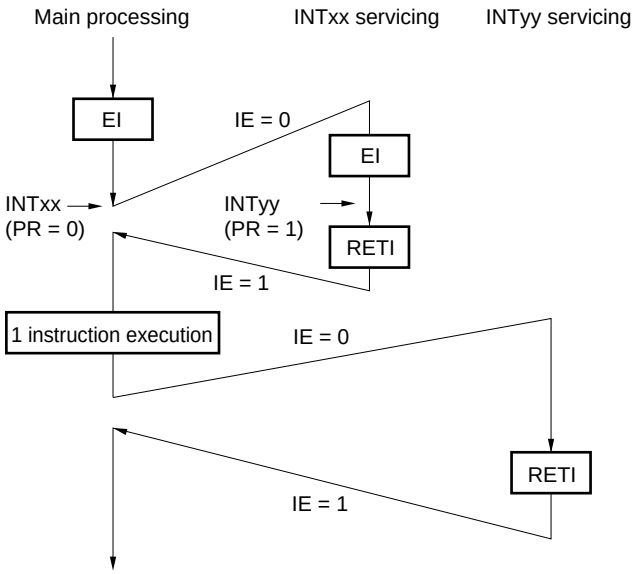
Figure 17-17. Examples of Multiple Interrupt Servicing (1/2)

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

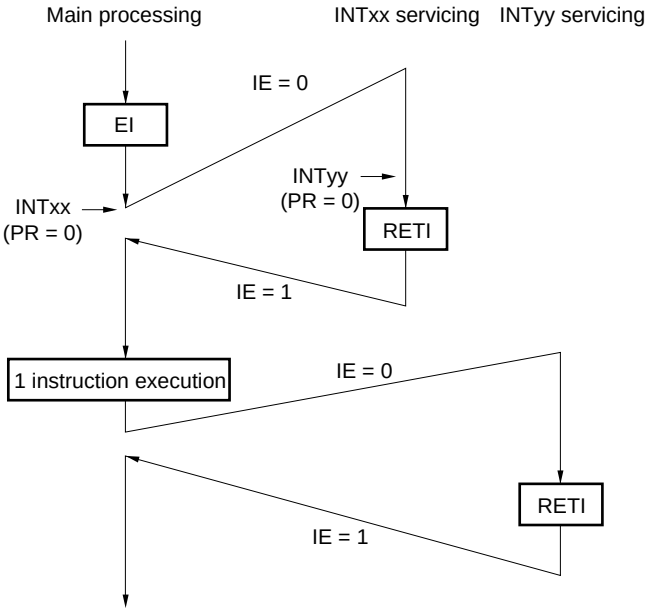


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 17-17. 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

17.4.4 Interrupt request hold

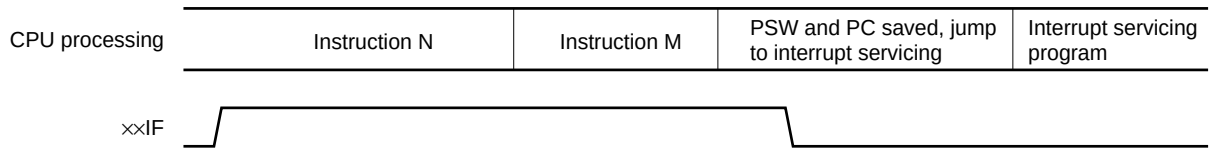
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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 17-18 shows the timing at which interrupt requests are held pending.

Figure 17-18. 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 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 Standby function

The standby function is mounted onto all 78K0/Fx2-L microcontroller products.

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, or internal low-speed 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. **When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.**
2. **When transitioning to the STOP mode, it is possible to achieve low power consumption by setting RMC = 56H.**
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 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.**
4. **Stop the comparator before executing the STOP instruction.**

18.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, refer to **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 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 18-1. 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.

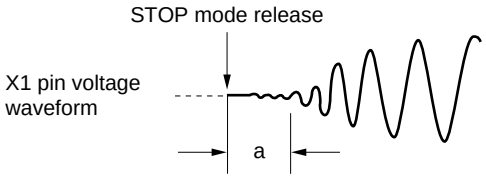
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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 18-2. 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		
				f_X = 10 MHz	f_X = 20 MHz
0	0	1	$2^{11}/f_X$	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_X$	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_X$	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_X$	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_X$	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.

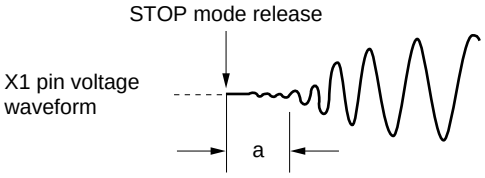
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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 f_X: X1 clock oscillation frequency

18.2 Standby Function Operation

18.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock or internal high-speed oscillation clock.

The operating statuses in the HALT mode are shown below.

Table 18-1. Operating Statuses in HALT Mode

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{IH})	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_{IH}	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)
	f_{IL}		Status before HALT mode was set is retained		
PLL					
CPU			Operation stopped		
Flash memory					
RAM			Status before HALT mode was set is retained		
Port (latch)					
16-bit timer	X0		Operable		
	X1				
16-bit timer/event counter 00					
8-bit timer/event counter 51					
8-bit timer H1					
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
A/D converter			Operable		
Comparators 0 to 2					
Serial interface	UART6				
	CSI11				
	IICA				
Multiplier			Operation stopped		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock, f_{IL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

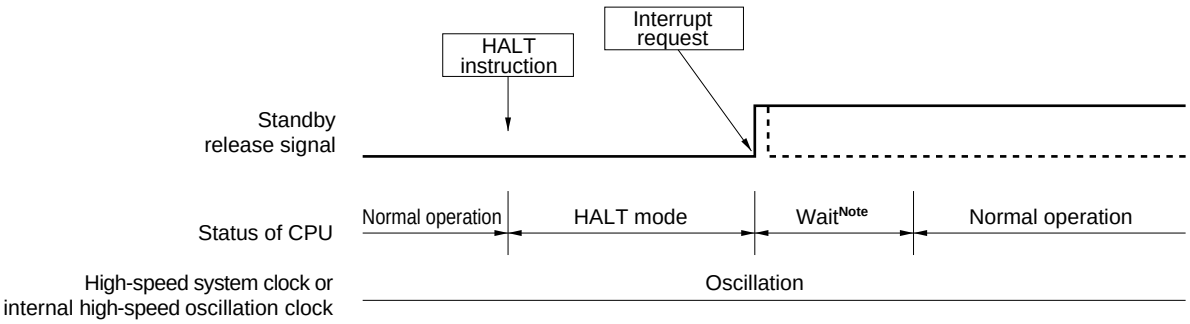
(2) HALT mode release

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 18-3. 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

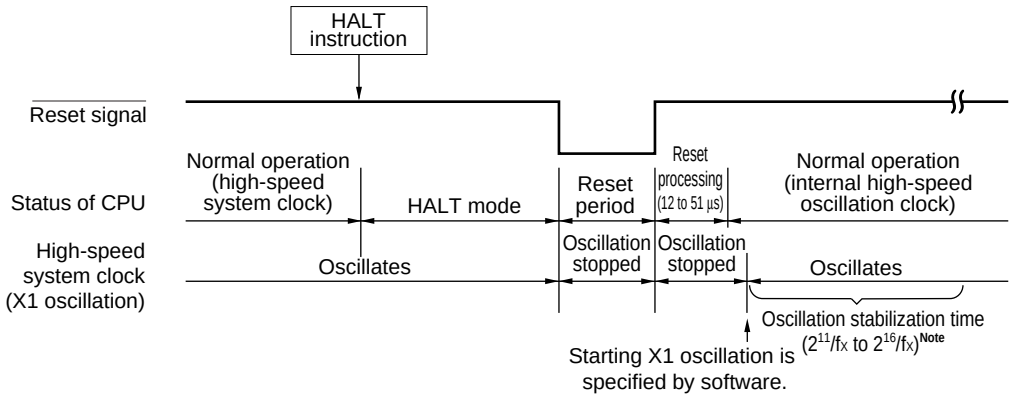
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(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.

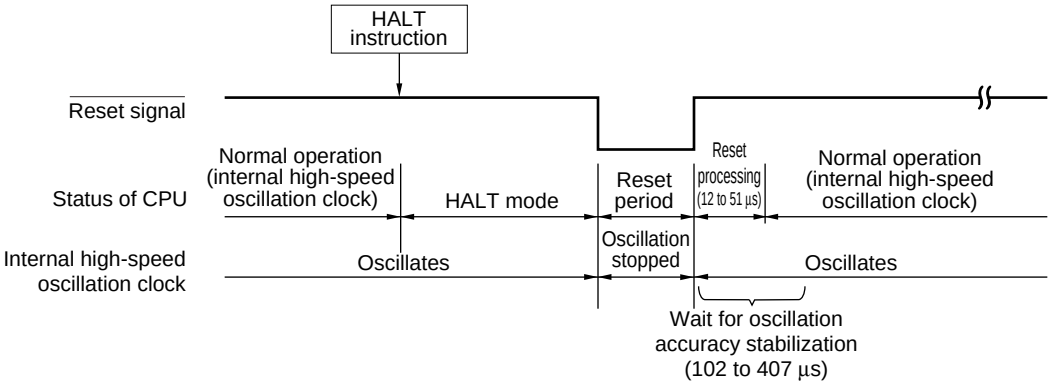
Figure 18-4. HALT Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



Note Oscillation stabilization time is not required when using the external main system clock (f_{EXCLK}) as the high-speed system clock.

(2) When internal high-speed oscillation clock is used as CPU clock



Remark f_X: X1 clock oscillation frequency

Table 18-2. 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

18.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed system clock or internal high-speed oscillation 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 18-3. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Stopped		
		f_X			
		f_{EXCLK}	Input invalid		
	f_{IL}		Status before STOP mode was set is retained		
PLL			Not operating (executing the STOP instruction during PLL operation is prohibited)		
CPU			Operation stopped		
Flash memory					
RAM			Status before STOP mode was set is retained		
Port (latch)					
16-bit timer	X0		Operation stopped		
	X1				
16-bit timer/event counter 00					
8-bit timer/event counter 51			Operable only when TI51 is selected as the count clock		
8-bit timer H1			Operable only when f_{IL}, $f_{IL}/2^6$, $f_{IL}/2^{15}$ is selected as the count clock		
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
A/D converter			Operation stopped		
Comparators 0 to 2			Comparator n is operable when both CnDFS0 = 0 and CnDFS1 = 0 are set. (n = 0 to 2)		
Serial interface	UART6		Operation stopped		
	CSI11		Operable only when external clock is selected as the serial clock		
	IICA		Wakeup by address match operable		
Multiplier			Operation stopped		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock, f_{IL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

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

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. When transitioning to the STOP mode, it is possible to achieve low power consumption by setting RMC = 56H.

3. 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.

4. 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

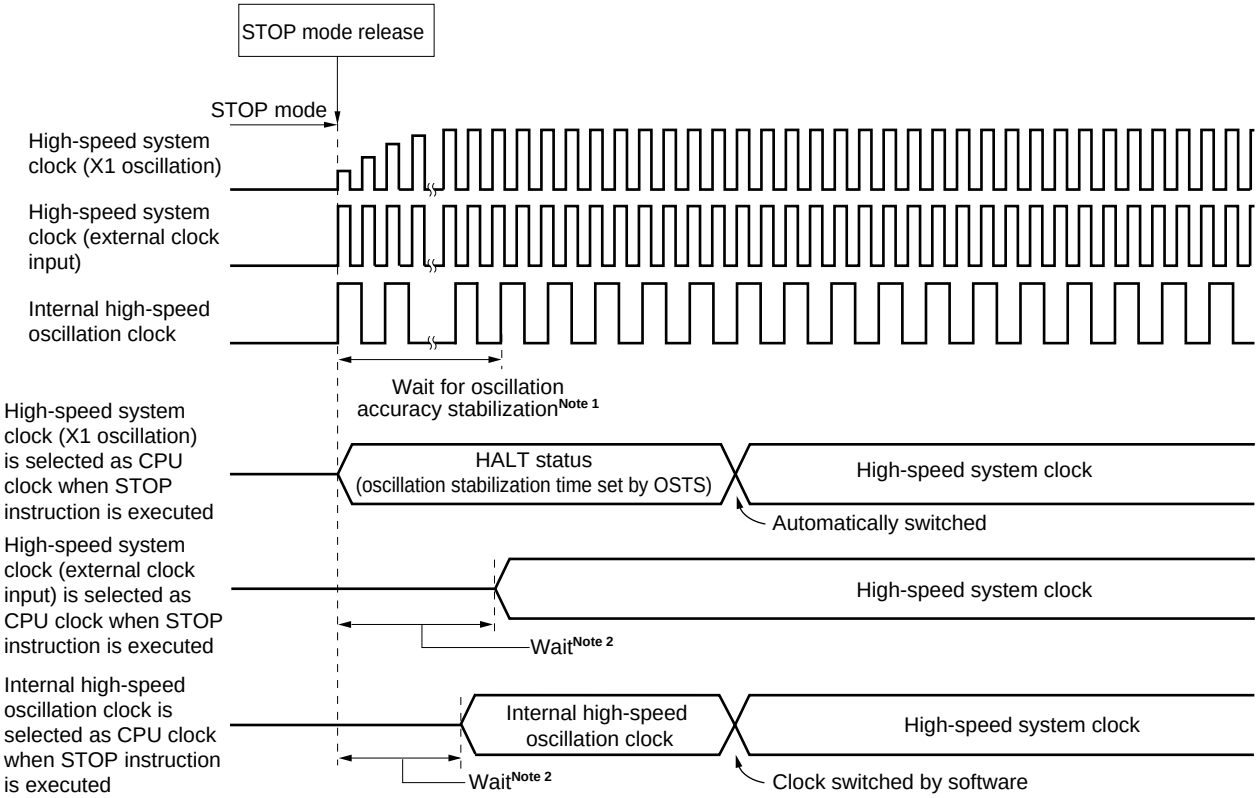
Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

5. Execute the STOP instruction after having confirmed that the internal high-speed oscillator is operating stably (RSTS = 1).

6. Be sure to stop the PLL operation (PLLON = 0) before executing the STOP instruction.

(2) STOP mode release

Figure 18-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Notes 1. The wait time for oscillation accuracy stabilization is as follows:

- RMC register = 00H: 102 to 407 μs
- RMC register = 56H: 120 to 481 μs

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

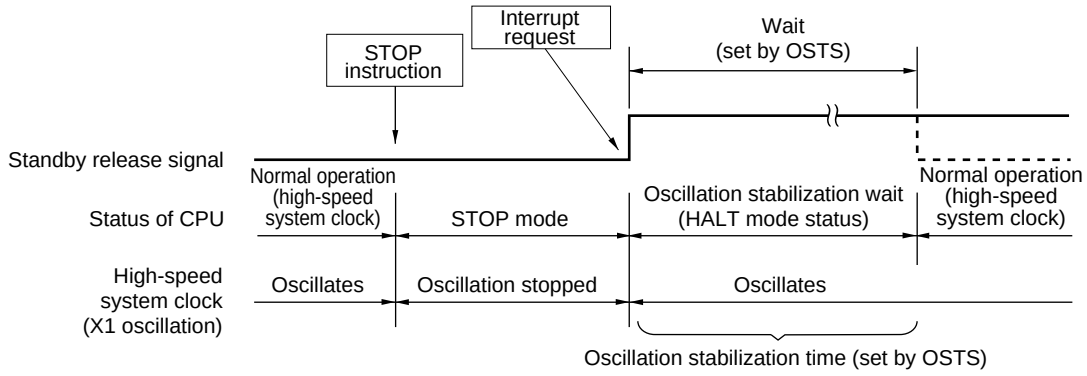
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

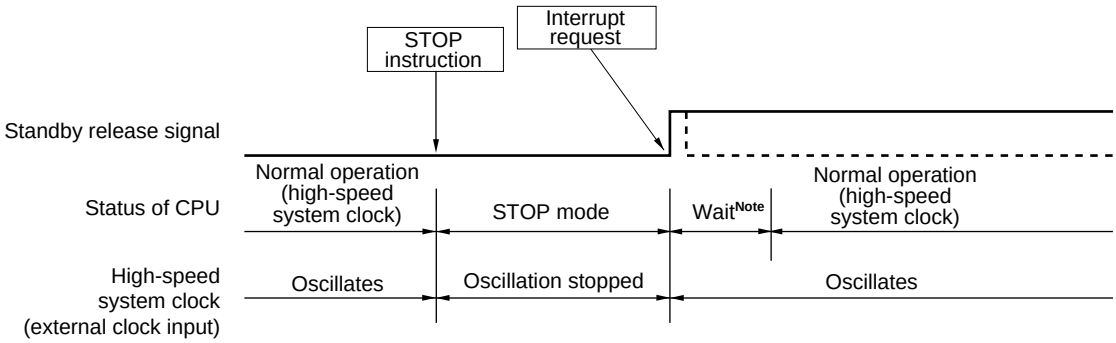
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 18-6. 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



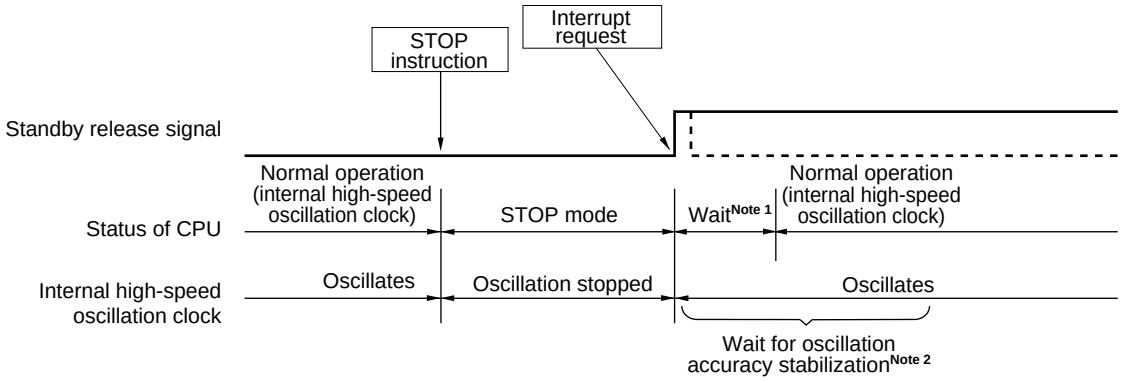
Note The wait time is as follows:

- 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 18-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock



Notes 1. 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

2. The wait time for oscillation accuracy stabilization is as follows:

- RMC register = 00H: 102 to 407 μs
- RMC register = 56H: 120 to 481 μs

Caution The above-mentioned oscillation accuracy stabilization wait time will be stated after the device has been evaluated.

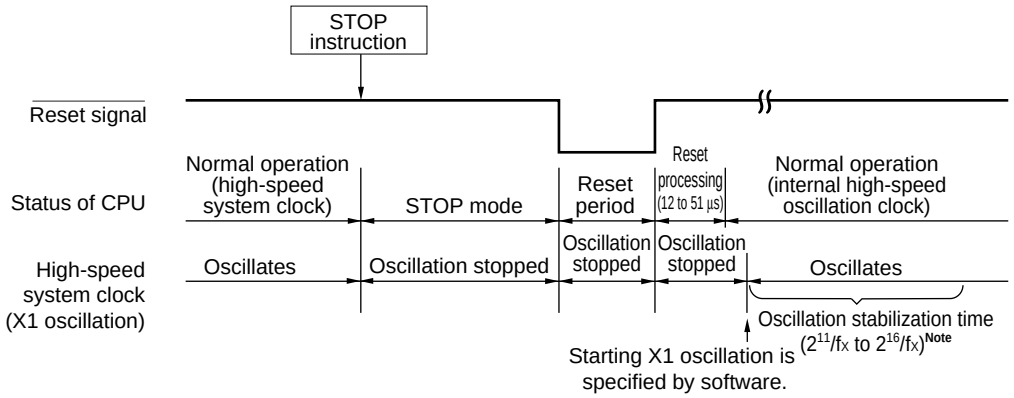
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 18-7. STOP Mode Release by Reset (1/2)

(1) When high-speed system clock is used as CPU clock

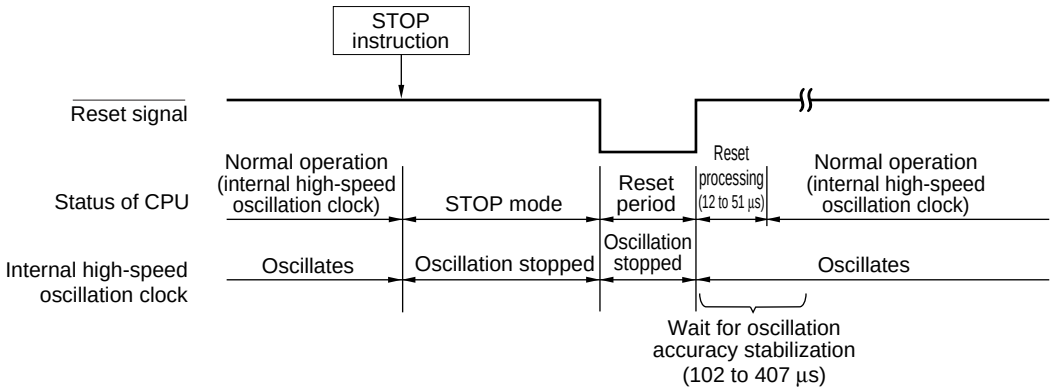


Note Oscillation stabilization time is not required when using the external main system clock (f_{EXCLK}) as the high-speed system clock.

Remark f_X: X1 clock oscillation frequency

Figure 18-7. STOP Mode Release by Reset (2/2)

(2) When internal high-speed oscillation clock is used as CPU clock



Caution The above-mentioned reset processing time and oscillation accuracy stabilization wait time will be stated after the device has been evaluated.

Table 18-4. 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 19 RESET FUNCTION

The reset function is mounted onto all 78K0/Fx2-L microcontroller products.

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 the low-voltage detector (LVI)

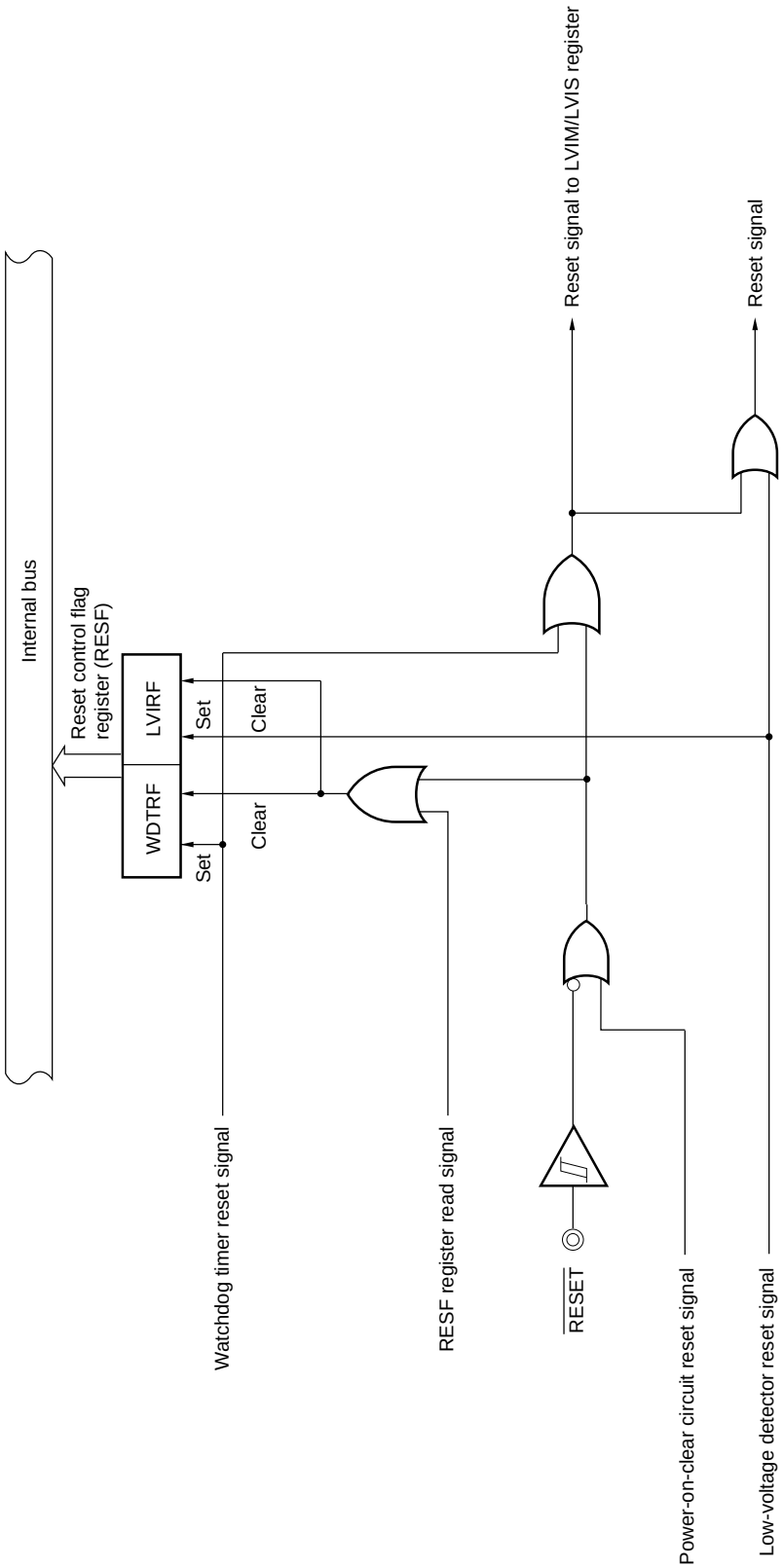
External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

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 19-1 and 19-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

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 (refer to **Figures 19-2** to **19-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (refer to **CHAPTER 20 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 21 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.
(If an external reset is effected upon power application, the period during which the supply voltage is outside the operating range (V_{DD} < 1.8 V) is not counted in the 10 μs. However, the low-level input may be continued before POC is released.)
2. During reset signal generation, the X1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
3. When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input. However, because SFR is initialized, the port pins become high-impedance.

Figure 19-1. Block Diagram of Reset Function



Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level selection register

Figure 19-2. Timing of Reset by $\overline{\text{RESET}}$ Input

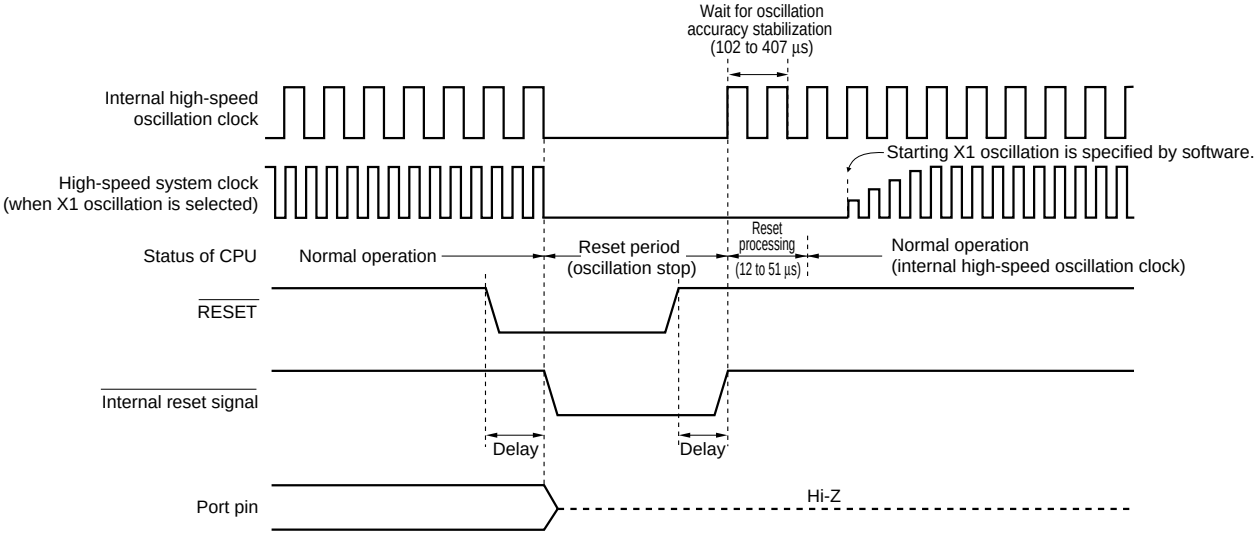
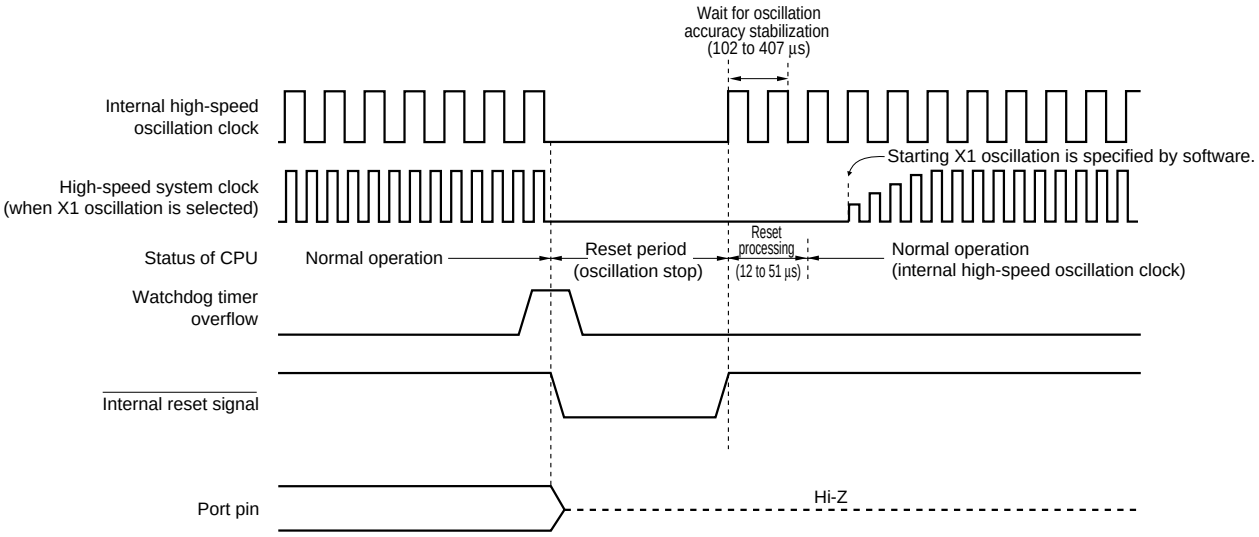
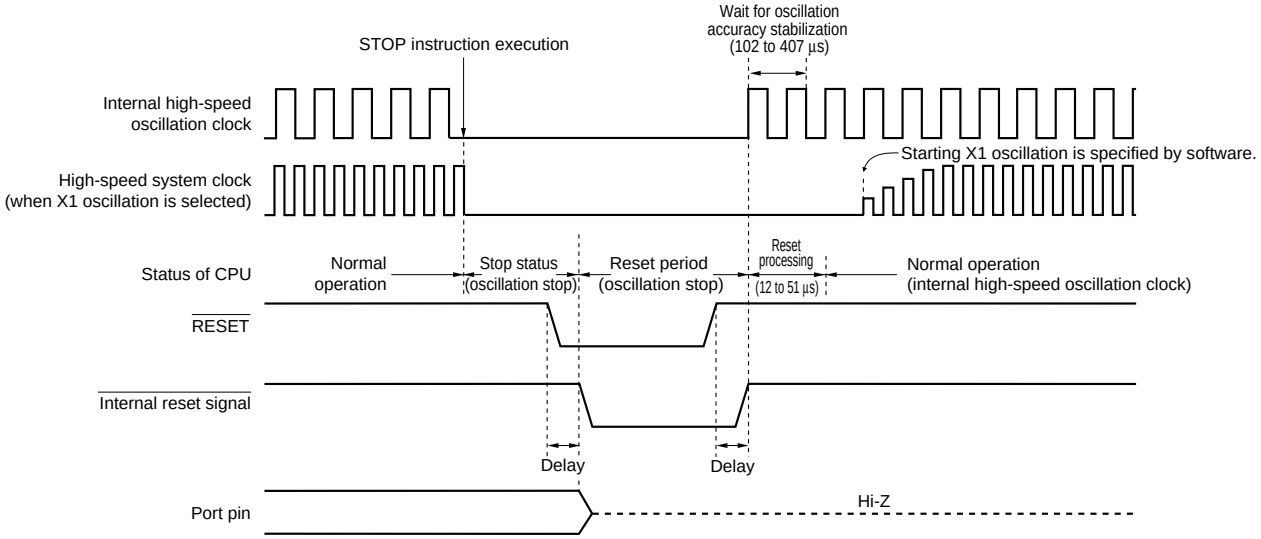


Figure 19-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

Figure 19-4. 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, refer to **CHAPTER 20 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 21 LOW-VOLTAGE DETECTOR**.

Table 19-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (X1 and X2 pins are input port mode)
		f_{EXCLK}	Clock input invalid (EXCLK pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Flash memory			
RAM			Operation stopped (The value, however, is retained when the voltage is at least the power-on clear detection voltage.)
Port (latch)			Operation stopped
16-bit timers		X0	
		X1	
16-bit timer/event counter 00			
8-bit timer/event counter 51			
8-bit timer H1			
Watchdog timer			
A/D converter			
Comparators 0 to 2			
Serial interface		UART6	
		CSI11	
		IICA	
Multiplier			
External interrupt			
Power-on-clear function			Operable
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)
On-chip debug function			Operation stopped

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock, f_{IL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (1/4)

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers 0, 2, 3, 6, 7, 12 (P0, P2, P3, P6, P7, P12) (output latches)		00H
Port mode registers 0, 2, 3, 6, 7 (PM0, PM2, PM3, PM6, PM7)		FFH
Pull-up resistor option registers 0, 3, 6 (PU0, PU3, PU6)		00H
Port input mode register 6 (PIM6)		00H
Port output mode register 6 (POM6)		00H
Port alternate switch control register (MUXSEL)		00H
Internal memory size switching register (IMS)		CFH[Note 3]

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. Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated below after release of reset.

Products			IMS	ROM Capacity	Internal High-Speed RAM Capacity
78K0/FY2-L	78K0/FA2-L	78K0/FB2-L			
μPD78F0854	μPD78F0857	–	61H	4 KB	384 bytes
μPD78F0855	μPD78F0858	μPD78F0864	42H	8 KB	512 bytes
μPD78F0856	μPD78F0859	μPD78F0865	04H	16 KB	768 bytes

Remark The special function registers (SFRs) mounted depend on the product. Refer to **3.2.3 Special function registers (SFRs)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (2/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Clock operation mode select register (OSCCTL)		00H
Processor clock control register (PCC)		01H
Internal oscillation mode/PLL control 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 timers X0 and X1	16-bit timer X0 operation control registers 0 to 4 (TX0CTL0, TX0CTL1, TX0CTL2, TX0CTL3, TX0CTL4)	00H
	16-bit timer X1 operation control registers 0 to 2, 4 (TX1CTL0, TX1CTL1, TX1CTL2, TX1CTL4)	00H
	16-bit timer X0 output control register 0 (TX0IOC0)	00H
	16-bit timer X1 output control register 0 (TX1IOC0)	00H
	16-bit timer X0 compare registers 0 to 3 (TX0CR0, TX0CR1, TX0CR2, TX0CR3)	0000H
	16-bit timer X1 compare registers 0 to 3 (TX1CR0, TX1CR1, TX1CR2, TX1CR3)	0000H
	16-bit timer X0 capture/compare register 0 (TX0CCR0)	0000H
	16-bit timer X1 capture/compare register 0 (TX1CCR0)	0000H
	High-impedance output function enable register (HIZTREN)	00H
	High-impedance output mode select register (HIZTRS)	00H
	High-impedance output function control register 0 (HZA0CTL0)	00H
16-bit timer/event counter 00	16-bit timer counter 00 (TM00)	0000H
	16-bit timer capture/compare registers 000, 010 (CR000, CR010)	0000H
	16-bit timer mode control register 00 (TMC00)	00H
	Prescaler mode register 00 (PRM00)	00H
	Capture/compare control register 00 (CRC00)	00H
	16-bit timer output control register 00 (TOC00)	00H
8-bit timer/event counter 51	8-bit timer counter 51 (TM51)	00H
	8-bit compare register 51 (CR51)	00H
	Timer clock selection register 51 (TCL51)	00H
	8-bit timer mode control register 51 (TMC51)	00H
8-bit timer H1	8-bit timer H1 compare registers 01, 11 (CMP01, CMP11)	00H
	8-bit timer H1 mode register 1 (TMHMD1)	00H
	8-bit timer H1 carrier control register 1 (TMCYC1)	00H
Watchdog timer	Watchdog timer enable register (WDTE)	1AH/9AH[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. The reset value of WDTE is determined by the option byte setting.

Remark The special function registers (SFRs) mounted depend on the product. Refer to **3.2.3 Special function registers (SFRs)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (3/4)

Hardware		Status After Reset Acknowledgment[Note]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register L (ADCRL)	00H
	8-bit A/D conversion result register H (ADCRH)	00H
	A/D converter mode register 0 (ADM0)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register 0 (ADPC0)	00H
	A/D port configuration register 1 (ADPC1)	00H
Comparator	Comparator 0 control register (C0CTL)	00H
	Comparator 0 internal reference voltage setting register (C0RVM)	00H
	Comparator 1 control register (C1CTL)	00H
	Comparator 1 internal reference voltage setting register (C1RVM)	00H
	Comparator 2 control register (C2CTL)	00H
	Comparator 2 internal reference voltage setting register (C2RVM)	00H
	Comparator output flag register (CMPFLG)	00H
Serial interface UART6	Receive buffer register 6 (RXB6)	FFH
	Transmit buffer register 6 (TXB6)	FFH
	Asynchronous serial interface operation mode register 6 (ASIM6)	01H
	Asynchronous serial interface reception error status register 6 (ASIS6)	00H
	Asynchronous serial interface transmission status register 6 (ASIF6)	00H
	Clock selection register 6 (CKSR6)	00H
	Baud rate generator control register 6 (BRGC6)	FFH
	Asynchronous serial interface control register 6 (ASICL6)	16H
	Input switch control register (ISC)	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.

Remark The special function registers (SFRs) mounted depend on the product. Refer to **3.2.3 Special function registers (SFRs)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (4/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Serial interface CSI11	Transmit buffer register 11 (SOTB11)	00H
	Serial I/O shift register 11 (SIO11)	00H
	Serial operation mode register 11 (CSIM11)	00H
	Serial clock selection register 11 (CSIC11)	00H
Serial interface IICA	IICA shift register (IICA)	00H
	IICA status register 0 (IICAS0)	00H
	IICA flag register 0 (IICAF0)	00H
	IICA control register 0 (IICACTL0)	00H
	IICA control register 1 (IICACTL1)	00H
	IICA low-level width setting register (IICWL)	FFH
	IICA high-level width setting register (IICWH)	FFH
	Slave address register 0 (SVA0)	00H
Multiplier	Multiplication input date register A (MULAL, MULAH)	00H
	Multiplication input date register B (MULBL, MULBH)	00H
	16-bit higher multiplication result storage register (MUL0H)	0000H
	16-bit lower multiplication result storage register (MUL0L)	0000H
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]
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 registers 0, 1 (EGPCTL0, EGPCTL1)	00H
	External interrupt falling edge enable registers 0, 1 (EGNCTL0, EGNCTL1)	00H
Regulator	Regulator mode control register (RMC)	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. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI (Except reset by LVI default start function)	Reset by LVI default start function
RESF	WDTRF flag	Cleared (0)	Cleared (0)	Set (1)	Held	Cleared (0)
	LVIRF flag			Held	Set (1)	
LVIM		Cleared (00H)	Cleared (00H)	Cleared (00H)	Held	Cleared (00H)
LVIS						

Remark The special function registers (SFRs) mounted depend on the product. Refer to **3.2.3 Special function registers (SFRs)**.

19.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/Fx2-L microcontrollers. The reset control flag register (RESF) is used to store which source has generated the reset request.

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

$\overline{\text{RESET}}$ input, reset by power-on-clear (POC) circuit, and reading RESF set RESF to 00H.

Figure 19-5. 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 19-3.

Table 19-3. RESF Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI (Except reset by LVI default start function)	Reset by LVI default start function
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held	Cleared (0)
LVIRF			Held	Set (1)	

CHAPTER 20 POWER-ON-CLEAR CIRCUIT

20.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) is mounted onto all 78K0/Fx2-L microcontroller products.
The power-on-clear circuit has the following functions.

- Generates internal reset signal at power on.
- The reset signal is released when the supply voltage (V_{DD}) exceeds POC detection voltage (V_{POR} = 1.61 V ±0.09 V).

Caution If the LVI default function enabled is set by using an option byte, the reset signal is not released until the supply voltage (V_{DD}) exceeds 1.91 V ±0.1 V.

- Compares supply voltage (V_{DD}) and POC detection voltage (V_{PDR} = 1.59 V ±0.09 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

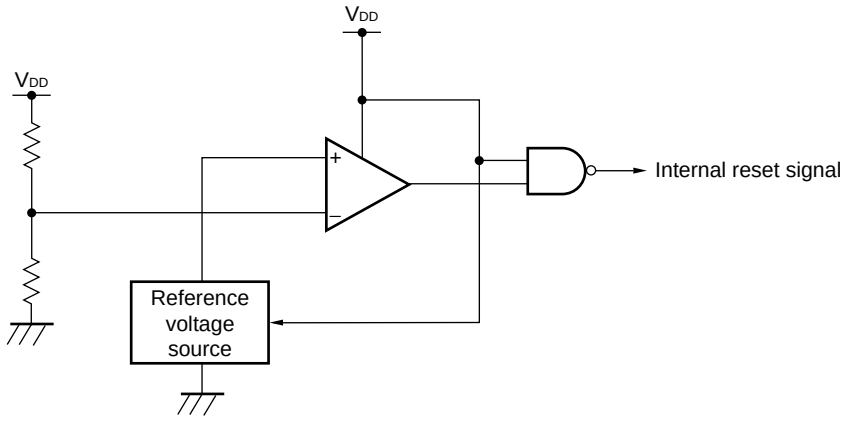
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark The 78K0/Fx2-L microcontrollers incorporate multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT) and 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, refer to **CHAPTER 19 RESET FUNCTION**.

20.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 20-1.

Figure 20-1. Block Diagram of Power-on-Clear Circuit



20.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds POC detection voltage (V_{POR} = 1.61 V ±0.09 V, the reset status is released.

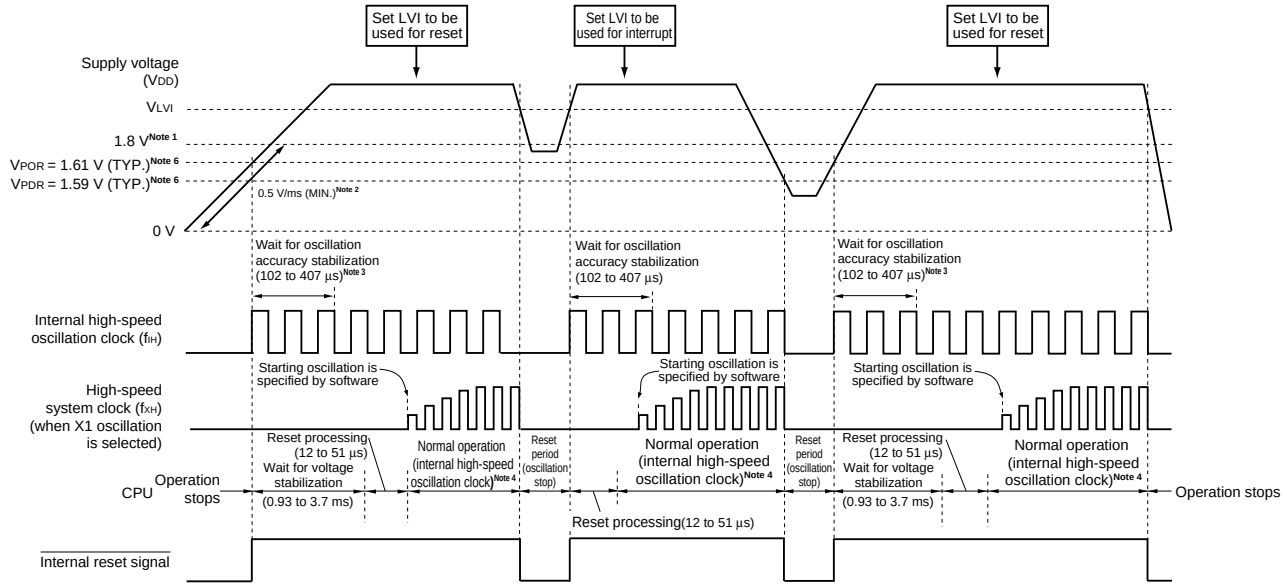
Caution If the LVI default function enabled is set by using an option byte, the reset signal is not released until the supply voltage (V_{DD}) exceeds 1.91 V ±0.1 V.

- The supply voltage (V_{DD}) and POC detection voltage (V_{PDR} = 1.59 V ±0.09 V are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 20-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) When LVI is OFF upon power application (option byte: LVISTART = 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 rate at which the voltage rises to 1.8 V after power application is slower than 0.5 V/ms (MIN.), input a low level to the $\overline{\text{RESET}}$ pin before the voltage reaches to 1.8 V, or set LVI to ON by default by using an option byte (option byte: LVISTART = 1).

3. The internal voltage stabilization wait time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

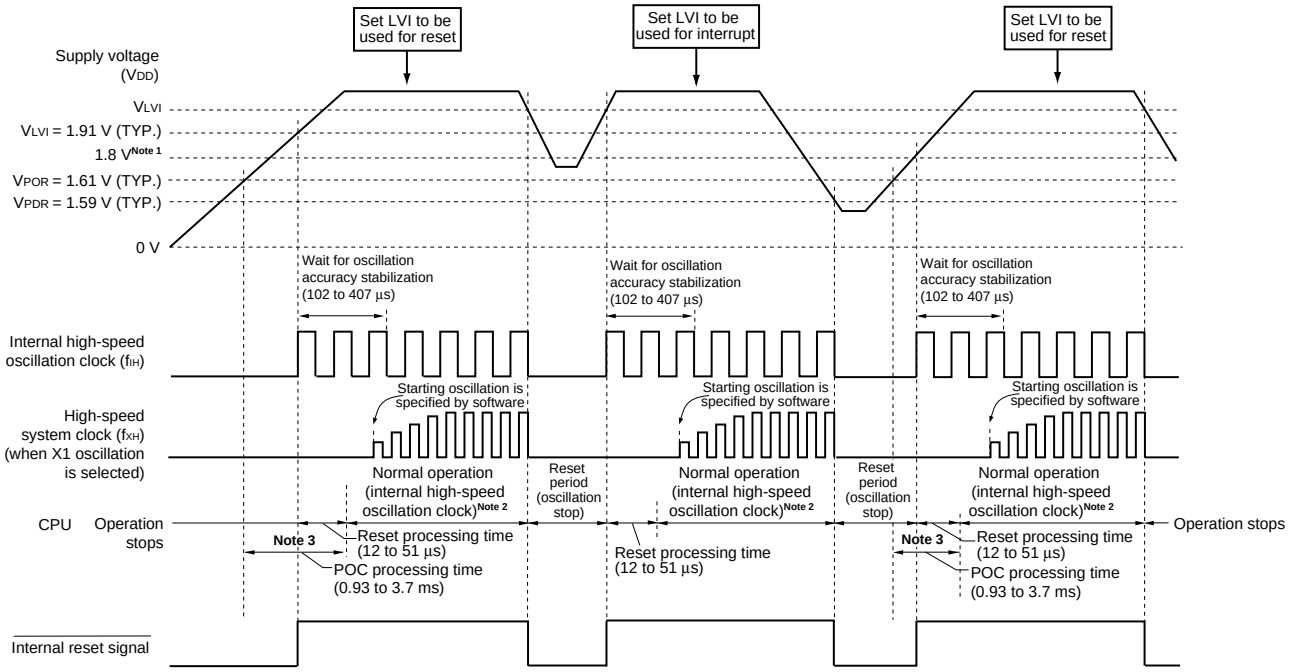
4. The CPU clock can be switched from the internal high-speed oscillation clock to the high-speed system clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time.

Caution Set the low-voltage detector by software after the reset status is released (refer to CHAPTER 21 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

Figure 20-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) When LVI is ON upon power application (option byte: LVISTART = 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 CPU clock can be switched from the internal high-speed oscillation clock to the high-speed system clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time.

3. The following times are required between reaching the POC detection voltage (1.59 V (TYP.)) and starting normal operation.

- When the time to reach 1.91 V (TYP.) from 1.59 V (TYP.) is less than 3.7 ms:
  A POC processing time of about 1.0 to 3.8 ms is required between reaching 1.59 V (TYP.) and starting normal operation.
- When the time to reach 1.91 V (TYP.) from 1.59 V (TYP.) is greater than 3.7 ms:
  A reset processing time of about 12 to 51 μs is required between reaching 1.91 V (TYP.) and starting normal operation.

Caution Set the low-voltage detector by software after the reset status is released (refer to CHAPTER 21 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

20.4 Cautions for Power-on-Clear Circuit

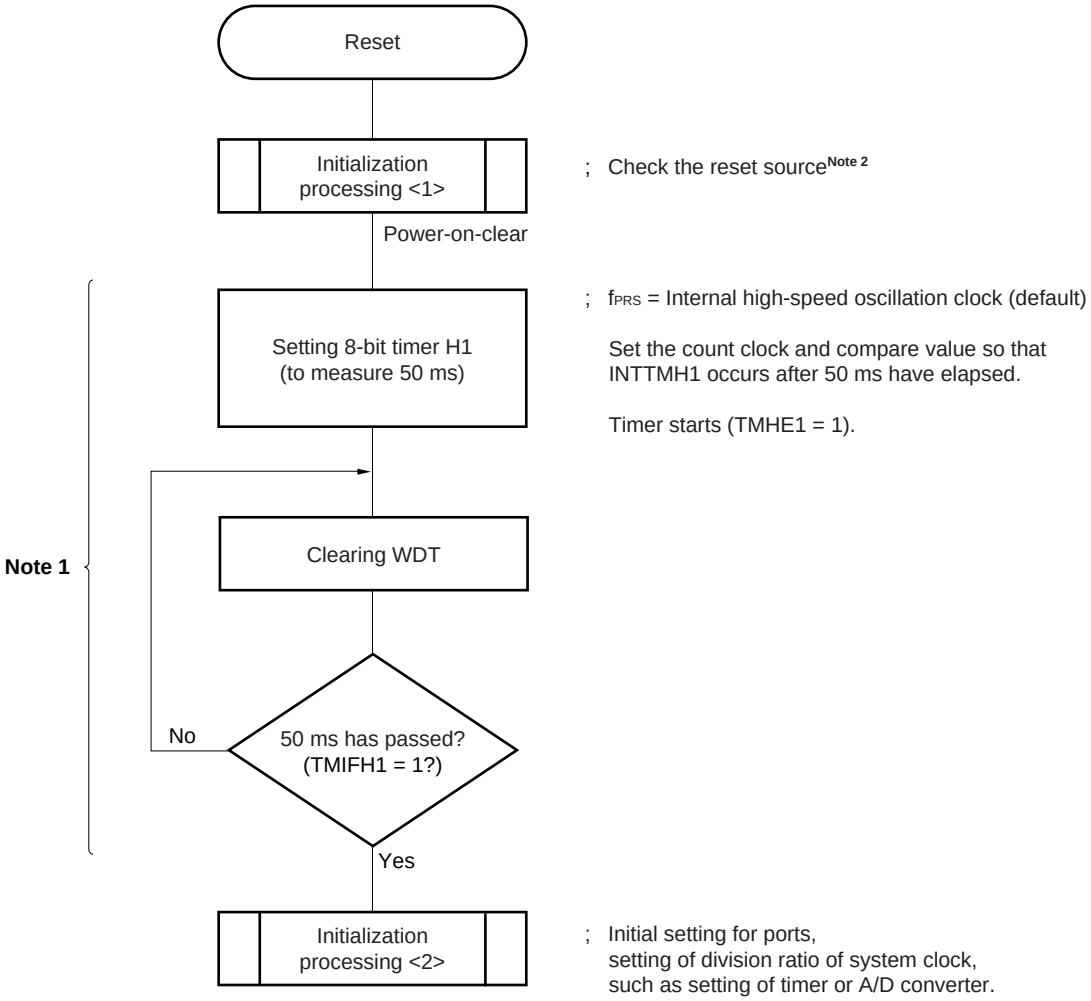
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 20-3. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

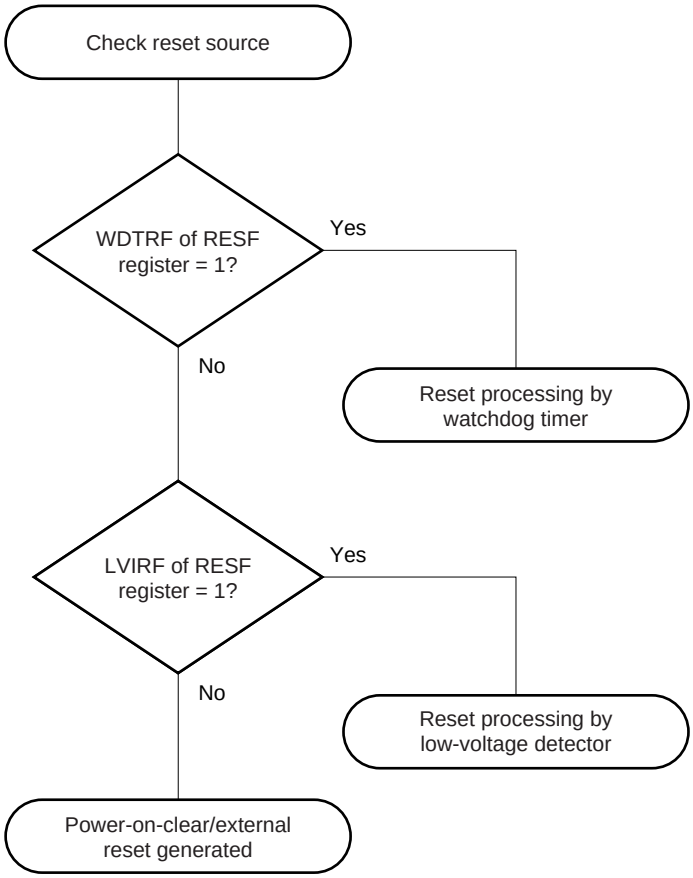


Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Figure 20-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



CHAPTER 21 LOW-VOLTAGE DETECTOR

21.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) is mounted onto all 78K0/Fx2-L microcontroller products.
The low-voltage detector has the following functions.

- The LVI circuit compares the supply voltage (V_{DD}) with the LVI detection voltage (V_{LVI}), and generates an internal reset or internal interrupt signal.
- The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < the LVI detection voltage (V_{LVI} = 1.91 V ±0.1 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < the LVI detection voltage (V_{LVI} = 1.91 V ±0.1 V).
- A reset or an interrupt can be selected to be generated after detection by software.
- Detection levels (V_{LVI},16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage (V_{DD})	
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 ($V_{DD} \geq V_{LVI}$).

Remark LVIMD: Bit 1 of low-voltage detection register (LVIM)

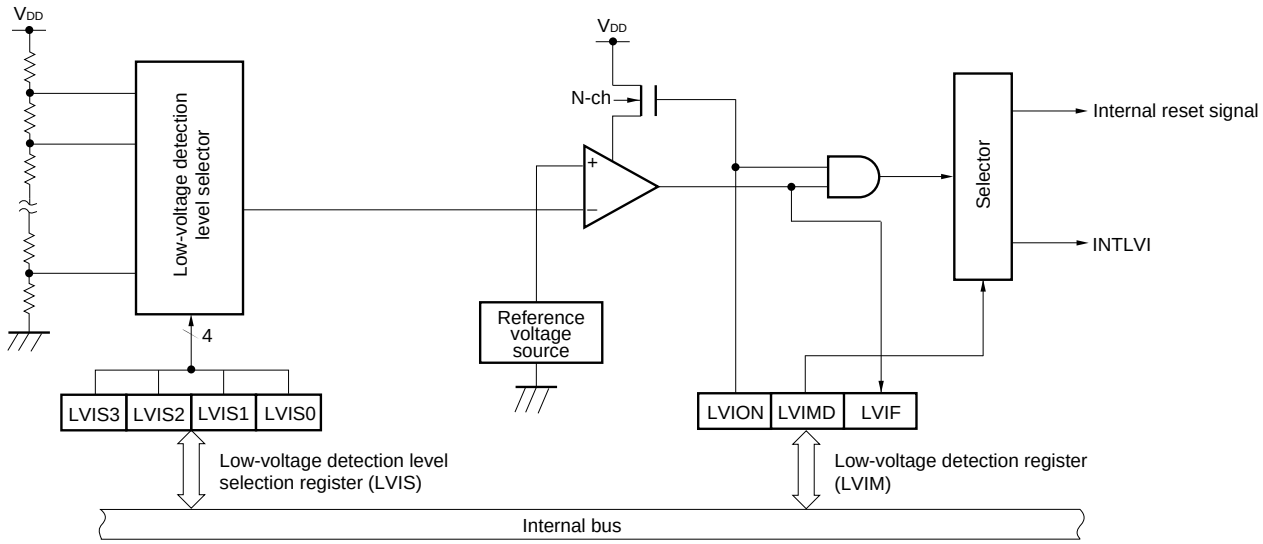
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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, refer to **CHAPTER 19 RESET FUNCTION**.

21.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 21-1.

Figure 21-1. Block Diagram of Low-Voltage Detector



21.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level select register (LVIS)

(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 21-2. 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	0	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVIMD[Note 3]	Low-voltage detection operation mode (interrupt/reset) selection
0	Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the LVI detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).
1	Generates an internal reset signal when the supply voltage (V_{DD}) < the LVI detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$.

LVIF	Low-voltage detection flag
0	Supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI}), or when LVI operation is disabled
1	Supply voltage (V_{DD}) < LVI detection voltage (V_{LVI})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte. This register is not cleared (00H) by LVI reset (except reset by LVI default start function). The value of this register is reset to “00H” by other resets.

2. Bit 0 is read-only.

3. LVION and LVIMD 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 the operation stabilizes, an external input (minimum pulse width: 200 μs) of 200 μs or more is required until LVIF is set (1) after the voltage drops to the LVI detection voltage or less.

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. If LVI operation is disabled (clears LVION) when LVI is used in interrupt mode (LVIMD = 0) and the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

(2) Low-voltage detection level select register (LVIS)

This register selects the low-voltage detection level.

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

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 21-3. Format of Low-Voltage Detection Level Select 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} (4.22 ±0.1 V)
0	0	0	1	V_{LVI1} (4.07 ±0.1 V)
0	0	1	0	V_{LVI2} (3.92 ±0.1 V)
0	0	1	1	V_{LVI3} (3.76 ±0.1 V)
0	1	0	0	V_{LVI4} (3.61 ±0.1 V)
0	1	0	1	V_{LVI5} (3.45 ±0.1 V)
0	1	1	0	V_{LVI6} (3.30 ±0.1 V)
0	1	1	1	V_{LVI7} (3.15 ±0.1 V)
1	0	0	0	V_{LVI8} (2.99 ±0.1 V)
1	0	0	1	V_{LVI9} (2.84 ±0.1 V)
1	0	1	0	V_{LVI10} (2.68 ±0.1 V)
1	0	1	1	V_{LVI11} (2.53 ±0.1 V)
1	1	0	0	V_{LVI12} (2.38 ±0.1 V)
1	1	0	1	V_{LVI13} (2.22 ±0.1 V)
1	1	1	0	V_{LVI14} (2.07 ±0.07 V)
1	1	1	1	V_{LVI15} (1.91 ±0.1 V)

Note The reset value changes depending on the reset source.

If the LVIS register is reset by LVI (except reset by LVI default start function), it is not reset but holds the current value.

The value of this register is reset to “00H” by 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.

21.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

Compares the supply voltage (V_{DD}) and LVI detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.

Remark The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 1.91 V ±0.1 V).

(2) Used as interrupt (LVIMD = 0)

Compares the supply voltage (V_{DD}) and LVI 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).

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)

21.4.1 When used as reset

(1) When LVI default start function stopped is set (LVISTART = 0)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set the LVI detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<R> <4> Use software to wait for the following time (210 μs (Total)).

- Operation stabilization time (10 μs(MAX.))
- Minimum pulse width: (200 μs(MIN.))

<5> Wait until it is checked that (supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.

<6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

Figure 21-4 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

2. If supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

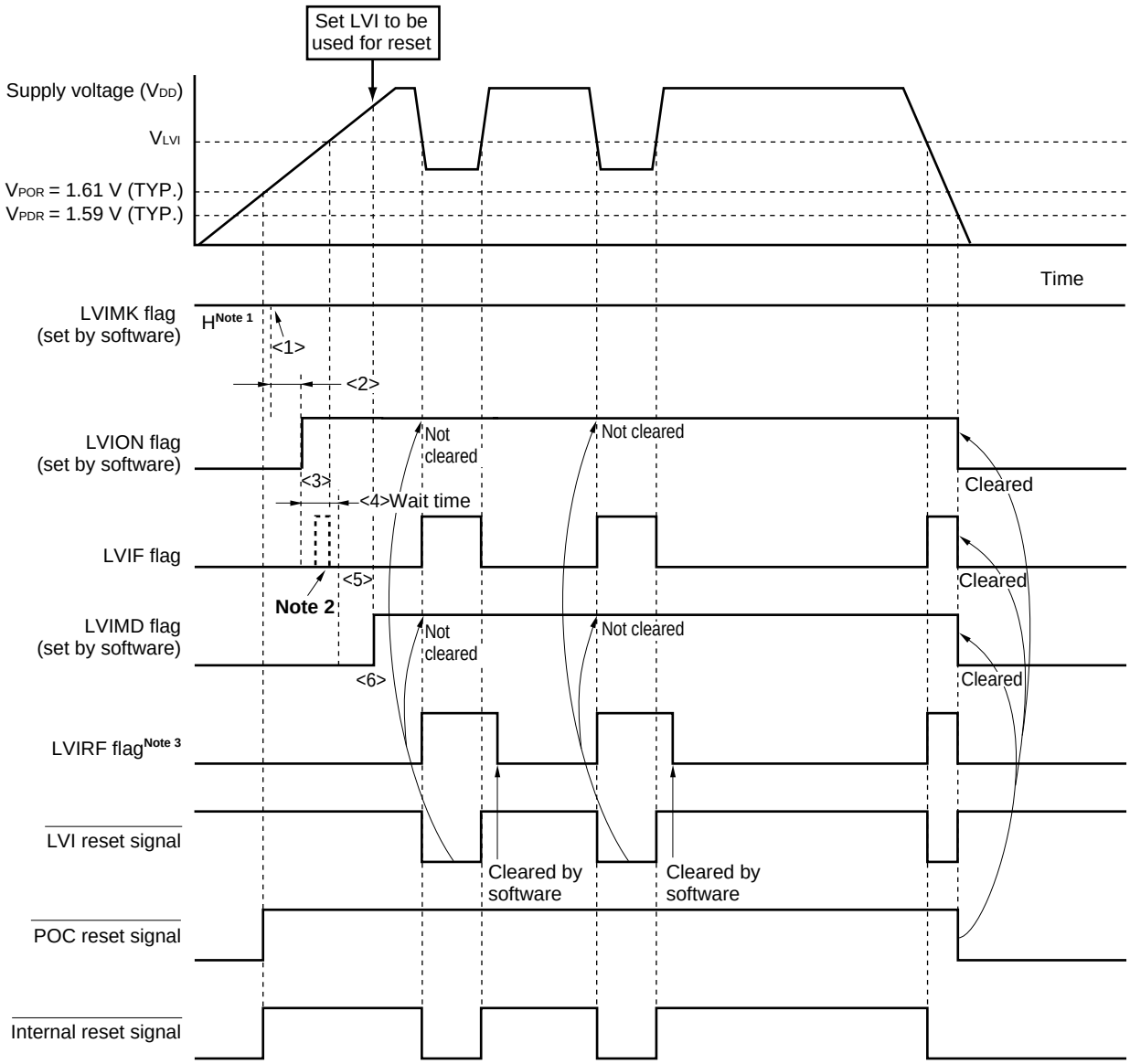
- When stopping operation

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 21-4. Timing of Low-Voltage Detector Internal Reset Signal Generation (LVISTART = 0)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, refer to **CHAPTER 19 RESET FUNCTION**.

Remarks 1. <1> to <6> in Figure 21-4 above correspond to <1> to <6> in the description of “When starting operation” in **21.4.1 (1) When LVI default start function stopped is set (LVISTART = 0).**

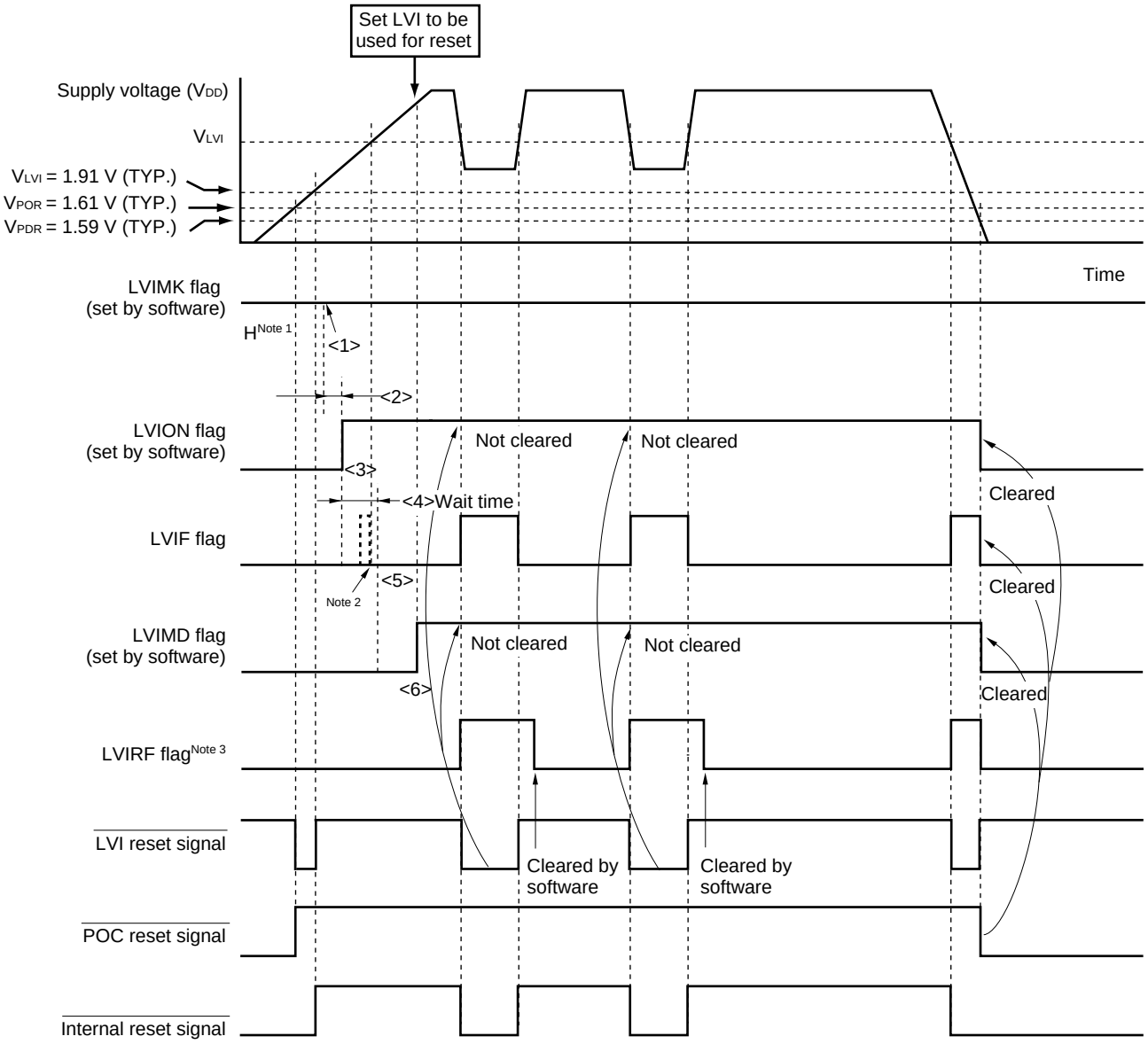
2. V_{POR}: POC power supply rise detection voltage

V_{PDR}: POC power supply fall detection voltage

(2) When LVI default start function enabled is set (LVISTART = 1)

The setting when operation starts and when operation stops is the same as described in **21.4.1 (1) When LVI default start function stopped is set (LVISTART = 0)**.

Figure 21-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (LVISTART = 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF).
For details of RESF, refer to **CHAPTER 19 RESET FUNCTION**.

Remarks 1. <1> to <6> in Figure 21-5 above correspond to <1> to <6> in the description of “When starting operation” in **21.4.1 (1) When LVI default start function stopped is set (LVISTART = 0).**

2. VPOR: POC power supply rise detection voltage
VPDR: POC power supply fall detection voltage

21.4.2 When used as interrupt

(1) When LVI default start function stopped is set (LVISTART = 0)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set the LVI detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<3> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<R> <5> Use software to wait for the following time (210 μs (Total)).

- Operation stabilization time(10 μs (MAX.))
- Minimum pulse width: (200 μs (MIN.))

<6> Confirm that “supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI})” when detecting the falling edge of V_{DD}, or “supply voltage (V_{DD}) < LVI detection voltage (V_{LVI})” when detecting the rising edge of V_{DD}, at bit 0 (LVIF) of LVIM.

<7> Clear the interrupt request flag of LVI (LVIIF) to 0.

<8> Release the interrupt mask flag of LVI (LVIMK).

<9> Execute the EI instruction (when vector interrupts are used).

Figure 21-6 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

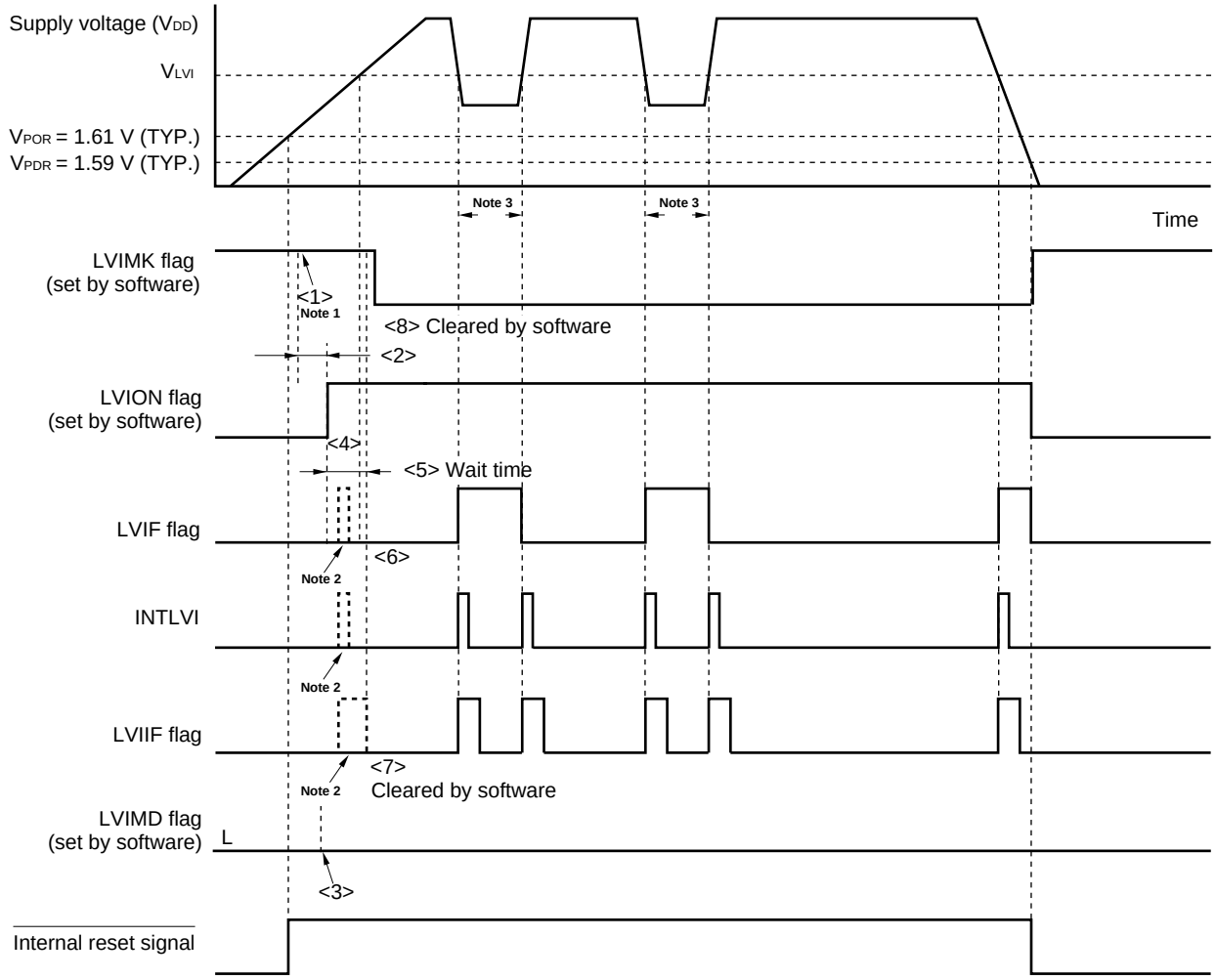
- When stopping operation

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 21-6. Timing of Low-Voltage Detector Interrupt Signal Generation (LVISTART = 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 LVI operation is disabled (clears LVION) when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remarks 1. <1> to <8> in Figure 21-6 above correspond to <1> to <8> in the description of “When starting operation” in **21.4.2 (1) When LVI default start function stopped is set (LVISTART = 0).**

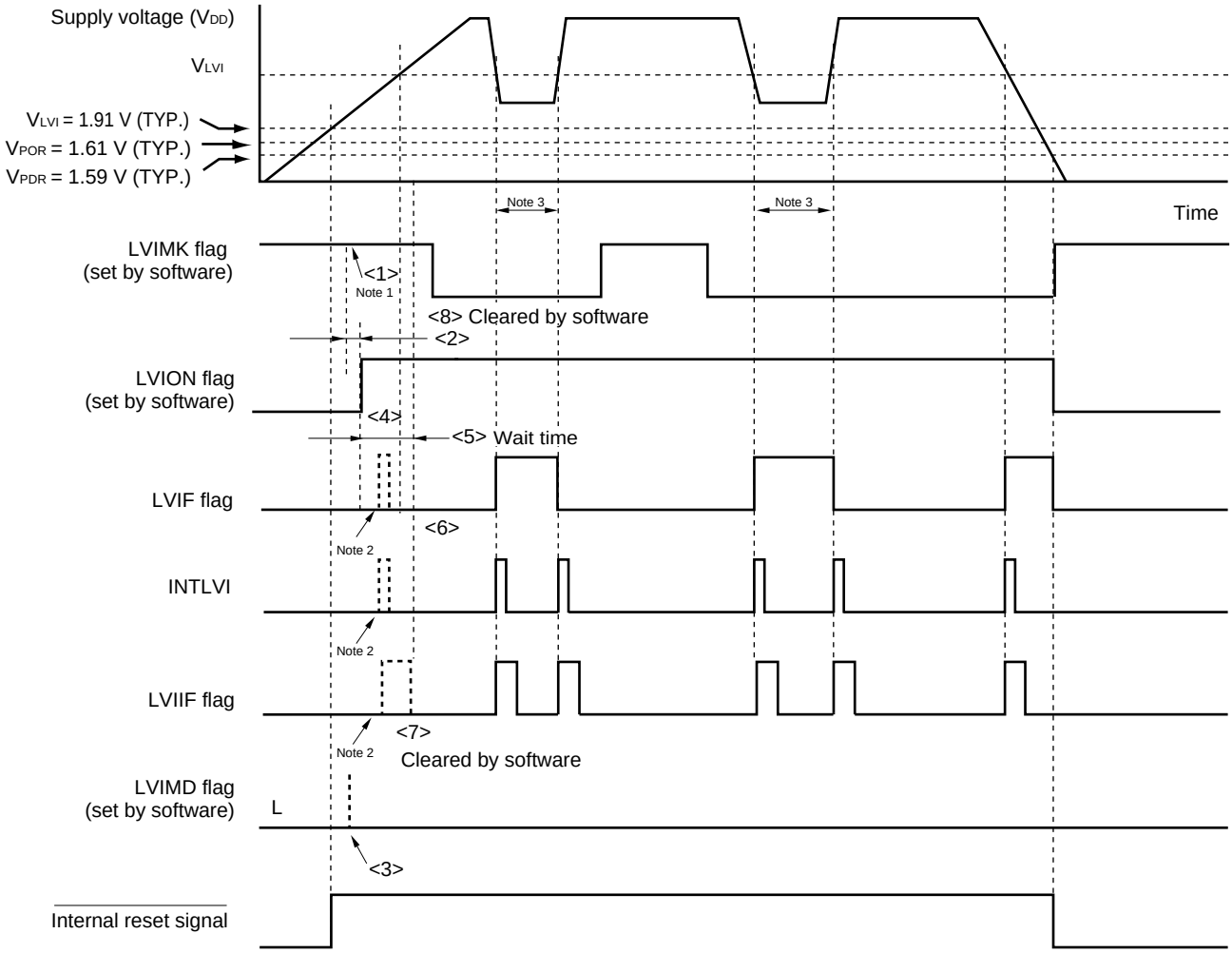
2. V_{POR}: POC power supply rise detection voltage

V_{PDR}: POC power supply fall detection voltage

(2) When LVI default start function enabled is set (LVISTART = 1)

The setting when operation starts and when operation stops is the same as described in **21.4.2 (1) When LVI default start function stopped is set (LVISTART = 0)**.

Figure 21-7. Timing of Low-Voltage Detector Interrupt Signal Generation (LVISTART = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVI operation is disabled (clears LVION) when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remarks 1. <1> to <8> in Figure 21-7 above correspond to <1> to <8> in the description of "When starting operation" in **21.4.2 (1) When LVI default start function stopped is set (LVISTART = 0).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

21.5 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.

Operation example 1: When used as reset

The system may be repeatedly reset and released from the reset status.

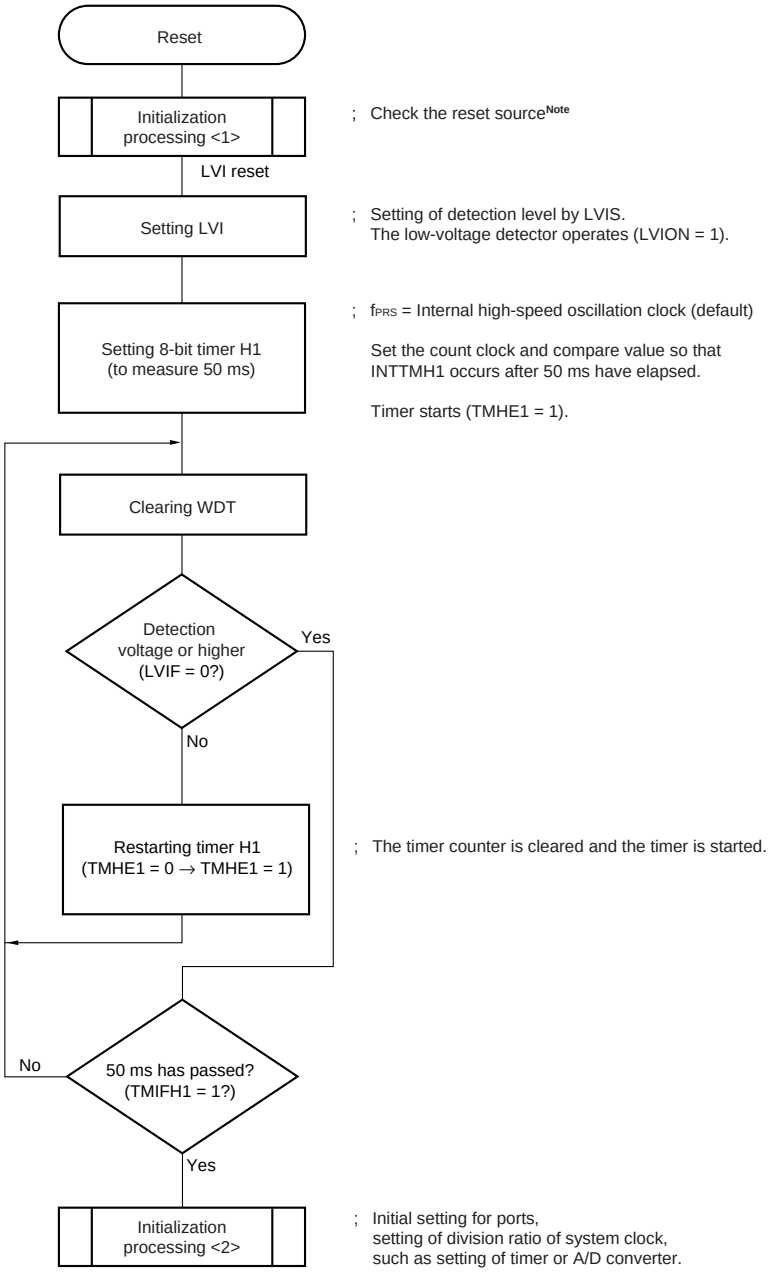
The time from reset release through microcontroller operation start can be set arbitrarily by the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (refer to **Figure 21-8**).

Figure 21-8. Example of Software Processing After Reset Release (1/2)

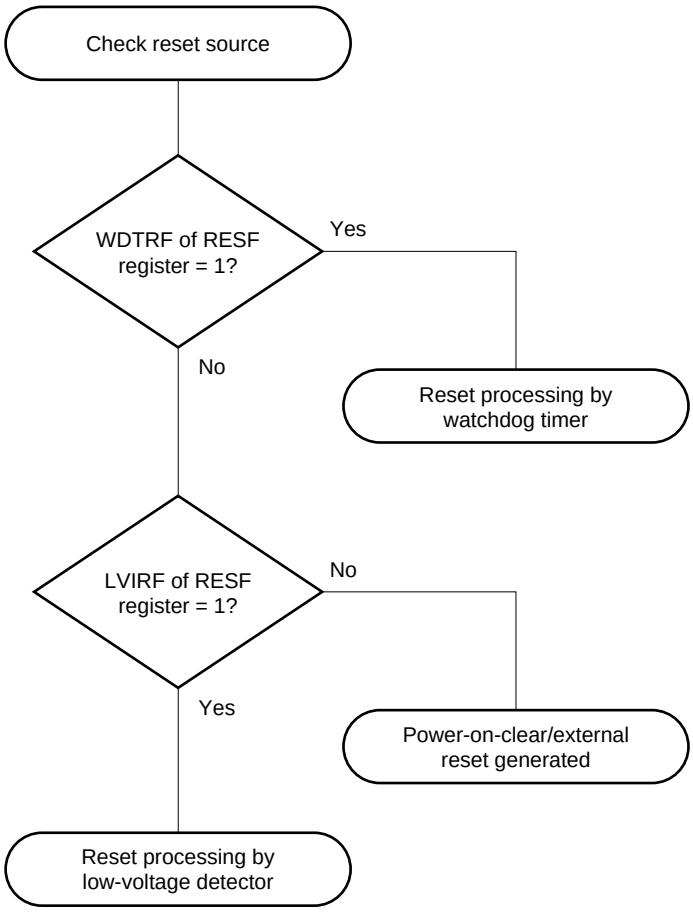
• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

Figure 21-8. Example of Software Processing After Reset Release (2/2)

• Checking reset source



Operation example 2: When used as interrupt

Interrupt requests may be generated frequently.

Take the following action.

<Action>

Confirm that "supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < LVI detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.

CHAPTER 22 REGULATOR

22.1 Regulator Overview

The 78K0/Fx2-L microcontrollers contain a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

The regulator output voltage is normally 2.4 V (TYP.), and in the low power consumption mode, 2.0 V (TYP.).

22.2 Registers Controlling Regulator

(1) Regulator mode control register (RMC)

This register sets the output voltage of the regulator.

RMC is set with an 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 22-1. Format of Regulator Mode Control Register (RMC)

Address: FF3DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	Control of output voltage of regulator
56H	Low power consumption mode (fixed to 2.0 V)
00H	Normal power mode (fixed to 2.4 V)
Other than above	Setting prohibited

Cautions 1. To change the RMC register setting value from 56H to 00H and use a CPU operating frequency of 5 MHz or more, change the PCC and RCM registers when 10 μs or more has elapsed after the RMC register was set.

2. When using the setting fixed to the low power consumption mode, the RMC register can be used in the following cases.

<When X1 clock is selected as the CPU clock> $f_X \leq 5$ MHz and $f_{CPU} \leq 5$ MHz

<When the high-speed internal oscillation clock or external input clock are selected for the CPU clock> $f_{CPU} \leq 5$ MHz

3. When using the PLL, set RMC to 00H.

4. When transitioning to the STOP mode, it is possible to achieve low power consumption by setting RMC = 56H.

22.3 Cautions for Self Programming

1. Make sure that the regulator output voltage mode is fixed when executing self programming or EEPROM emulation.
2. Program area can be rewritten by using the self programming library in normal power mode.
3. Observe the following points when rewriting the flash memory in low power consumption mode:
   - Data area can be rewritten in low power consumption mode, but program area cannot.
     Data area: Flash memory area handled as data
     Program area: Flash memory area handled as the program
   - Flash memory that is erased and written in low power consumption mode cannot be accessed in normal power mode. To use this data in normal power mode, switch to low power consumption mode and transfer the flash memory contents to RAM.
   - A wait time of 2 ms is required before executing self programming after switching from normal power mode to low power consumption mode.

Remark For details of the self-programming function and the self programming library, refer to **"78K0 Microcontrollers User's Manual Self Programming Library Type 01 (U18274E)"** and **"78K0 Microcontrollers Self Programming Library Type 01 Ver. 3.10 Operating Precautions (notification document) (ZUD-CD-09-0122)"**.
For details of the EEPROM emulation library, refer to **"78K0 Microcontrollers User's Manual EEPROM Emulation Library Type 01 (U18275E)"** and **"78K0 Microcontrollers EEPROM Emulation Library Type 01 Ver.2.10 Operating Precautions (notification document) (ZUD-CD-09-0165)"**.

CHAPTER 23 OPTION BYTE

23.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the 78K0/Fx2-L microcontrollers 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.

(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

- LVI default start operation control
  - During LVI default start function enabled (LVISTART = 1)
    The device is in the reset state after reset release or upon power application and until the supply voltage reaches 1.91 V (TYP.). It is released from the reset state when the voltage exceeds 1.91 V (TYP.).
    If the supply voltage rises to 1.8 V after reset release or power application at a rate slower than 0.5 V/ms (MIN.), LVI default start function operation is recommended.
  - During LVI default start function stopped (LVISTART = 0)
    The device is in the reset state after reset release or upon power application and until the supply voltage reaches 1.61 V (TYP.). It is released from the reset state when the voltage exceeds 1.61 V (TYP.).

Caution LVISTART can only be written by using a dedicated flash memory programmer. It cannot be set or change during self-programming or boot swap operation during self-programming.

(3) 0082H/1082H

○ Internal high-speed oscillation clock frequency selection
- 4 MHz (TYP.)
- 8 MHz (TYP.)

Caution Set a value that is the same as that of 0082H to 1082H because 0082H and 1082H are switched during the boot swap operation.

(4) 0083H/1083H

○ Pin selection used during on-chip debugging
- TOOLC0/X1, TOOLD0/X2
- TOOLC1/P31, TOOLD1/P32

Caution Set a value that is the same as that of 0083H to 1083H because 0083H and 1083H are switched during the boot swap operation.

(5) 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 Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot swap operation.

23.2 Format of Option Byte

The format of the option byte is shown below.

Figure 23-1. Format of Option Byte (1/3)

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	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3.97 s)

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 0 (LSRSTOP) of the Internal oscillation mode/PLL control 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_{IL}: Internal low-speed oscillation clock frequency

2. ( ): f_{IL} = 33 kHz (MAX.)

Figure 23-1. Format of Option Byte (2/3)

Address: 0081H/1081H[Notes 1, 2]

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

LVISTART	LVI default start operation control
0	LVI is OFF by default upon power application (LVI default start function stopped)
1	LVI is ON by default upon power application (LVI default start function enabled)

Notes 1. LVISTART 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. 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 LVI default start, 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[Note]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	R4M8MSEL

R4M8MSEL	Internal high-speed oscillation clock frequency selection
0	8 MHz (TYP.)
1	4 MHz (TYP.)

Note Set a value that is the same as that of 0082H to 1082H because 0082H and 1082H are switched during the boot swap operation.

Caution Be sure to clear bits 7 to 1 to "0".

Figure 23-1. Format of Option Byte (3/3)

Address: 0083H/1083H[Note]

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

OCDPSEL	Pin selection used during on-chip debugging
0	TOOLC1/P31, TOOLD1/P32
1	TOOLC0/X1, TOOLD0/X2

Note Set a value that is the same as that of 0083H to 1083H because 0083H and 1083H are switched during the boot swap operation.

Caution Be sure to clear bits 7 to 5, and 0 to "0" and set bits 4 to 2 to "1".

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 Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot swap operation.

Caution Be sure to clear bits 7 to 2 to "0".

Remark For the on-chip debug security ID, refer to **CHAPTER 25 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: 2^7/fIL,
                         ; Internal low-speed oscillator can be stopped by software.
         DB     00H      ; LVI default start function stopped
         DB     01H      ; Internal high-speed oscillation clock frequency 4 MHz (TYP.)
         DB     1EH      ; Use the TOOLC0/X1, TOOLD0/X2 pins
         DB     02H      ; Operation enabled.  Does not erase data of the flash memory in case
                         ; authentication of the on-chip debug security ID fails.
```

Remark Referencing of the option byte is performed during reset processing. For the reset processing timing, refer to **CHAPTER 19 RESET FUNCTION**.

CHAPTER 24 FLASH MEMORY

The 78K0/Fx2-L microcontrollers incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

24.1 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 Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated Table 24-1 after release of reset.

Figure 24-1. 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
0	0	0	768 bytes
0	1	0	512 bytes
0	1	1	384 bytes
1	1	0	(Default value)
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	0	0	1	4 KB
0	0	1	0	8 KB
0	1	0	0	16 KB
1	1	1	1	(Default value)
Other than above				Setting prohibited

Table 24-1. Set Values of Internal Memory Size Switching Register

Products			IMS Setting
78K0/FY2-L	78K0/FA2-L	78K0/FB2-L	
μPD78F0854	μPD78F0857	–	61H
μPD78F0855	μPD78F0858	μPD78F0864	42H
μPD78F0856	μPD78F0859	μPD78F0865	04H

24.2 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/Fx2-L 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/Fx2-L microcontrollers are mounted on the target system.

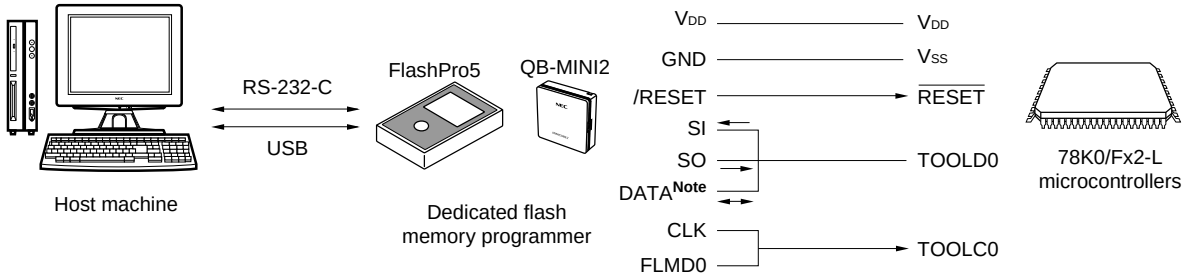
Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

24.3 Programming Environment

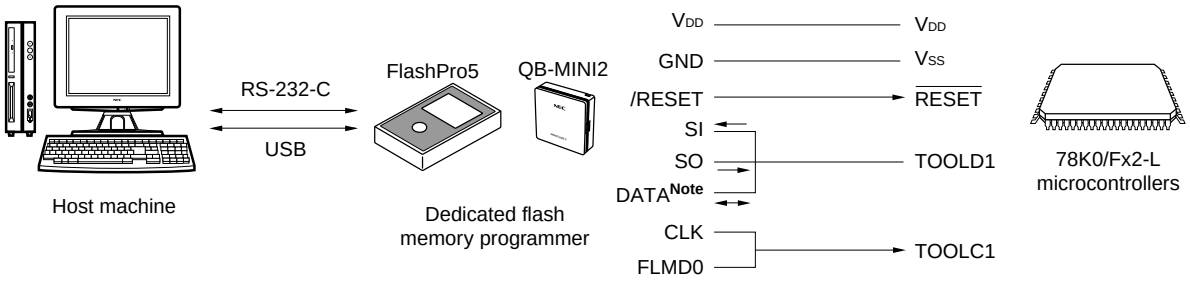
The environment required for writing a program to the flash memory of the 78K0/Fx2-L microcontrollers are illustrated below.

Figure 24-2. Environment for Writing Program to Flash Memory

(1) When using the TOOLC0 and TOOLD0 pins



(2) When using the TOOLC1 and TOOLD1 pins



Note QB-MINI2 only

A host machine that controls the dedicated flash memory programmer is necessary.

To interface between the dedicated flash memory programmer and the 78K0/Fx2-L microcontrollers, the TOOLD0 or TOOLD1 pins is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

Table 24-2. Pin Connection

Dedicated Flash memory programmer			78K0/Fx2-L microcontrollers
Signal Name	I/O	Pin Function	Pin Name
CLK	Output	Clock output to 78K0/Fx2-L microcontrollers	TOOLC0/TOOLC1
SI	Input	Receive signal	TOOLD0/TOOLD1
SO	Output	Transmit signal	
DATA**Note**	I/O	Input/output signal for data communication during debugging	
/RESET	Output	Reset signal	$\overline{\text{RESET}}$
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND	–	Ground	V_{SS}

Note QB-MINI2 only

24.4 Connection of Pins on Board

To write the flash memory on-board, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

24.4.1 TOOL pins

The pins used for communication in flash memory programming mode are shown in the table below.

Table 24-3. Pins Used for Communication in Flash Memory Programming Mode

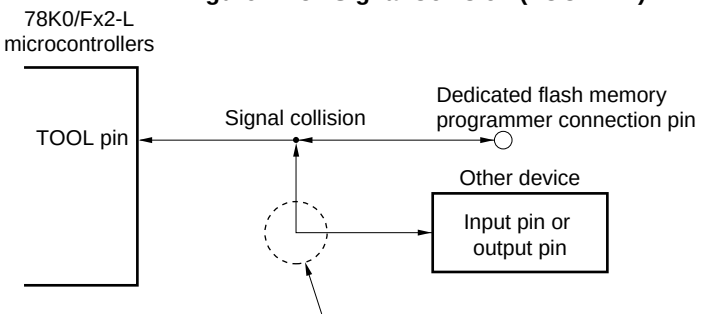
Pin Name	Connection of Pins
TOOLC0, TOOLC1	Connect this pin directly to the dedicated flash memory programmer or pull it down by connecting it to V_{SS} via a resistor (10 kΩ)
TOOLD0, TOOLD1	Connect this pin directly to the dedicated flash memory programmer or pull it up by connecting it to V_{DD} via a resistor (3 k to 10 kΩ)

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 is connected to the TOOL pin that is connected to another device, signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into a high-impedance state.

Figure 24-3. Signal Collision (TOOL Pin)



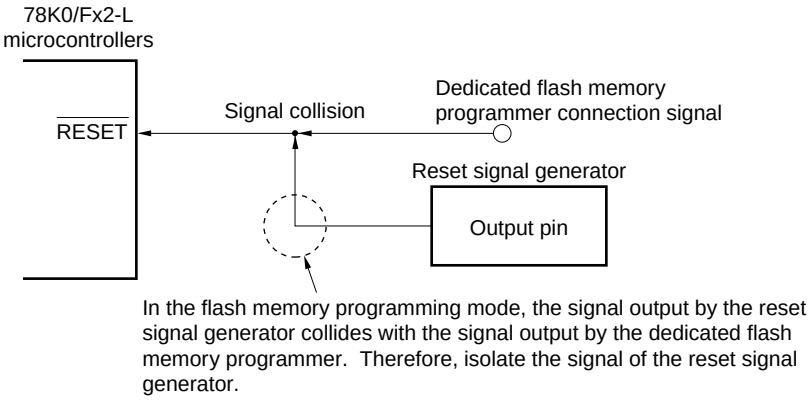
In the flash memory programming mode, the signal of the other device collides with the signal of the dedicated flash programmer. Therefore, isolate the signal of the other device.

24.4.2 $\overline{\text{RESET}}$ 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 24-4. Signal Collision ($\overline{\text{RESET}}$ Pin)



24.4.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

24.4.4 REGC pin

Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

24.4.5 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the internal high-speed oscillation clock (f_{IH}) is used.

24.4.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

However, 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 (AV_{REF} and AV_{SS}) as those in the normal operation mode.

24.4.7 On-board writing when connecting crystal/ceramic resonator

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 processed as described below.

The state of the pins in the self programming mode is the same as that in the HALT mode.

When using the X1 (TOOLC0) and X2 (TOOLD0) pins as the serial interface for flash memory programming, signals will collide if an external device is connected. To prevent the conflict of signals, isolate the connection with the external device.

Similarly, when a capacitor is connected to the X1 and X2 pins, the waveform during communication is changed, and thus communication may be disabled depending on the capacitor capacitance. Make sure to isolate the connection with the capacitor during flash programming.

In cases when a crystal or ceramic resonator has been selected to generate the system clock, and the decision has been made to execute on-board flash programming with the resonator mounted on the device because it is difficult to isolate the resonator, be sure to thoroughly evaluate the flash memory programming with the resonator mounted on the device before executing the processing described next.

- Mount the minimum-possible test pads between the device and the resonator, and connect the programmer via the test pad. Keep the wiring as short as possible (refer to **Figure 24-5 and Table 24-4**).

Figure 24-5. Example of Mounting Test Pads

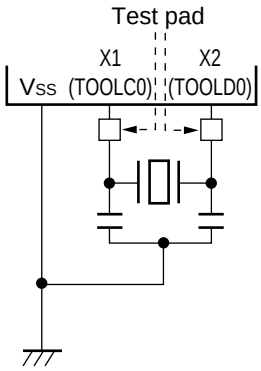


Table 24-4. Clock to Be Used and Mounting of Test Pads

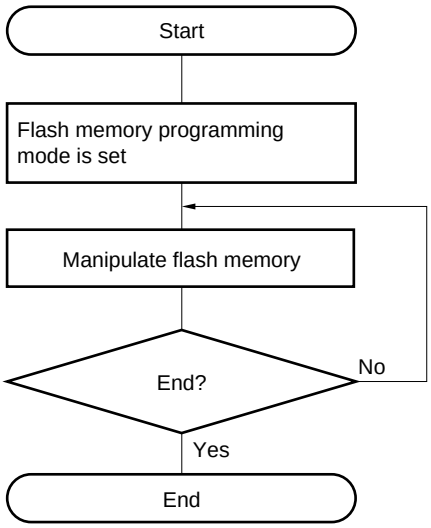
Clock to Be Used		Mounting of Test Pads
High-speed internal oscillation clock		Not required
External clock		
Crystal/ceramic oscillation clock	Before resonator is mounted	
	After resonator is mounted	Required

24.5 Programming Method

24.5.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 24-6. Flash Memory Manipulation Procedure



24.5.2 Flash memory programming mode

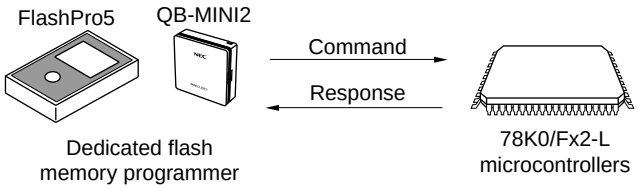
To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0/Fx2-L microcontrollers in the flash memory programming mode. The system switches to the flash memory programming mode once the dedicated flash memory programmer is connected and communication starts.

Change the mode by using a jumper when writing the flash memory on-board.

24.5.3 Communication commands

The 78K0/Fx2-L microcontrollers communicate with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0/Fx2-L microcontrollers are called commands, and the signals sent from the 78K0/Fx2-L microcontrollers to the dedicated flash memory programmer are called response.

Figure 24-7. Communication Commands



The flash memory control commands of the 78K0/Fx2-L microcontrollers are listed in the table below. All these commands are issued from the programmer and the 78K0/Fx2-L microcontrollers perform processing corresponding to the respective commands.

Table 24-5. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Chip Erase	Erases the entire flash memory.
	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Silicon Signature	Gets 78K0/Fx2-L information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0/Fx2-L 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.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

The 78K0/Fx2-L microcontrollers return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0/Fx2-L microcontrollers are listed below.

Table 24-6. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

24.6 Security Settings

The 78K0/Fx2-L 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.

The batch erase (chip erase), block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by on-board/off-board programming and self programming. Each security setting can be used in combination.

All the security settings are cleared by executing the batch erase (chip erase) command.

Table 24-7 shows the relationship between the erase and write commands when the 78K0/Fx2-L microcontroller security function is enabled.

Table 24-7. 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 24-8 shows how to perform security settings in each programming mode.

Table 24-8. 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 set information library.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of writing		
Prohibition of rewriting boot cluster 0		

24.7 Processing Time for Each Command When PG-FP5 Is Used (Reference)

The following table shows the processing time for each command (reference) when the PG-FP5 is used as a dedicated flash memory programmer.

Table 24-9. Processing Time for Each Command When PG-FP5 Is Used (Reference) (1/2)

(1) Products with internal ROMs of the 4 KB: µPD78F0854, 78F0857

Command of PG-FP5	Port: UART-Internal-OSC (Internal high-speed oscillation clock (f_{IH}: 8 MHz (typ.)), Speed: 500,000 bps
Signature	0.5 s (typ.)
Blankcheck	0.5 s (typ.)
Erase	0.5 s (typ.)
Program	1 s (typ.)
Verify	1 s (typ.)
E.P.V	1 s (typ.)
Checksum	0.5 s (typ.)
Security	0.5 s (typ.)

(2) Products with internal ROMs of the 8 KB: µPD78F0855, 78F0858, 78F0864

Command of PG-FP5	Port: UART-Internal-OSC (Internal high-speed oscillation clock (f_{IH}: 8 MHz (typ.)), Speed: 500,000 bps
Signature	0.5 s (typ.)
Blankcheck	0.5 s (typ.)
Erase	1 s (typ.)
Program	1.5 s (typ.)
Verify	1 s (typ.)
E.P.V	1.5 s (typ.)
Checksum	0.5 s (typ.)
Security	0.5 s (typ.)

Caution When executing boot swapping, do not use the E.P.V. command with the dedicated flash memory programmer.

Table 24-9. Processing Time for Each Command When PG-FP5 Is Used (Reference) (2/2)

(3) Products with internal ROMs of the 16 KB: μPD78F0856, 78F0859, 78F0865

Command of PG-FP5	Port: UART-Internal-OSC (Internal high-speed oscillation clock (f_{IH}: 8 MHz (typ.)), Speed: 500,000 bps
Signature	0.5 s (typ.)
Blankcheck	0.5 s (typ.)
Erase	1 s (typ.)
Program	2.5 s (typ.)
Verify	1.5 s (typ.)
E.P.V	2.5 s (typ.)
Checksum	1 s (typ.)
Security	0.5 s (typ.)

Caution When executing boot swapping, do not use the E.P.V. command with the dedicated flash memory programmer.

24.8 Flash Memory Programming by Self-Programming

The 78K0/Fx2-L microcontrollers support a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the 78K0/Fx2-L microcontroller self-programming library, it can be used to upgrade the program in the field.

If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. If an unmasked interrupt request is generated in the EI state, the request branches directly from the self-programming library to the interrupt routine. After the self-programming mode is later restored, self-programming can be resumed. However, the interrupt response time is different from that of the normal operation mode.

Cautions 1. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.

2. Make sure that the regulator output voltage mode is fixed when executing self programming or EEPROM emulation.

3. Program area can be rewritten by using the self programming library in normal power mode.

4. Observe the following points when rewriting the flash memory in low power consumption mode:

- **Data area can be rewritten in low power consumption mode, but program area cannot.**
  Data area: Flash memory area handled as data
  Program area: Flash memory area handled as the program
- **Flash memory that is erased and written in low power consumption mode cannot be accessed in normal power mode. To use this data in normal power mode, switch to low power consumption mode and transfer the flash memory contents to RAM.**
- **Blocks cannot be overwritten by using the self programming library. Be sure to erase a block first before rewriting data to it.**
- **A wait time of 2 ms is required before executing self programming after switching from normal power mode to low power consumption mode.**

Remark For details of the self-programming function and the self programming library, refer to **"78K0 Microcontrollers User's Manual Self Programming Library Type 01 (U18274E)"** and **"78K0 Microcontrollers Self Programming Library Type 01 Ver. 3.10 Operating Precautions (notification document) (ZUD-CD-09-0122)"**.

For details of the EEPROM emulation library, refer to **"78K0 Microcontrollers User's Manual EEPROM Emulation Library Type 01 (U18275E)"** and **"78K0 Microcontrollers EEPROM Emulation Library Type 01 Ver.2.10 Operating Precautions (notification document) (ZUD-CD-09-0165)"**.

24.8.1 Register controlling self programming mode

The self programming mode is controlled by the self programming mode control register (FPCTL).

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

Reset signal generation clears FPCTL to 00H.

Figure 24-8. Format of Self Programming Mode Control Register (FPCTL)

Address: FF2BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FPCTL	0	0	0	0	0	0	0	FLMDPUP **Note**

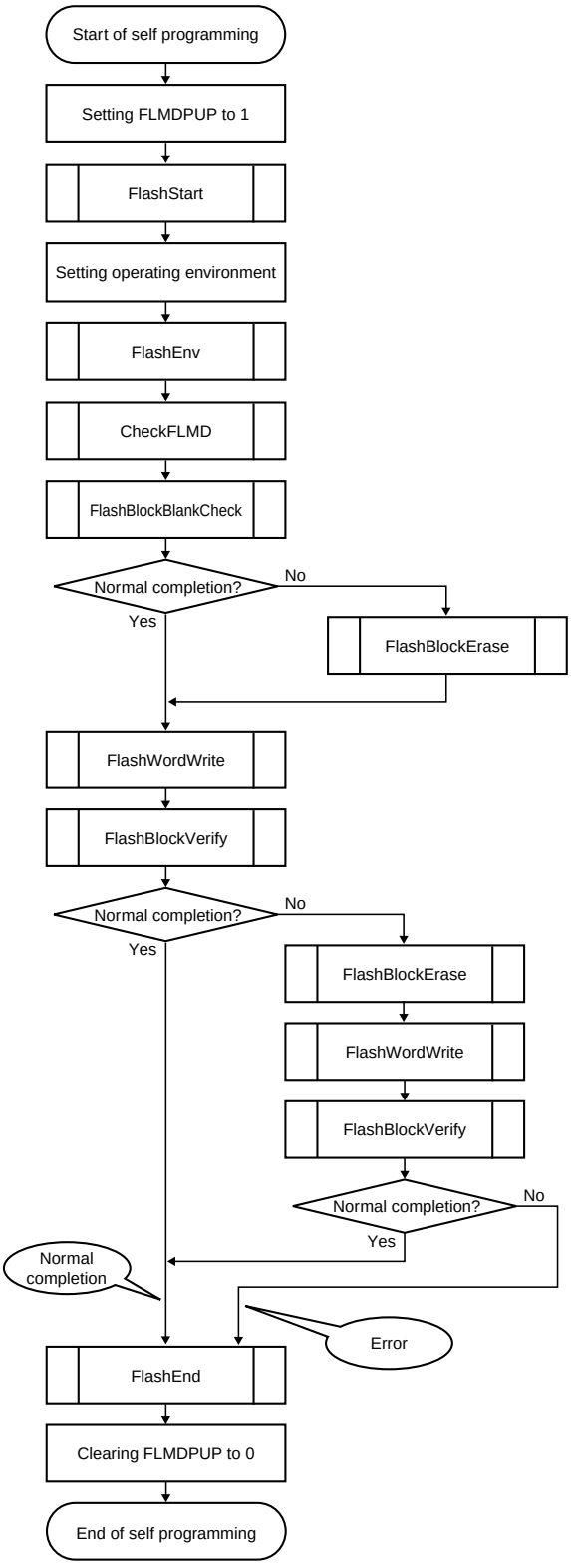
FLMDPUP **Note**	Self programming mode control
0	Normal operation mode
1	Self programming mode

Note The FLMDPUP bit must be set to 0 (normal operation mode) while the regular user program is being executed, and set to 1 (self programming mode) while self programming is being executed. The flash memory rewrite circuit does not operate in normal operation mode, so even though the firmware and software for rewriting will work, no actual rewriting will take place.

24.8.2 Flow of self programming (Rewriting Flash Memory)

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 24-9. Flow of Self Programming (Rewriting Flash Memory)



Remark For details of the self programming function and the self programming library, refer to **"78K0 Microcontrollers User's Manual Self Programming Library Type 01 (U18274E)"** and **"78K0 Microcontrollers Self Programming Library Type 01 Ver. 3.10 Operating Precautions (notification document) (ZUD-CD-09-0122)"**.

24.8.3 Boot swap function

If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

The boot swap function is used to avoid this problem.

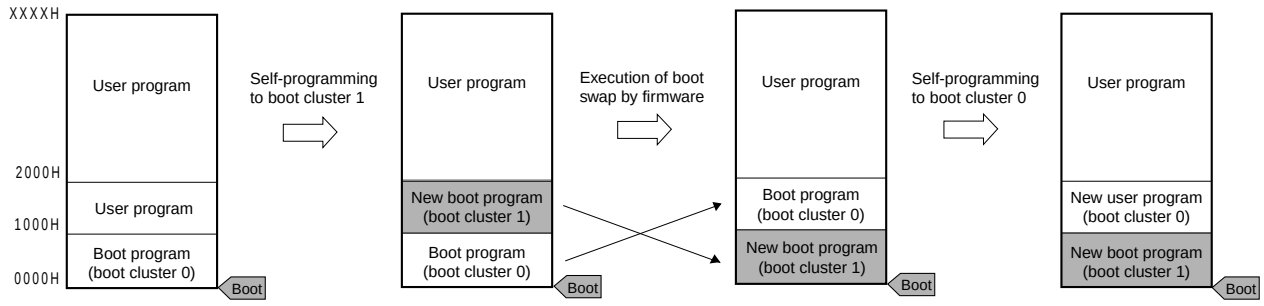
Before erasing boot cluster 0[Note], which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the 78K0/Fx2-L microcontrollers, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0.

As a result, even if a power failure occurs while the boot programming area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Caution The products whose ROM size is 4 KB can not use the boot swap function.

Figure 24-10. Boot Swap Function

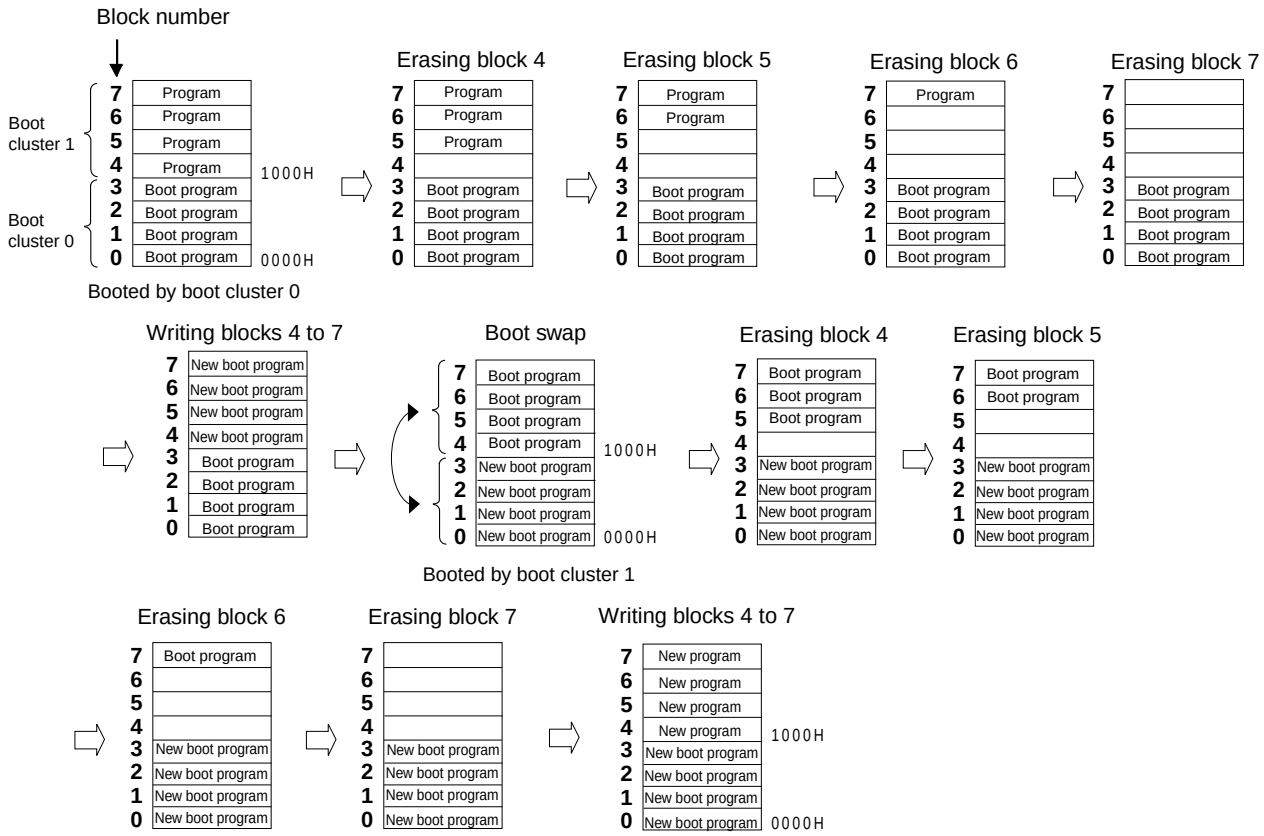


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 24-11. Example of Executing Boot Swapping



24.9 Creating ROM Code to Place Order for Previously Written Product

Before placing an order with Renesas Electronics for a previously written product, the ROM code for the order must be created.

To create the ROM code, use the Hex Consolidation Utility (hereafter abbreviated to HCU) on the finished programs (hex files) and optional data (such as security settings for flash memory programs).

The HCU is a software tool that includes functions required for creating ROM code.

The HCU can be downloaded at the Renesas Electronics website.

(1) Website

http://www2.renesas.com/micro/en/ods/ → Click Version-up Service.

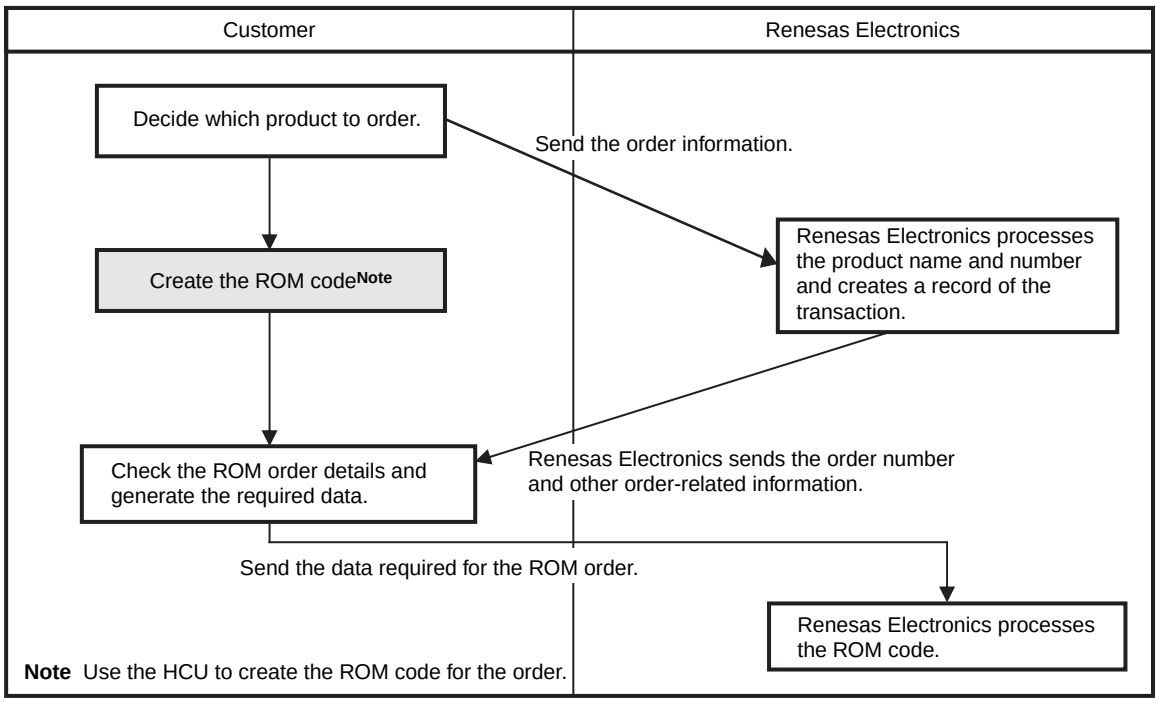
(2) Downloading the HCU

To download the HCU, click Software for previously written flash products and then HCU_GUI.

Remark For details about how to install and use the HCU, see the materials (the user's manual) that comes with the HCU at the above website.

24.9.1 Procedure for using ROM code to place an order

Use the HCU to create the ROM code by following the procedure below, and then place your order with Renesas Electronics. For details, see the ROM Code Ordering Method Information (C10302J).



CHAPTER 25 ON-CHIP DEBUG FUNCTION

25.1 Connecting QB-MINI2 to 78K0/Fx2-L Microcontrollers

The 78K0/Fx2-L microcontrollers use the V_{DD}, $\overline{RESET}$, TOOLC0/X1 (or TOOLC1/P31), TOOLD0/X2 (or TOOLD1/P32), and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2). Whether TOOLC0/X1 and TOOLC1/P31, or TOOLD0/X2 and TOOLD1/P32 are used can be selected.

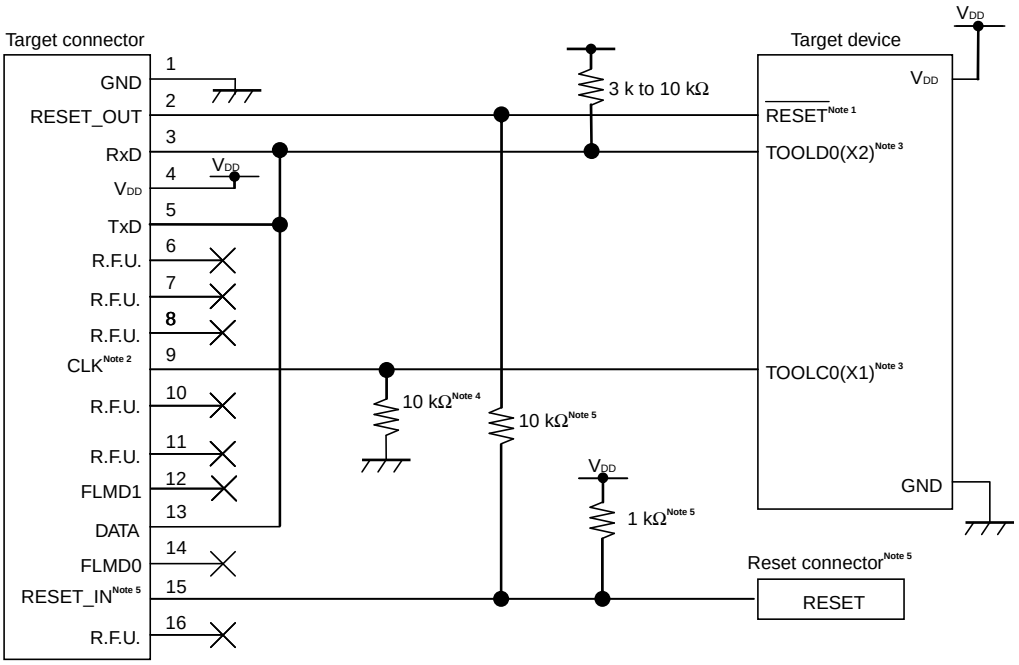
Cautions 1. The 78K0/Fx2-L microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. When transitioning to STOP mode during on-chip debugging, oscillation of the internal high-speed oscillator continues, but the on-chip debug operation is not affected.

3. If operating in the standalone mode after writing to a load module file (extension: *.lnk or *.lmf) that has debugging information, pull up TOOLD0. Note that operation is not guaranteed in this environment.

Figure 25-1. Connection Example of QB-MINI2 and 78K0/Fx2-L Microcontrollers (1/3)

(1) When using the TOOLC0 and TOOLD0 pins (X1 oscillator or EXCLK input clock is not used, both debugging and programming are performed)



Notes 1. If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.

2. A clock signal provided on the 78K0-OCD board, a 4, 8, or 16 MHz clock signal generated in QB-MINI2, or the clock signal generated by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.
Only the internal high-speed oscillator of the device can be used during flash programming.

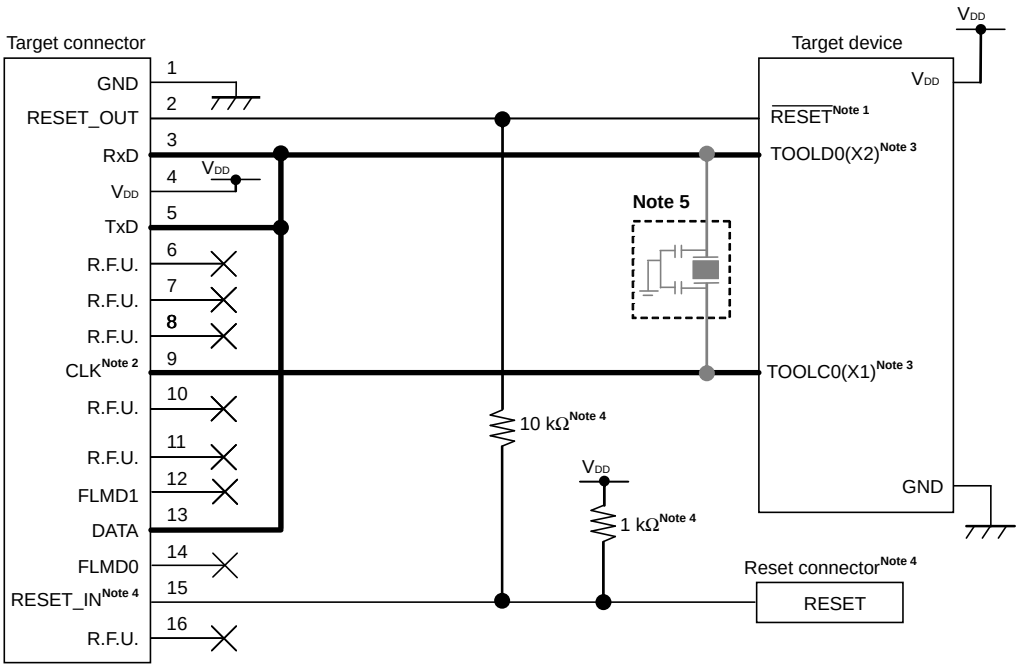
3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when QB-MINI2 is not connected.)

4. This is the processing for the pin that is unused (the input is left open) when the target device operates (when QB-MINI2 is not connected). (This processing is not required if an oscillator circuit is used.)

5. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less).

Figure 25-1. Connection Example of QB-MINI2 and 78K0/Ix2 Microcontrollers (2/3)

(2) When using the TOOLC0 and TOOLD0 pins (with X1/X2 oscillator is used, both debugging and programming are performed)



Notes 1. If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.

2. A clock signal provided on the 78K0-OCD board, a 4, 8, or 16 MHz clock signal generated in QB-MINI2, or the clock signal generated by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.
Only the internal high-speed oscillator of the device can be used during flash programming.

3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when QB-MINI2 is not connected.)

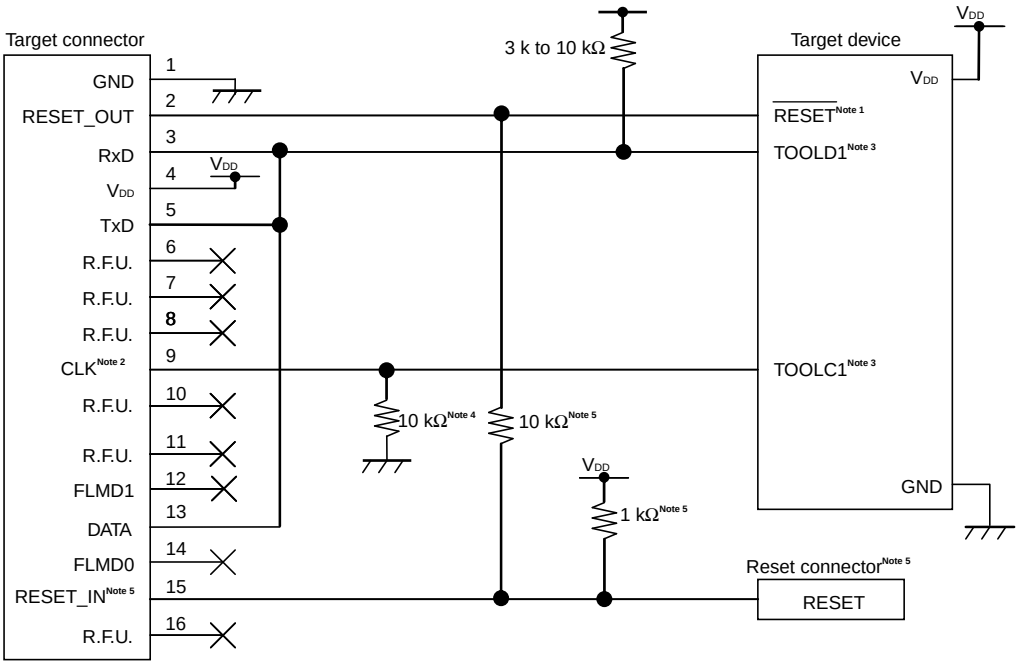
4. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin** of QB-MINI2 On-Chip Debug Emulator with Programming Function (18371E).

5. Never connect an oscillation circuit to the 78K0-OCD board during on-chip debugging and flash programming.
To prevent an oscillation circuit from not oscillating due to wiring capacitance when the target device operates (when QB-MINI2 is not connected), also consider countermeasures such as disconnecting the oscillation circuit from the target connectors by setting the jumpers.
A program that was downloaded using the debugger does not operate when QB-MINI2 is not connected.

Caution The bold lines in the figure (TOOLD0 and TOOLC0) must be designed so that the device pins are less than 30 mm from the QB-MINI2 connectors or the paths must be shielded by connecting them to GND.

Figure 25-1. Connection Example of QB-MINI2 and 78K0/Fx2-L Microcontrollers (3/3)

(3) When using the TOOLC1 and TOOLD1 pins (both debugging and programming are performed)



Notes 1. If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.

2. The clock signal generated by the clock circuit on the target system or by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.
Only the internal high-speed oscillator of the device can be used during flash programming.

3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when QB-MINI2 is not connected.)

4. This is the processing for the pin that is unused (the input is left open) when the target device operates (when QB-MINI2 is not connected). (This processing is not required if the pin is set to output.)

5. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less).

25.2 On-Chip Debug Security ID

The 78K0/Fx2-L microcontrollers have an on-chip debug operation control bit in the flash memory at 0084H (refer to **CHAPTER 23 OPTION BYTE**) and an on-chip debug security ID setting area at 0085H to 008EH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 1084H and 1085H to 108EH in advance, because 0084H, 0085H to 008EH and 1084H, and 1085H to 108EH are switched.

For details on the on-chip debug security ID, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

Table 25-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
0085H to 008EH	Any ID code of 10 bytes
1085H to 108EH	

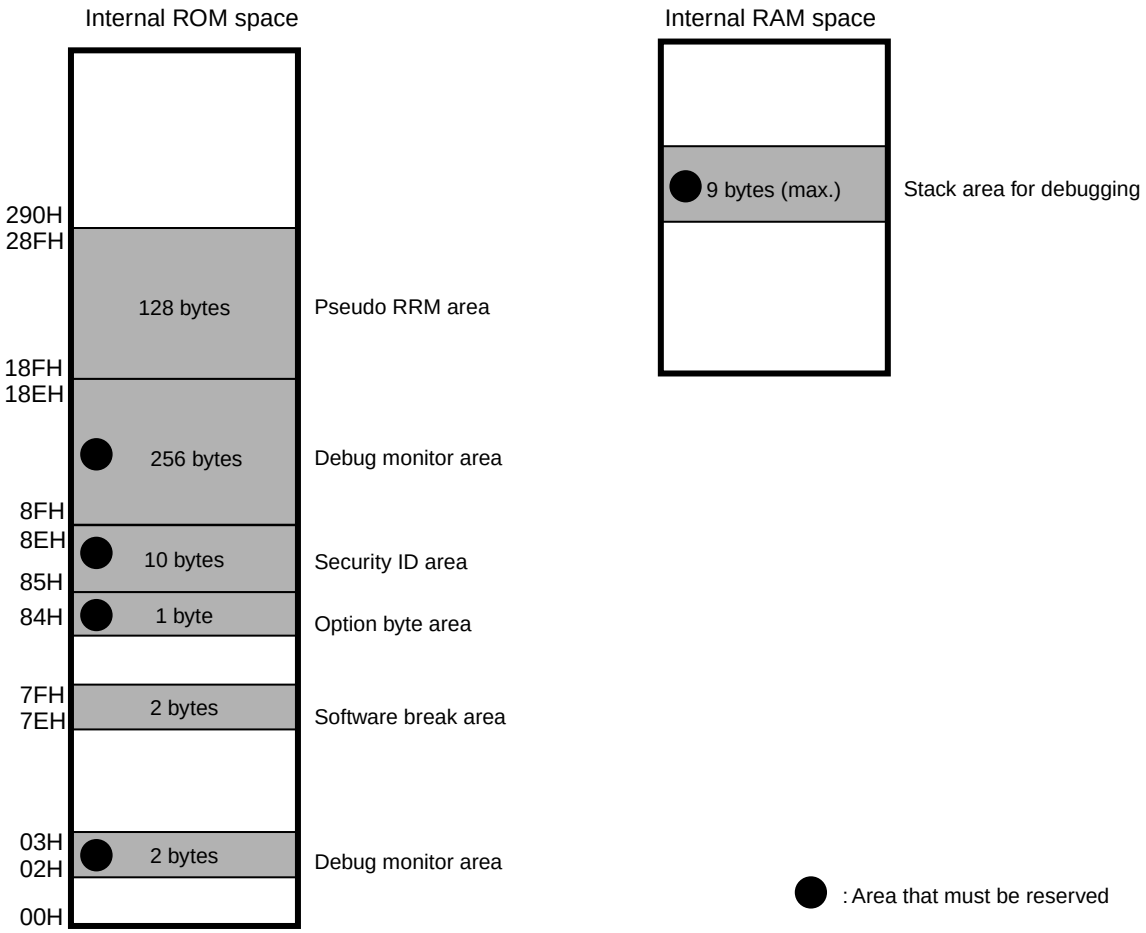
25.3 Securing of User Resources

QB-MINI2 uses the user memory spaces (shaded portions in Figure 25-2) to implement communication with the target device, or each debug functions. The areas marked with a dot (•) are always used for debugging, and other areas are used for each debug function used.

These areas can be secured by using user programs or the linker option.

For details on the securing of these areas, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

Figure 25-2. Reserved Area Used by QB-MINI2



CHAPTER 26 INSTRUCTION SET

This chapter lists each instruction set of the 78K0/Fx2-L 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)**.

26.1 Conventions Used in Operation List

26.1.1 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 26-1. 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, refer to **Table 3-6 Special Function Register List**.

26.1.2 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)
∨: Exclusive logical sum (exclusive OR)
$\overline{\ \ }$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

26.1.3 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

26.2 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) + 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	× × ×

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.

26.3 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

First Operand \ Second 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

Second Operand / First 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 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)

Target products: 78K0/FY2-L: μPD78F0854(A), 78F0855(A), 78F0856(A)
78K0/FA2-L: μPD78F0857(A), 78F0858(A), 78F0859(A)
78K0/FB2-L: μPD78F0864(A), 78F0865(A)

Cautions 1. The 78K0/Fx2-L microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product as follows.

(1) Port functions

Port	78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
	16 Pins	20 Pins	30 Pins
Port 0	P00, P01		P00 to P02
Port 2	P20 to P23	P20 to P25	P20 to P27
Port 3	P30	P30 to P32	P30 to P37
Port 6	P60, P61		
Port 7	–		P70
Port 12	P121, P122		

(The remaining table is on the next page.)

(2) Non-port functions

Function		78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
		16 Pins	20 Pins	30 Pins
Power supply, ground		V_{DD}, V_{SS}, AV_{REF}		V_{DD}, AV_{REF}, V_{SS}, AV_{SS}
Regulator		REGC		
Reset		$\overline{\text{RESET}}$		
Clock oscillation		X1, X2, EXCLK		
Interrupt		INTP0, INTP1	INTP0 to INTP3	INTP0 to INTP5
Timer	TMX0	–	TOX00, TOX01	TOX00, TOX01, TOX10, TOX11
	TM00	TI000, TI010, TO00		
	TM51	TI51		
	TMH1	TOH1		
Serial interface	UART6	RxD6, TxD6		
	IICA	SCLA0, SDAA0		
	CSI11	–		$\overline{\text{SCK11}}$, SI11, SO11, $\overline{\text{SSI11}}$
A/D converter		ANI0 to ANI3	ANI0 to ANI5	ANI0 to ANI8
Comparator		CMP2+	CMP0+ to CMP2+	CMP0+ to CMP2+, CMPCOM
On-chip debug function		TOOLC0, TOOLC1	TOOLC0, TOOLC1, TOOLD0, TOOLD1	

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
	V_{SS}		–0.5 to +0.3	V
	AV_{REF}		–0.5 to V_{DD} + 0.3 Note 1	V
	AV_{SS}		–0.5 to +0.3	V
REGC pin input voltage Note 2	V_{IREGC}		–0.5 to +3.6 and –0.3 to V_{DD} +0.3	V
Input voltage	V_{I1}	P00 to P02, P30 to P37, P60, P61, P121, P122, X1, X2, $\overline{\text{RESET}}$	–0.3 to V_{DD} + 0.3 Note 1	V
	V_{I2}	P20 to P27, P70	–0.3 to AV_{REF} + 0.3 Note 1 and –0.3 to V_{DD} + 0.3 Note 1	V
Output voltage	V_{O1}	P00 to P02, P30 to P37, P60, P61	–0.3 to V_{DD} + 0.3 Note 1	V
	V_{O2}	P20 to P27, P70	–0.3 to AV_{REF} + 0.3 Note 1	V
Analog input voltage	V_{AN}	ANI0 to ANI8	–0.3 to AV_{REF} + 0.3 Note 1 and –0.3 to V_{DD} + 0.3 Note 1	V

Notes 1. Must be 6.5 V or lower.

2. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

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 P02, P30 to P37, P60, P61	−10	mA
		Total of all pins −80 mA	P02, P60, P61	−30	mA
			P00, P01, P30 to P37	−55	mA
	I_{OH2}	Per pin	P20 to P27, P70	−0.5	mA
		Total of all pins		−2	mA
Output current, low	I_{OL1}	Per pin	P00 to P02, P30 to P37, P60, P61	30	mA
		Total of all pins 200 mA	P02, P60, P61	85	mA
			P00, P01, P30-P37	140	mA
	I_{OL2}	Per pin	P20 to P27, P70	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A			−40 to +85	°C
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.

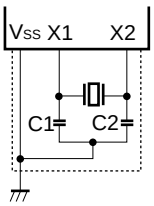
2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

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} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic/ crystal resonator		X1 clock oscillation frequency (f_X)Note	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		10.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

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 High-speed Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Internal high-speed oscillator	Oscillation frequency (f_{IH} = 4 MHz) deviation[Notes 1, 2]	RSTS = 1	T_A = –20 to +70°C			±2	%
			T_A = –40 to +85°C			±3	%
	Oscillation frequency (f_{IH} = 8 MHz) deviation[Notes 1, 2]		T_A = –40 to +85°C			±3	%
	Oscillation frequency (f_{IH})[Note 1]	RSTS = 0	In low power consumption mode (RMC = 56H)	1.86	4.2	7.42	MHz
			In normal power mode (RMC = 00H)	1.86	5.0	8.7	MHz

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Internal high-speed oscillation frequency (4 MHz or 8 MHz) is set by the option byte. Refer to **CHAPTER 23 OPTION BYTE**.

PLL Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
PLL input clock frequency	f_{PLLIN}	High-speed system clock is selected (f_X = 4 MHz)		3.92	4.00	4.08	MHz
		Internal high-speed oscillation clock is selected (f_{IH} = 4 MHz)	T_A = –20 to +70°C	3.92	4.00	4.08	MHz
			T_A = –40 to +85°C	3.88	4.00	4.12	MHz
PLL output clock frequency	f_{PLL}			f_{PLLIN} × 10			MHz
Long-term jitter	T_{LJ}	f_{PLL} = 40 MHz, more than 400 counts			0.0029	0.07	%f_{PLL}

Internal Low-speed Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions			MIN.	TYP.	MAX.	Unit
Internal low-speed oscillator	Oscillation clock frequency (f_{IL})	Trimming	In low power consumption mode (RMC = 56H)		25.5	30	34.5	kHz
			In normal power mode (RMC = 00H)	2.7 V ≤ V_{DD} < 5.5 V	27	30	33	kHz
				1.8 V ≤ V_{DD} < 2.7 V	25.5	30	34.5	kHz

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (1/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P02, P30 to P37, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0	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 P02, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			–9.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–7.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			–3.0	mA
		Total of P00, P01, P30 to P37	4.0 V ≤ V_{DD} ≤ 5.5 V			–24.0	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 P00 to P02, P30 to P37, P60, P61[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			–33.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–26.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			–13.0	mA
	I_{OH2}	Per pin for P20 to P27, P70	AV_{REF} = V_{DD}			–100	μA
Output current, low[Note 2]	I_{OL1}	Per pin for P00 to P02, P30 to P37	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, P61	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 P02, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			23.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			6.0	mA
		Total of P00, P01, P30 to P37	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 P00 to P02, P30 to P37, P60, P61, P122[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			68.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			50.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			26.0	mA
	I_{OL2}	Per pin for P20 to P27, P70	AV_{REF} = V_{DD}			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} = –12 mA

Total output current of pins = (–12 × 0.7)/(50 × 0.01) = –16.8 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 (2/5)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P37, P122		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P20 to P27, P70	AV_{REF} = V_{DD}	0.7AV_{REF}		AV_{REF}	V
	V_{IH3}	P60, P61 (I/O port mode)		0.7V_{DD}		V_{DD}	V
	V_{IH4}	P00 to P02, P30 to P36, $\overline{\text{RESET}}$, EXCLK		0.8V_{DD}		V_{DD}	V
	V_{IH5}	P60, P61 (SMBus input mode)	2.4 V ≤ V_{DD} < 3.4 V	2.1			V
			1.8 V ≤ V_{DD} < 2.4 V, 3.4 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}			V
	V_{IH6}	P121/<TI000>/<INTP0>	When switching with alternate function is disabled (default), INTP0SEL0 = TM00SEL0 = 0	0.7V_{DD}		V_{DD}	V
			When switching with alternate function is enabled (78K0/FB2-L only), INTP0SEL0 = 1 or TM00SEL0 = 1	0.7V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P37, P122		0		0.3V_{DD}	V
	V_{IL2}	P20 to P27, P70	AV_{REF} = V_{DD}	0		0.3AV_{REF}	V
	V_{IL3}	P60, P61 (I/O port mode)		0		0.3V_{DD}	V
	V_{IL4}	P00 to P02, P30 to P36, $\overline{\text{RESET}}$, EXCLK		0		0.2V_{DD}	V
	V_{IL5}	P60, P61 (SMBus input mode)	2.4 V ≤ V_{DD} < 3.4 V	0		0.8	V
			1.8 V ≤ V_{DD} < 2.4 V, 3.4 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
	V_{IL6}	P121/<TI000>/<INTP0>	When switching with alternate function is disabled (default), INTP0SEL0 = TM00SEL0 = 0	0		0.4V_{DD}	V
			When switching with alternate function is enabled (78K0/FB2-L only), INTP0SEL0 = 1 or TM00SEL0 = 1	0		0.3V_{DD}	V
Output voltage, high	V_{OH1}	P00 to P02, P30 to P37, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} < 5.5 V, I_{OH1} = –2.5 mA	V_{DD} – 0.5			V
			1.8 V ≤ V_{DD} < 5.5 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27, P70	AV_{REF} = V_{DD}, I_{OH2} = –100 μA	V_{DD} – 0.5			V

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 (3/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, low	V_{OL1}	P00 to P02, P30 to P37	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} < 5.5 V, I_{OL1} = 5.0 mA			0.7	V
			1.8 V ≤ V_{DD} < 5.5 V, I_{OL1} = 2.0 mA			0.5	V
			1.8 V ≤ V_{DD} < 5.5 V, I_{OL1} = 1.0 mA			0.5	V
			1.8 V ≤ V_{DD} < 5.5 V, I_{OL1} = 0.5 mA			0.4	V
	V_{OL2}	P20 to P27, P70	AV_{REF} = V_{DD}, I_{OL2} = 0.4 mA			0.4	V
	V_{OL3}	P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OL1} = 5.0 mA			0.6	V
			2.7 V ≤ V_{DD} < 5.5 V, I_{OL1} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} < 5.5 V, I_{OL1} = 2.0 mA			0.4	V
Input leakage current, high	I_{LIH1}	P00 to P02, P30 to P37, P60, P61, $\overline{\text{RESET}}$	V_I = V_{DD}			1	μA
	I_{LIH2}	P20 to P27, P70	V_I = AV_{REF} = V_{DD}			1	μA
	I_{LIH3}	P121, P122	V_I = V_{DD} I/O port mode			1	μA
		X1, X2	OSC mode			20	μA
Input leakage current, low	I_{LIL1}	P00 to P02, P30 to P37, P60, P61, $\overline{\text{RESET}}$	V_I = V_{SS}			–1	μA
	I_{LIL2}	P20 to P27, P70	V_I = V_{SS}, AV_{REF} = V_{DD}			–1	μA
	I_{LIL3}	P121, P122	V_I = V_{SS} I/O port mode			–1	μA
		X1, X2	OSC mode			–20	μA
<R> Pull-up resistor	R_{PLU1}	P00-P02, P30 to P37, P60, P61	V_I = V_{SS}	10	20	100	kΩ
	R_{PLU2}	$\overline{\text{RESET}}$		75	150	300	kΩ

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 (4/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}[Note 2]	Operating mode	f_{XH} = 20 MHz[Note 4], V_{DD} = 5.0 V, RMC = 00H	Square wave input		3.2	5.5	mA
				Resonator connection		4.5	6.9	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V, RMC = 00H	Square wave input		1.6	2.8	mA
				Resonator connection		2.3	3.9	mA
			f_{XH} = 10 MHz, V_{DD} = 3.0 V, RMC = 00H	Square wave input		1.5	2.7	mA
				Resonator connection		2.2	3.2	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V, RMC = 00H	Square wave input		0.9	1.6	mA
				Resonator connection		1.3	2.0	mA
			f_{XH} = 5 MHz, V_{DD} = 2.0 V, RMC = 00H	Square wave input		0.7	1.4	mA
				Resonator connection		1.0	1.6	mA
			f_{IH} = 4 MHz, V_{DD} = 3.0 V, RMC = 56H			0.5	1.4	mA
			f_{IH} = 8 MHz, V_{DD} = 5.0 V, RMC = 00H			1.3	2.5	mA
			f_{IH} = 4 MHz, f_{CPU} = 1 MHz,[Note 5] V_{DD} = 3.0 V, RMC = 56H			0.22	0.65	mA
			f_{PLL} = 40 MHz, f_{CPU} = 20 MHz, V_{DD} = 5.0 V, RMC = 00H	f_{XH} = 4 MHz	Square wave input	4.5	8.0	mA
					Resonator connection	4.8	9.2	mA
				f_{IH} = 4 MHz Note 6	Internal oscillation	4.5	8.0	mA
	I_{DD2}[Note 3]	HALT mode	f_{XH} = 20 MHz[Note 4], V_{DD} = 5.0 V, RMC = 00H	Square wave input		0.8	2.6	mA
				Resonator connection		2.0	4.4	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V, RMC = 00H	Square wave input		0.4	1.3	mA
				Resonator connection		1.0	2.4	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V, RMC = 00H	Square wave input		0.2	0.65	mA
				Resonator connection		0.5	1.1	mA
			f_{IH} = 4 MHz, V_{DD} = 3.0 V, RMC = 56H			0.2	0.5	mA
			f_{IH} = 8 MHz, V_{DD} = 5.0 V, RMC = 00H			0.3	1.2	mA
			f_{CPU} = 20 MHz, V_{DD} = 5.0 V, RMC = 00H	f_{XH} = 4 MHz	Square wave input	1.3	3.0	mA
					Resonator connection	1.6	4.6	mA
				f_{IH} = 4 MHz Note 6	Internal oscillation	1.3	3.0	mA

(**Note** is listed on next page.)

Notes 1. Total current flowing into the internal power supply (V_{DD}, AV_{REF}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors, the pull-down resistors and the output current of the port are not included.

2. Not including the current flowing into the oscillator other than the circuit which generates the clock supplied to CPU. Not including the current flowing into the LVI circuit, A/D converter, comparator, watchdog timer, and serial interface.

3. Not including the current flowing into the oscillation circuit other than the circuit which generates the clock supplied to CPU. Not including the current flowing into the LVI circuit, A/D converter, comparator, watchdog timer, 16-bit timer X0, X1, 8-bit timer/event counter51, 8-bits timer H1 (When using the 30 kHz internal low-speed oscillation clock as the count clock), and serial interface.

4. When OSCCTL.AMPH is set to 1

5. This value is when PCC2 = 0, PCC1 = 1, and PCC0 = 0.

6. Internal high-speed oscillation clock frequency is set by using the option byte (R4M8MSEL = 0: 8 MHz, R4M8MSEL = 1: 4 MHz)

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 (5/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD3}[Note 3]	STOP mode	V_{DD} = 3.0 V (regulator output for internal operation + POC only), RMC = 56H	T_A = –40 to +50°C		0.3	2.7	μA
				T_A = –40 to +70°C		0.3	3.7	μA
				T_A = –40 to +85°C		0.3	5.5	μA
Watchdog timer operating current[Note 3]	I_{WDT}	In 30 kHz internal low-speed oscillation clock operation		V_{DD} = 3.0 V		0.28	0.35	μA
TMH1 operating current[Note 4]	I_{TMH}	When using the 30 kHz internal low-speed oscillation clock as the count clock		V_{DD} = 3.0 V		0.35	1.5	μA
LVI operating current [Note 5]	I_{LVI}	V_{DD} = 3.0 V				9	18	μA
A/D converter operating current[Note 6]	I_{ADC}	During conversion at maximum speed	High-speed mode 1	AV_{REF} = V_{DD} = 5.0 V		1.72	3.2	mA
			High-speed mode 2	AV_{REF} = V_{DD} = 3.0 V		0.72	1.6	mA
			Normal mode	AV_{REF} = V_{DD} = 5.0 V		0.86	1.9	mA
			Low-voltage mode	AV_{REF} = V_{DD} = 3.0 V		0.37	0.8	mA
Comparator operating current[Note 7]	I_{CMP}	When internal reference voltage is not used and one comparator channel is operating		AV_{REF} = V_{DD} = 5.0 V			240	μA
				AV_{REF} = V_{DD} = 3.0 V			200	μA
		When internal reference voltage is used and one comparator channel is operating		AV_{REF} = V_{DD} = 5.0 V			300	μA
				AV_{REF} = V_{DD} = 3.0 V			240	μA
Reset current	I_{DDrst}	After reset ($\overline{RESET}$ pull-up resistor current + leakage current)		AV_{REF} = V_{DD} = 5.0 V		35	100	μA

Notes 1. Total current flowing into the internal power supply (V_{DD}, AV_{REF}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors, the pull-down resistors and the output current of the port are not included.

2. Not including the current flowing into the oscillator other than the circuit which generates the clock supplied to CPU. Not including the current flowing into the LVI circuit, watchdog timer, and 8-bits timer H1 (When using the 30 kHz internal low-speed oscillation clock as the count clock).

3. Current flowing only to the watchdog timer (including the operating current of the 30 kHz internal oscillator). The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.

4. Current flowing only to the 8-bit timer H1. The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{TWH} when the 8-bit timer H1 operates (When using the 30 kHz internal low-speed oscillation clock as the count clock).

5. Current flowing only to the LVI circuit. The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates.

6. Current flowing only to the A/D converter (AV_{REF}). The current value of the 78K0/Fx2-L 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.

7. Current flowing only to the comparator (AV_{REF}). The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{CMP} when the comparator operates in an operation mode or the HALT mode.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

AC Characteristics

(1) Basic operation

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

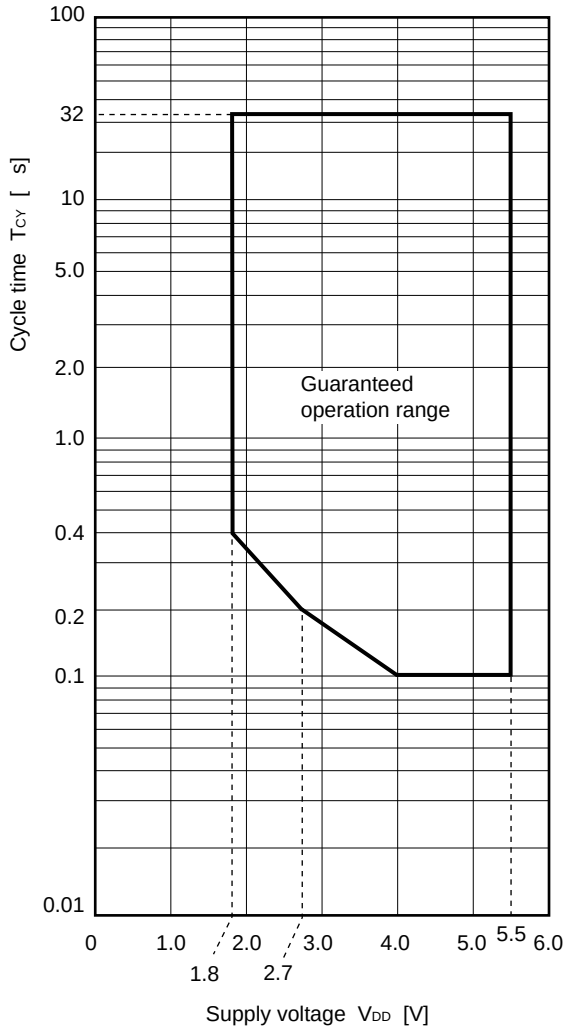
Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	In normal power mode (RMC = 00H)	4.0 V ≤ V_{DD} ≤ 5.5 V	0.1		32	μs
				2.7 V ≤ V_{DD} < 4.0 V	0.2		32	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.4 Note 1		32	μs
			In low power consumption mode (RMC = 56H)		0.4 Note 1		32	μs
Peripheral hardware clock frequency	f_{PRS}	f_{PRS} = f_{XP}		4.0 V ≤ V_{DD} ≤ 5.5 V			20	MHz
				2.7 V ≤ V_{DD} < 4.0 V			10	MHz
				1.8 V ≤ V_{DD} < 2.7 V			5	MHz
		f_{PRS} = f_{IH}		f_{IH} = 8 MHz	7.6		8.4	MHz
				f_{IH} = 4 MHz	3.88		4.12	MHz
External main system clock frequency	f_{EXCLK}	4.0 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		2.7 V ≤ V_{DD} < 4.0 V			1.0		10.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V			1.0		5.0	MHz
External main system clock input high-level width, low-level width	t_{EXCLKH}, t_{EXCLKL}				(1/f_{EXCLK} ×1/2) –1			ns
TI000, TI010 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 2			μs
		2.7 V ≤ V_{DD} < 4.0 V			2/f_{sam} + 0.2 Note 2			μs
		1.8 V ≤ V_{DD} < 2.7 V			2/f_{sam} + 0.5 Note 2			μs
TI51 input frequency	f_{TI5}	2.7 V ≤ V_{DD} ≤ 5.5 V					10.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V					5.0	MHz
TI51 input high-level width, low-level width	t_{TIH5}	4.0 V ≤ V_{DD} ≤ 5.5 V			50			ns
		2.7 V ≤ V_{DD} < 4.0 V			50			ns
		1.8 V ≤ V_{DD} < 2.7 V			100			ns
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs
Comparator input high-level width, low-level width	t_{CMPL}				125			ns

Notes 1. 0.38 μs when operating with the internal high-speed oscillation clock.

2. Selection of f_{sam} = f_{PRS}, f_{PRS}/4, f_{PRS}/256 is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00). Note that when selecting the TI000 valid edge as the count clock, f_{sam} = f_{PRS}.

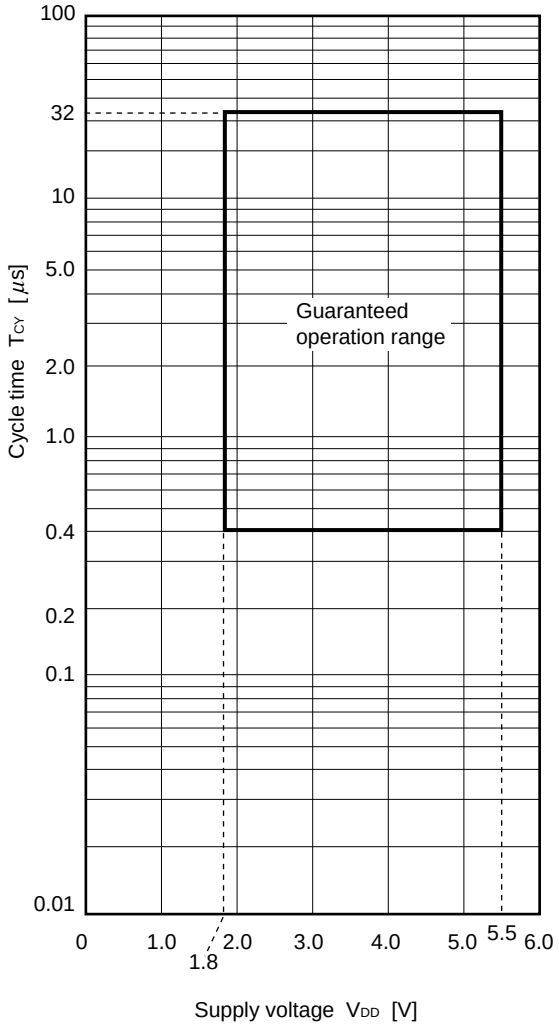
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

T_{CY} vs. V_{DD} (Main System Clock Operation, RMC = 00H (Normal Power Mode))

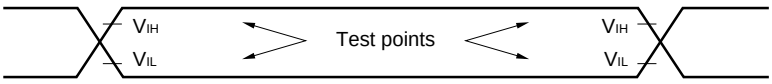


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

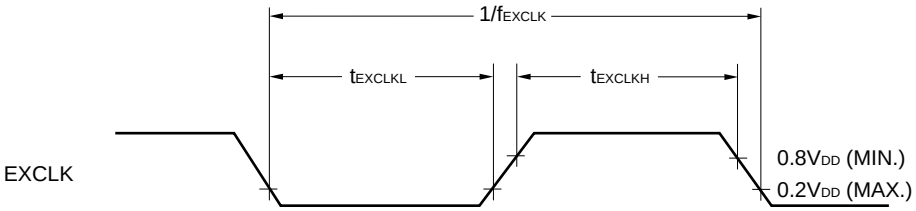
T_{CY} vs. V_{DD} (Main System Clock Operation, RMC = 56H (Low Power Mode))



AC Timing Test Points

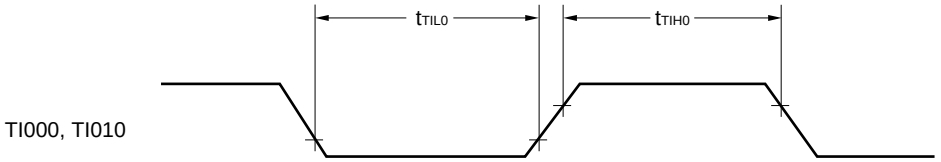


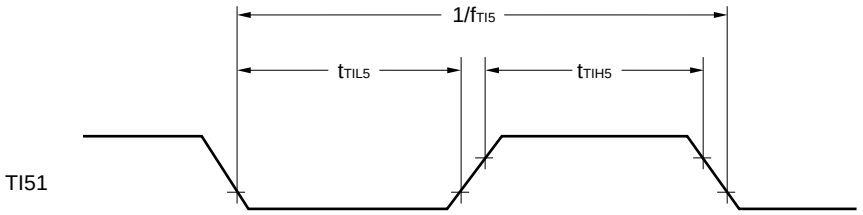
External Main System Clock Timing



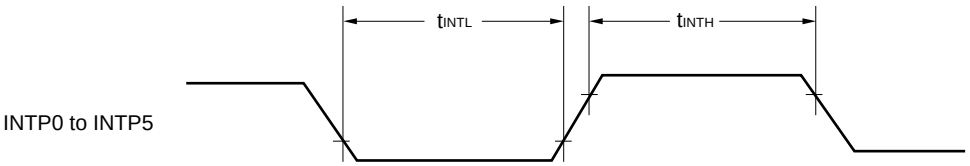
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

TI Timing

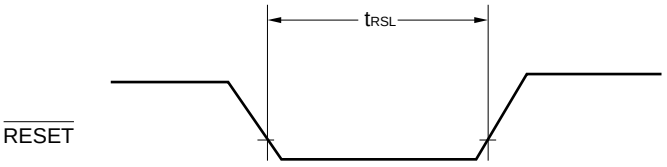




Interrupt Request Input Timing



$\overline{\text{RESET}}$ 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} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

(a) UART6 (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(b) IICA

Parameter	Symbol	Conditions	Standard Mode		High-Speed Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode : $f_{PRS} \geq 3.5$ MHz, Normal mode : $f_{PRS} \geq 1$ MHz	0	100	0	400	kHz
Setup time of start condition and stop condition	$t_{SU: STA}$		4.7	–	0.6	–	μs
Hold time[Note 1]	$t_{HD: STA}$		4.0	–	0.6	–	μs
Hold time when SCLA0 = “L”	t_{LOW}		4.7	–	1.3	–	μs
Hold time when SCLA0 = “H”	t_{HIGH}		4.0	–	0.6	–	μs
Data setup time (reception)	$t_{SU: DAT}$		250	–	100	–	ns
Data hold time (transmission)[Note 2]	$t_{HD: DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU: STO}$		4.0	–	0.6	–	μs
Bus free time between stop condition and start condition	t_{BUF}		4.7	–	1.3	–	μs
Rise time of SDAA0 and SCLA0 signals	t_R			1000	20+ $0.1C_b$	300	ns
Fall time of SDAA0 and SCLA0 signals	t_F			300	20+ $0.1C_b$	300	ns
Total load capacitance value of each communication line (SCLA0, SDAA0)	C_b			400		400	pF

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.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(c) CSI11 (master mode, $\overline{SCK11}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK11}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.0 V	400			ns
		1.8 V ≤ V_{DD} < 2.7 V	600			ns
$\overline{SCK11}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 20$ Note 1			ns
		2.7 V ≤ V_{DD} < 4.0 V	$t_{KCY1}/2 - 30$ Note 1			ns
		1.8 V ≤ V_{DD} < 2.7 V	$t_{KCY1}/2 - 60$ Note 1			ns
SI11 setup time (to $\overline{SCK11}$↑)	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} < 4.0 V	100			ns
		1.8 V ≤ V_{DD} < 2.7 V	190			ns
SI11 hold time (from $\overline{SCK11}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{SCK11}$↓ → to SO11 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 $\overline{SCK11}$ and SO11 output lines.

(d) CSI11 (slave mode, $\overline{SCK11}$... external clock input)

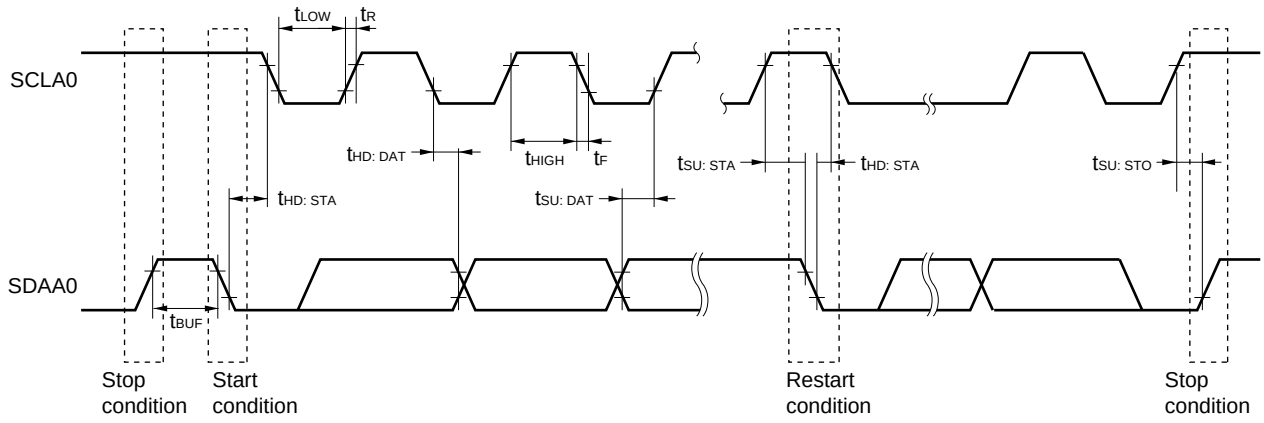
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK11}$ cycle time	t_{KCY2}		400			ns
$\overline{SCK11}$ high-/low-level width	t_{KH2}, t_{KL2}		$t_{KCY2}/2$			ns
SI11 setup time (to $\overline{SCK11}$↑)	t_{SIK2}		80			ns
SI11 hold time (from $\overline{SCK11}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{SCK11}$↓ →to SO11 output	t_{KSO2}	C = 50 pF Note			120	ns

Note C is the load capacitance of the SO11 output line.

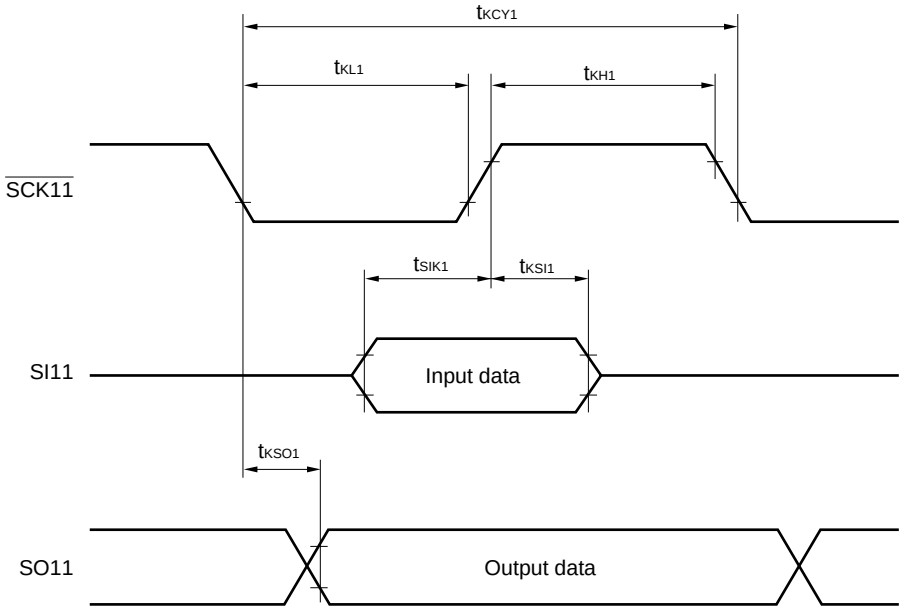
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Serial Transfer Timing

IICA:



CSI11:



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Analog Characteristics

(1) A/D Converter

(T_A = –40 to +85°C, 1.8 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Overall error[Notes 1, 2]	AINL	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
			2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		Low-voltage mode	1.8 V ≤ AV_{REF} < 2.7 V			±1.2	%FSR
Conversion time	tCONV	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V	3.3		66	μs
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V	4.4		66	μs
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V	6.6		66	μs
			2.7 V ≤ AV_{REF} < 4.0 V	13.2		66	μs
		Low-voltage mode	1.8 V ≤ AV_{REF} < 2.7 V	44		66	μs
Zero-scale error[Notes 1, 2]	EZS	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
			2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		Low-voltage mode	1.8 V ≤ AV_{REF} < 2.7 V			±0.6	%FSR
Full-scale error[Notes 1, 2]	EFS	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
			2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		Low-voltage mode	1.8 V ≤ AV_{REF} < 2.7 V			±0.6	%FSR
Integral non-linearity error[Note 1]	ILE	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±4.5	LSB
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
			2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
		Low-voltage mode	1.8 V ≤ AV_{REF} < 2.7 V			±6.5	LSB
Differential non-linearity error[Note 1]	DLE	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±2.0	LSB
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
			2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
		Low-voltage mode	1.8 V ≤ AV_{REF} < 2.7 V			±2.0	LSB
Analog input voltage	VAIN	1.8 V ≤ AV_{REF} ≤ 5.5 V		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(2) Comparator

(T_A = –40 to +85°C, 2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

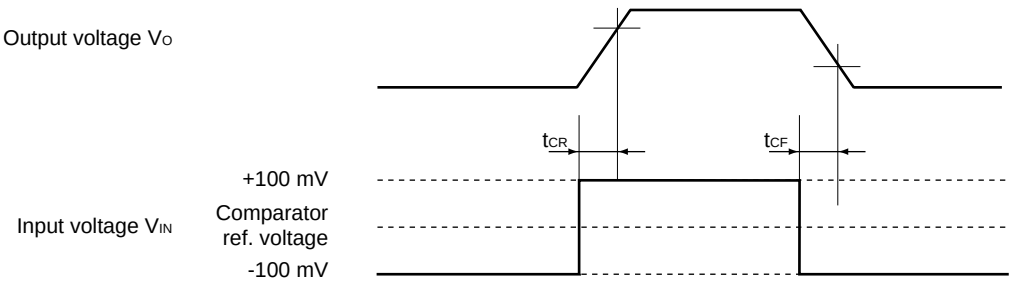
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOCMP}			±5	±40	mV
Input voltage range	V_{ICMP}	CMP0+, CMP1+, CMP2+	0		AV_{REF}	V
		CMPCOM	0.045		0.9 AV_{REF}	V
Internal reference voltage deviation	ΔV_{IREF}				60 Note 1	mV
Response time	t_{CR}, t_{CF}	Input amplitude ±100 mV		70	200	ns
Operation stabilization wait time Note 2	t_{CMP}				1	μs
Reference voltage stabilization wait time	t_{VR}	CVRE: 0 → 1 Note 3			20	μs
		CVRE = 1, When the reference voltage level is changed Note 4			5	μs

Notes 1. When the CnVRS4 to CnVRS0 bits (n = 0 to 2) are set to 1, 1, 1, 1, 1, the internal reference voltage range is 1.58 V ±60 mV.

2. Time required until a state is entered where the DC and AC specifications of the comparator are satisfied after the operation of the comparator has been enabled (CMPnEN bit = 1: n = 0 to 2)

3. Enable comparator output (CnOE bit = 1; n = 0 to 2) after enabling operation of the internal reference voltage generator (by setting the CVRE bit to 1) and waiting for the operation stabilization time to elapse.

4. Enable comparator output (CnOE bit = 1; n = 0 to 2) after enabling operation of the internal reference voltage generator (by setting the CVRE bit to 1), changing the internal reference voltage level, and waiting for the operation stabilization time to elapse.



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

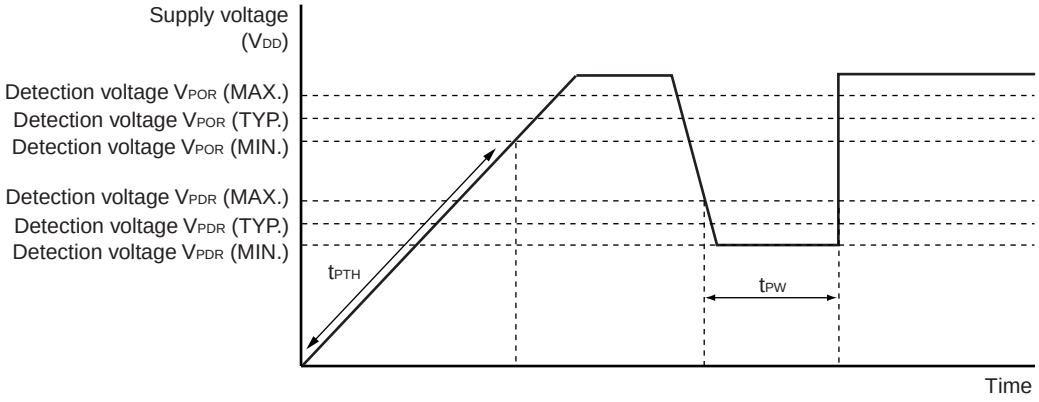
(3) POC

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}		1.52	1.61	1.70	V
	V_{PDR}		1.50	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POR}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time					200	μs

Caution When 1.8 V ≤ V_{DD} < 2.7 V, the CPU can be operated at f_{IH} = 4 MHz (TYP.).

POC Circuit Timing



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(4) Supply Voltage Rise Time

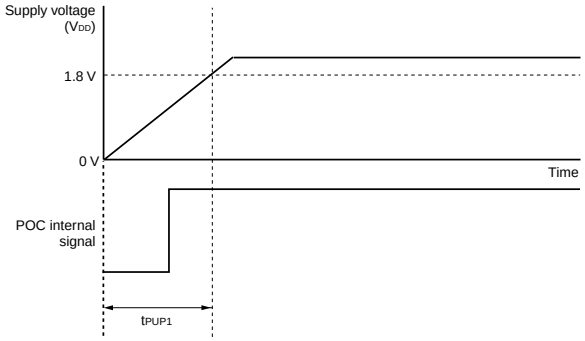
(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) Note (V_{DD}: 0 V → 1.8 V)	t_{PUP1}	LVI default start function stopped is set (LVISTART (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.)) Note (releasing $\overline{\text{RESET}}$ input → V_{DD}: 1.8 V)	t_{PUP2}	LVI default start function stopped is set (LVISTART (Option Byte) = 0), when $\overline{\text{RESET}}$ input is used			1.9	ms

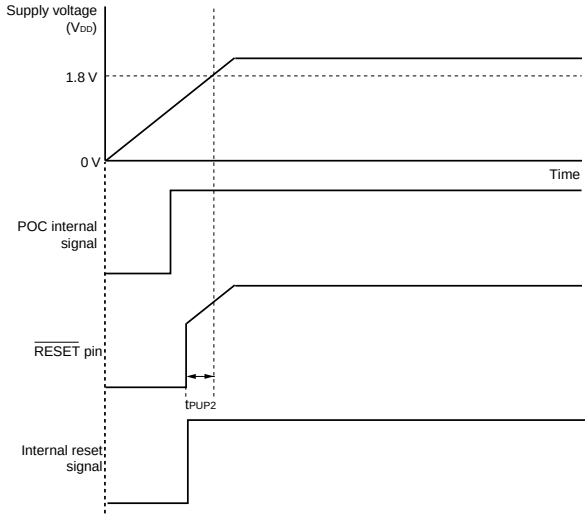
Note Make sure to raise the power supply in a shorter time than this.

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.

(5) LVI

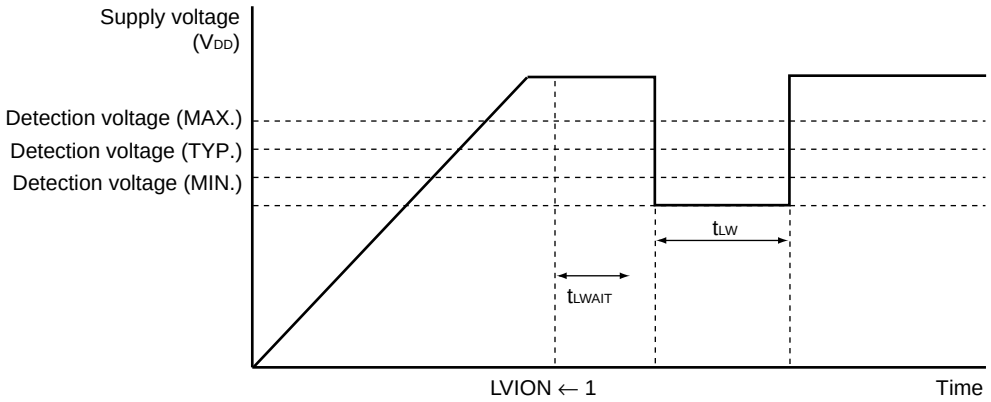
(T_A = –40 to +85°C, $V_{PDR} \le V_{DD} \le 5.5$ V, $AV_{REF} \le V_{DD}$, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.12	4.22	4.32	V
		V_{LVI1}		3.97	4.07	4.17	V
		V_{LVI2}		3.82	3.92	4.02	V
		V_{LVI3}		3.66	3.76	3.86	V
		V_{LVI4}		3.51	3.61	3.71	V
		V_{LVI5}		3.35	3.45	3.55	V
		V_{LVI6}		3.20	3.30	3.40	V
		V_{LVI7}		3.05	3.15	3.25	V
		V_{LVI8}		2.89	2.99	3.09	V
		V_{LVI9}		2.74	2.84	2.94	V
		V_{LVI10}		2.58	2.68	2.78	V
		V_{LVI11}		2.43	2.53	2.63	V
		V_{LVI12}		2.28	2.38	2.48	V
		V_{LVI13}		2.12	2.22	2.32	V
		V_{LVI14}		2.00	2.07	2.14	V
		V_{LVI15}		1.81	1.91	2.01	V
	Supply voltage when power supply voltage is turned on	V_{DDLVI}	When LVI default start function enabled is set (LVISTART = 1)	1.81	1.91	2.01	V
Minimum pulse width		t_{LW}		200			μs
Detection delay time						200	μs
Operation stabilization wait time[Note]		t_{LWAIT}				10	μs

Note Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

Remark $V_{LVI\,(n-1)} > V_{LVIn}$: n = 1 to 15

LVI Circuit Timing

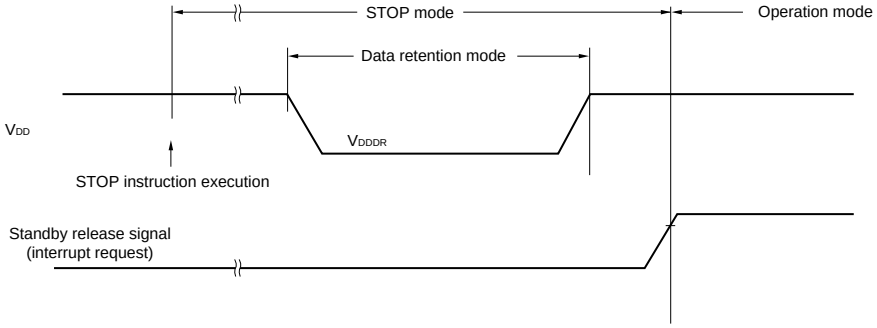


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

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}		1.50 Note		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{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	In normal power mode (RMC = 00H)	When a flash memory programmer is used, and the libraries provided by Renesas Electronics are used	Retention: 15 years	1000			Times
				When the EEPROM emulation libraries provided by Renesas Electronics are used, and the rewritable ROM size is 4 KB	Retention: 5 years	10000			Times
Operating temperature		When a flash memory programmer is used: 10 to 40 °C, during self-programming: –40 to +85 °C							
Operating voltage range		In normal power mode (RMC = 00H)		When a flash memory programmer is used		2.7 to 5.5 V@8 MHz (MAX.)			
				During self-programming		2.7 to 5.5 V@20 MHz (MAX.)			

Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)

Target products: 78K0/FY2-L: μPD78F0854(A2), 78F0855(A2), 78F0856(A2)
78K0/FA2-L: μPD78F0857(A2), 78F0858(A2), 78F0859(A2)
78K0/FB2-L: μPD78F0864(A2), 78F0865(A2)

Cautions 1. The 78K0/Fx2-L microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product as follows.

(1) Port functions

Port	78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
	16 Pins	20 Pins	30 Pins
Port 0	P00, P01		P00 to P02
Port 2	P20 to P23	P20 to P25	P20 to P27
Port 3	P30	P30 to P32	P30 to P37
Port 6	P60, P61		
Port 7	–		P70
Port 12	P121, P122		

(The remaining table is on the next page.)

(2) Non-port functions

Function		78K0/FY2-L	78K0/FA2-L	78K0/FB2-L
		16 Pins	20 Pins	30 Pins
Power supply, ground		V_{DD}, V_{SS}, AV_{REF}		V_{DD}, AV_{REF}, V_{SS}, AV_{SS}
Regulator		REGC		
Reset		$\overline{\text{RESET}}$		
Clock oscillation		X1, X2, EXCLK		
Interrupt		INTP0, INTP1	INTP0 to INTP3	INTP0 to INTP5
Timer	TMX0	–	TOX00, TOX01	TOX00, TOX01, TOX10, TOX11
	TM00	TI000, TI010, TO00		
	TM51	TI51		
	TMH1	TOH1		
Serial interface	UART6	RxD6, TxD6		
	IICA	SCLA0, SDAA0		
	CSI11	–		$\overline{\text{SCK11}}$, SI11, SO11, $\overline{\text{SSI11}}$
A/D converter		ANI0 to ANI3	ANI0 to ANI5	ANI0 to ANI8
Comparator		CMP2+	CMP0+ to CMP2+	CMP0+ to CMP2+, CMPCOM
On-chip debug function		TOOLC0, TOOLC1	TOOLC0, TOOLC1, TOOLD0, TOOLD1	

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
	V_{SS}		–0.5 to +0.3	V
	AV_{REF}		–0.5 to V_{DD} + 0.3[Note 1]	V
	AV_{SS}		–0.5 to +0.3	V
REGC pin input voltage[Note 2]	V_{IREGC}		–0.5 to +3.6 and –0.3 to V_{DD} +0.3	V
Input voltage	V_{I1}	P00 to P02, P30 to P37, P60, P61, P121, P122, X1, X2, $\overline{\text{RESET}}$	–0.3 to V_{DD} + 0.3[Note 1]	V
	V_{I2}	P20 to P27, P70	–0.3 to AV_{REF} + 0.3[Note 1] and –0.3 to V_{DD} + 0.3[Note 1]	V
Output voltage	V_{O1}	P00 to P02, P30 to P37, P60, P61	–0.3 to V_{DD} + 0.3[Note 1]	V
	V_{O2}	P20 to P27, P70	–0.3 to AV_{REF} + 0.3[Note 1]	V
Analog input voltage	V_{AN}	ANI0 to ANI8	–0.3 to AV_{REF} + 0.3[Note 1] and –0.3 to V_{DD} + 0.3[Note 1]	V

Notes 1. Must be 6.5 V or lower.

2. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

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 P02, P30 to P37, P60, P61	–10	mA
		Total of all pins –80 mA	P02, P60, P61	–30	mA
			P00, P01, P30 to P37	–55	mA
	I_{OH2}	Per pin	P20 to P27, P70	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P00 to P02, P30 to P37, P60, P61	30	mA
		Total of all pins 200 mA	P02, P60, P61	85	mA
			P00, P01, P30-P37	140	mA
	I_{OL2}	Per pin	P20 to P27, P70	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A			–40 to +125	°C
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.

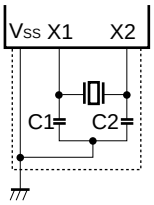
2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

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 +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic/ crystal resonator		X1 clock oscillation frequency (f_X)Note	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
			2.7 V ≤ V_{DD} < 4.0 V	2.0		10.0	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

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 High-speed Oscillator Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Internal high-speed oscillator	Oscillation frequency (f_{IH} = 4 MHz) deviation[Notes 1, 2]	RSTS = 1 (high accuracy)	T_A = –20 to +70°C			±2	%
			T_A = –40 to +85°C			±3	%
			T_A = –40 to +125°C			±5	%
	Oscillation frequency (f_{IH} = 8 MHz) deviation[Notes 1, 2]		T_A = –40 to +85°C			±3	%
			T_A = –40 to +125°C			±5	%
	Oscillation frequency (f_{IH})[Note 1]	RSTS = 0	In low power consumption mode (RMC = 56H)	1.86	4.2	7.42	MHz
			In normal power mode (RMC = 00H)	1.86	5.0	8.7	MHz

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. Internal high-speed oscillation frequency (4 MHz or 8 MHz) is set by the option byte. Refer to **CHAPTER 23 OPTION BYTE**.

PLL Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
PLL input clock frequency	f_{PLLIN}	High-speed system clock is selected (f_X = 4 MHz)		3.92	4.00	4.08	MHz
		Internal high-speed oscillation clock is selected (f_{IH} = 4 MHz)	T_A = –20 to +70°C	3.92	4.00	4.08	MHz
			T_A = –40 to +85°C	3.88	4.00	4.12	MHz
			T_A = –40 to +125°C	3.80	4.00	4.20	MHz
PLL output clock frequency	f_{PLL}			f_{PLLIN} × 10			MHz
Long-term jitter	T_{LJ}	f_{PLL} = 40 MHz, more than 400 counts			0.0029	0.10	%f_{PLL}

Internal Low-speed Oscillator Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal low-speed oscillator	Oscillation clock frequency (f_{IL})	In low power consumption mode (RMC = 56H)	25.5	30	34.5	kHz
		In normal power mode (RMC = 00H)	27	30	33	kHz

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

DC Characteristics (1/5)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P02, P30 to P37, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			–1.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			–1.0	mA
		Total of P02, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			–4.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			–3.0	mA
		Total of P00, P01, P30 to P37	4.0 V ≤ V_{DD} ≤ 5.5 V			–15.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
		Total of P00 to P02, P30 to P37, P60, P61[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			–19.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			–13.0	mA
	I_{OH2}	Per pin for P20 to P27, P70	AV_{REF} = V_{DD}			–100	μA
Output current, low[Note 2]	I_{OL1}	Per pin for P00 to P02, P30 to P37	4.0 V ≤ V_{DD} ≤ 5.5 V			4.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			2.0	mA
		Per pin for P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			8.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			2.0	mA
		Total of P02, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			6.0	mA
		Total of P00, P01, P30 to P37	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			20.0	mA
		Total of P00 to P02, P30 to P37, P60, P61[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			60.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			26.0	mA
	I_{OL2}	Per pin for P20 to P27, P70	AV_{REF} = V_{DD}			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} = –12 mA

Total output current of pins = (–12 × 0.7)/(50 × 0.01) = –16.8 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 (2/5)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P37, P122		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P20 to P27, P70	AV_{REF} = V_{DD}	0.7AV_{REF}		AV_{REF}	V
	V_{IH3}	P60, P61 (I/O port mode)		0.7V_{DD}		V_{DD}	V
	V_{IH4}	P00 to P02, P30 to P36, $\overline{\text{RESET}}$, EXCLK		0.8V_{DD}		V_{DD}	V
	V_{IH5}	P60, P61 (SMBus input mode)	2.7 V ≤ V_{DD} < 3.4 V	2.1			V
			3.4 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}			V
	V_{IH6}	P121/<TI000>/<INTP0>	When switching with alternate function is disabled (default), INTP0SEL0 = TM00SEL0 = 0	0.7V_{DD}		V_{DD}	V
			When switching with alternate function is enabled (78K0/FB2-L only), INTP0SEL0 = 1 or TM00SEL0 = 1	0.7V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P37, P122		0		0.3V_{DD}	V
	V_{IL2}	P20 to P27, P70	AV_{REF} = V_{DD}	0		0.3AV_{REF}	V
	V_{IL3}	P60, P61 (I/O port mode)		0		0.3V_{DD}	V
	V_{IL4}	P00 to P02, P30 to P36, $\overline{\text{RESET}}$, EXCLK		0		0.2V_{DD}	V
	V_{IL5}	P60, P61 (SMBus input mode)	2.7 V ≤ V_{DD} < 3.4 V	0		0.8	V
			3.4 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
	V_{IL6}	P121/<TI000>/<INTP0>	When switching with alternate function is disabled (default), INTP0SEL0 = TM00SEL0 = 0	0		0.4V_{DD}	V
			When switching with alternate function is enabled (78K0/FB2-L only), INTP0SEL0 = 1 or TM00SEL0 = 1	0		0.3V_{DD}	V
Output voltage, high	V_{OH1}	P00 to P02, P30 to P37, P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.5 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27, P70	AV_{REF} = V_{DD}, I_{OH2} = –100 μA	V_{DD} – 0.5			V

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 (3/5)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
	Output voltage, low	V_{OL1}	P00 to P02, P30 to P37	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA				0.7	V
				2.7 V ≤ V_{DD} < 4.0 V, I_{OL1} = 2.0 mA				0.7	V
		V_{OL2}	P20 to P27, P70	AV_{REF} = V_{DD}, I_{OL2} = 0.4 mA				0.4	V
		V_{OL3}	P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.0 mA				2.0	V
				4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 2.0 mA				0.6	V
				2.7 V ≤ V_{DD} < 4.0 V, I_{OL1} = 5.0 mA				0.6	V
<R>	Input leakage current, high	I_{LIH1}	P00 to P02, P30 to P37, P60, P61, $\overline{\text{RESET}}$	V_I = V_{DD}				5	μA
<R>		I_{LIH2}	P20 to P27, P70	V_I = AV_{REF} = V_{DD}				5	μA
		I_{LIH3}	P121, P122	V_I = V_{DD}	I/O port mode			5	μA
			X1, X2		OSC mode			20	μA
<R>	Input leakage current, low	I_{LIL1}	P00 to P02, P30 to P37, P60, P61, $\overline{\text{RESET}}$	V_I = V_{SS}				–5	μA
<R>		I_{LIL2}	P20 to P27, P70	V_I = V_{SS}, AV_{REF} = V_{DD}				–5	μA
		I_{LIL3}	P121, P122	V_I = V_{SS}	I/O port mode			–5	μA
			X1, X2		OSC mode			–20	μA
	Pull-up resistor	R_{PLU1}	P00 to P02, P30 to P37, P60	V_I = V_{SS}		10	20	100	kΩ
		R_{PLU2}	$\overline{\text{RESET}}$			75	150	300	kΩ

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 (4/5)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}[Note 2]	Operating mode	f_{XH} = 20 MHz[Note 4], V_{DD} = 5.0 V, RMC = 00H	Square wave input		3.2	8.3	mA
				Resonator connection		4.5	10.5	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V, RMC = 00H	Square wave input		1.6	3.9	mA
				Resonator connection		2.3	5.4	mA
			f_{XH} = 10 MHz, V_{DD} = 3.0 V, RMC = 00H	Square wave input		1.5	3.8	mA
				Resonator connection		2.2	4.5	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V, RMC = 00H	Square wave input		0.9	2.4	mA
				Resonator connection		1.3	3.0	mA
			f_{IH} = 4 MHz, V_{DD} = 3.0 V, RMC = 56H			0.5	1.5	mA
			f_{IH} = 8 MHz, V_{DD} = 5.0 V, RMC = 00H			1.3	3.2	mA
			f_{IH} = 4 MHz, f_{CPU} = 1 MHz,[Note 5] V_{DD} = 3.0 V, RMC = 56H			0.22	1.0	mA
			f_{PLL} = 40 MHz, f_{CPU} = 20 MHz, V_{DD} = 5.0 V, RMC = 00H	f_{XH} = 4 MHz	Square wave input	4.5	8.5	mA
					Resonator connection	4.8	9.7	mA
				f_{IH} = 4 MHz Note 6	Internal oscillation	4.5	8.5	mA
	I_{DD2}[Note 3]	HALT mode	f_{XH} = 20 MHz[Note 4], V_{DD} = 5.0 V, RMC = 00H		Square wave input	0.8	3.9	mA
					Resonator connection	2.0	6.6	mA
			f_{XH} = 10 MHz, V_{DD} = 5.0 V, RMC = 00H		Square wave input	0.4	1.3	mA
					Resonator connection	1.0	2.4	mA
			f_{XH} = 5 MHz, V_{DD} = 3.0 V, RMC = 00H		Square wave input	0.2	0.9	mA
					Resonator connection	0.5	1.6	mA
			f_{IH} = 4 MHz, V_{DD} = 3.0 V, RMC = 56H			0.2	0.7	mA
			f_{IH} = 8 MHz, V_{DD} = 5.0 V, RMC = 00H			0.3	1.3	mA
			f_{PLL} = 40 MHz, f_{CPU} = 20 MHz, V_{DD} = 5.0 V, RMC = 00H	f_{XH} = 4 MHz	Square wave input	1.3	3.4	mA
					Resonator connection	1.6	5.4	mA
				f_{IH} = 4 MHz Note 6	Internal oscillation	1.3	3.4	mA

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

Notes 1. Total current flowing into the internal power supply (V_{DD}, AV_{REF}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors, the pull-down resistors and the output current of the port are not included.

2. Not including the current flowing into the oscillator other than the circuit which generates the clock supplied to CPU. Not including the current flowing into the LVI circuit, A/D converter, comparator, watchdog timer, and serial interface.

3. Not including the current flowing into the oscillation circuit other than the circuit which generates the clock supplied to CPU. Not including the current flowing into the LVI circuit, A/D converter, comparator, watchdog timer, 16-bit timer X0, X1, 8-bit timer/event counter51, 8-bits timer H1 (When using the 30 kHz internal low-speed oscillation clock as the count clock), and serial interface.

4. When OSCCTL.AMPH is set to 1

5. This value is when PCC2 = 0, PCC1 = 1, and PCC0 = 0.

6. Internal high-speed oscillation clock frequency is set by using the option byte (R4M8MSEL = 0: 8 MHz, R4M8MSEL = 1: 4 MHz)

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 (5/5)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD3}[Note 3]	STOP mode	V_{DD} = 3.0 V, RMC = 56H	T_A = –40 to +50°C		0.3	2.7	μA
				T_A = –40 to +70°C		0.3	3.7	μA
				T_A = –40 to +85°C		0.3	5.5	μA
				T_A = –40 to +125°C		0.3	60	μA
Watchdog timer operating current[Note 3]	I_{WDT}	In 30 kHz internal low-speed oscillation clock operation		V_{DD} = 3.0 V		0.28	0.5	μA
TMH1 operating current[Note 4]	I_{TMH}	When using the 30 kHz internal low-speed oscillation clock as the count clock		V_{DD} = 3.0 V		0.35	2.5	μA
LVI operating current Note 5	I_{LVI}					9	27	μA
A/D converter operating current[Note 6]	I_{ADC}	During conversion at maximum speed	High-speed mode 1	AV_{REF} = V_{DD} = 5.0 V		1.72	3.2	mA
			High-speed mode 2	AV_{REF} = V_{DD} = 3.0 V		0.72	1.6	mA
			Normal mode	AV_{REF} = V_{DD} = 5.0 V		0.86	1.9	mA
			Low-voltage mode	AV_{REF} = V_{DD} = 3.0 V		0.37	0.8	mA
Comparator operating current[Note 7]	I_{CMP}	When internal reference voltage is not used and one comparator channel is operating		AV_{REF} = V_{DD} = 5.0 V			240	μA
				AV_{REF} = V_{DD} = 3.0 V			200	μA
		When internal reference voltage is used and one comparator channel is operating		AV_{REF} = V_{DD} = 5.0 V			300	μA
				AV_{REF} = V_{DD} = 3.0 V			240	μA
Reset current	I_{DDrst}	After reset ($\overline{RESET}$ pull-up resistor current + leakage current)		AV_{REF} = V_{DD} = 5.0 V		35	150	μA

Notes 1. Total current flowing into the internal power supply (V_{DD}, AV_{REF}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors, the pull-down resistors and the output current of the port are not included.

2. Not including the current flowing into the oscillator other than the circuit which generates the clock supplied to CPU. Not including the current flowing into the LVI circuit, watchdog timer, and 8-bits timer H1 (When using the 30 kHz internal low-speed oscillation clock as the count clock).

3. Current flowing only to the watchdog timer (including the operating current of the 30 kHz internal oscillator). The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.

4. Current flowing only to the 8-bit timer H1. The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{TWH} when the 8-bit timer H1 operates (When using the 30 kHz internal low-speed oscillation clock as the count clock).

5. Current flowing only to the LVI circuit. The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates.

6. Current flowing only to the A/D converter (AV_{REF}). The current value of the 78K0/Fx2-L 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.

7. Current flowing only to the comparator (AV_{REF}). The current value of the 78K0/Fx2-L microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{CMP} when the comparator operates in an operation mode or the HALT mode.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

AC Characteristics

(1) Basic operation

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

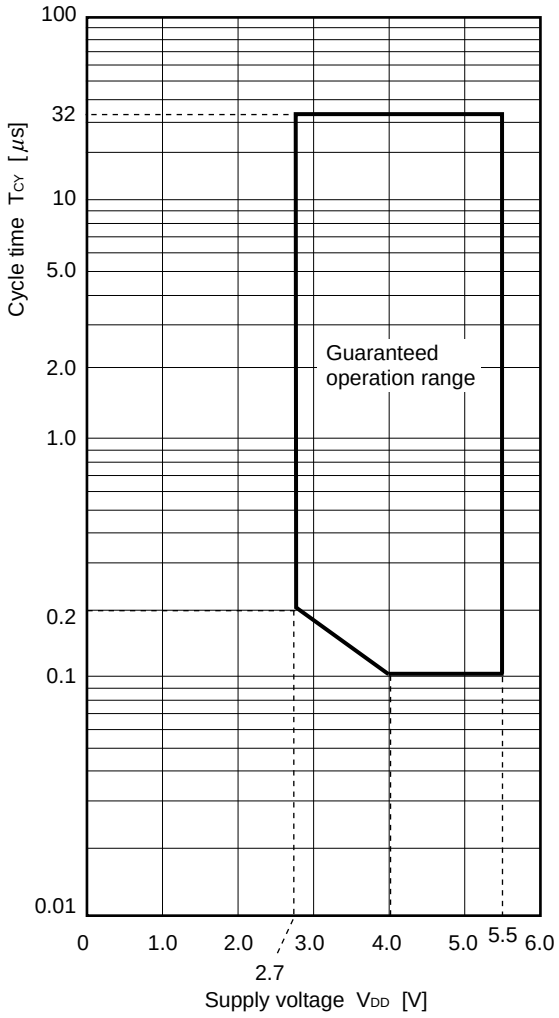
Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	In normal power mode (RMC = 00H)	4.0 V ≤ V_{DD} ≤ 5.5 V Using PLL	0.1		32	μs
				2.7 V ≤ V_{DD} < 4.0 V	0.2		32	μs
			In low power consumption mode (RMC = 56H)		0.4 Note 1		32	μs
Peripheral hardware clock frequency	f_{PRS}	f_{PRS} = f_{XP}		4.0 V ≤ V_{DD} ≤ 5.5 V			20	MHz
				2.7 V ≤ V_{DD} < 4.0 V			10	MHz
		f_{PRS} = f_{IH}		f_{IH} = 8 MHz	7.6		8.4	MHz
				f_{IH} = 4 MHz	3.88		4.12	MHz
External main system clock frequency	f_{EXCLK}	4.0 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		2.7 V ≤ V_{DD} < 4.0 V			1.0		10.0	MHz
External main system clock input high-level width, low-level width	t_{EXCLKH}, t_{EXCLKL}				(1/f_{EXCLK} ×1/2) –1			ns
TI000, TI010 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 2			μs
		2.7 V ≤ V_{DD} < 4.0 V			2/f_{sam} + 0.2 Note 2			μs
TI51 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V					10.0	MHz
		2.7 V ≤ V_{DD} < 4.0 V					5.0	MHz
TI51 input high-level width, low-level width	t_{TIH5}	4.0 V ≤ V_{DD} ≤ 5.5 V			50			ns
		2.7 V ≤ V_{DD} < 4.0 V			50			ns
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs
Comparator input high-level width, low-level width	t_{CMPL}				125			ns

Notes 1. 0.38 μs when operating with the internal high-speed oscillation clock.

2. Selection of f_{sam} = f_{PRS}, f_{PRS}/4, f_{PRS}/256 is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00). Note that when selecting the TI000 valid edge as the count clock, f_{sam} = f_{PRS}.

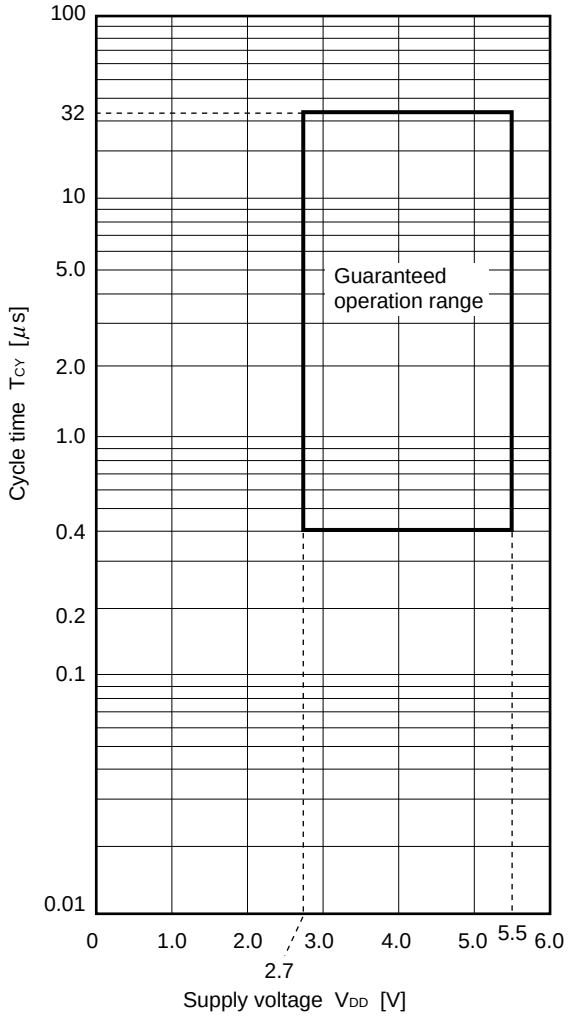
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

T_{CY} vs. V_{DD} (Main System Clock Operation, RMC = 00H (Normal Power Mode))

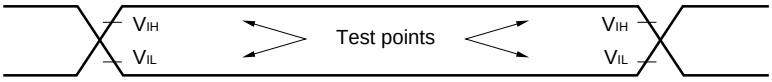


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

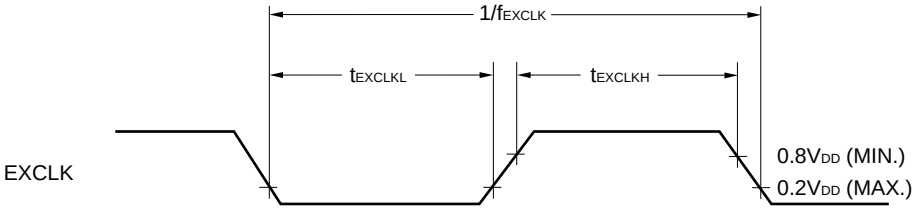
T_{CY} vs. V_{DD} (Main System Clock Operation, RMC = 56H (Low Power Mode))



AC Timing Test Points

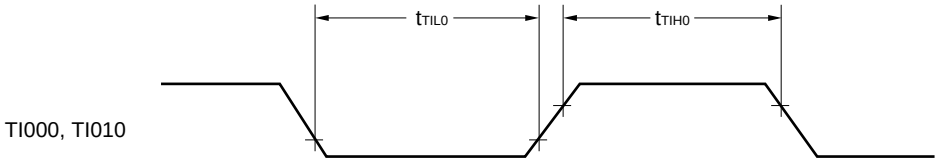


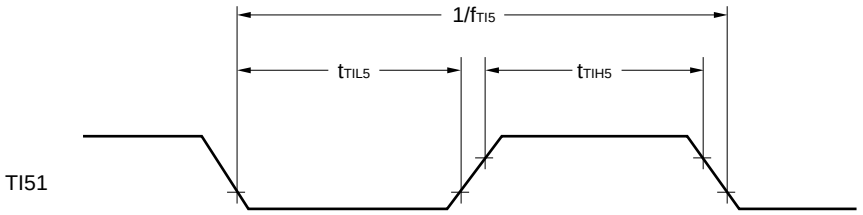
External Main System Clock Timing



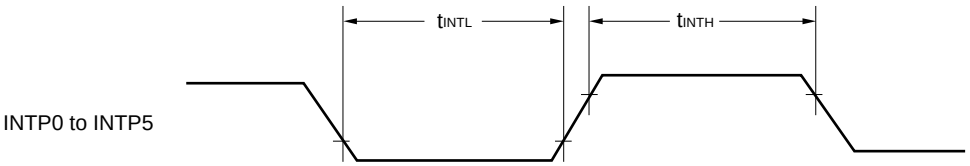
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

TI Timing

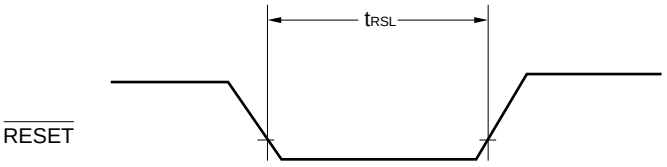




Interrupt Request Input Timing



$\overline{\text{RESET}}$ 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 +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

(a) UART6 (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(b) IICA

Parameter	Symbol	Conditions	Standard Mode		High-Speed Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode : $f_{PRS} \geq 3.5$ MHz, Normal mode : $f_{PRS} \geq 1$ MHz	0	100	0	400	kHz
Setup time of start condition and stop condition	$t_{SU:STA}$		4.7	–	0.6	–	μs
Hold time[Note 1]	$t_{HD:STA}$		4.0	–	0.6	–	μs
Hold time when SCLA0 = "L"	t_{LOW}		4.7	–	1.3	–	μs
Hold time when SCLA0 = "H"	t_{HIGH}		4.0	–	0.6	–	μs
Data setup time (reception)	$t_{SU:DAT}$		250	–	100	–	ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0	–	0.6	–	μs
Bus free time between stop condition and start condition	t_{BUF}		4.7	–	1.3	–	μs
Rise time of SDAA0 and SCLA0 signals	t_R			1000	20+ 0.1C_b	300	ns
Fall time of SDAA0 and SCLA0 signals	t_F			300	20+ 0.1C_b	300	ns
Total load capacitance value of each communication line (SCLA0, SDAA0)	C_b			400		400	pF

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.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(c) CSI11 (master mode, $\overline{\text{SCK11}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK11}}$ cycle time	t_{KCY1}	$4.0\ V \le V_{DD} \le 5.5\ V$	200			ns
		$2.7\ V \le V_{DD} < 4.0\ V$	400			ns
$\overline{\text{SCK11}}$ high-/low-level width	t_{KH1}, t_{KL1}	$4.0\ V \le V_{DD} \le 5.5\ V$	$t_{KCY1}/2 - 20$ Note 1			ns
		$2.7\ V \le V_{DD} < 4.0\ V$	$t_{KCY1}/2 - 30$ Note 1			ns
SI11 setup time (to $\overline{\text{SCK11}}\uparrow$)	t_{SIK1}	$4.0\ V \le V_{DD} \le 5.5\ V$	70			ns
		$2.7\ V \le V_{DD} < 4.0\ V$	100			ns
SI11 hold time (from $\overline{\text{SCK11}}\uparrow$)	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCK11}}\downarrow \rightarrow$ to SO11 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 $\overline{\text{SCK11}}$ and SO11 output lines.

(d) CSI11 (slave mode, $\overline{\text{SCK11}}$... external clock input)

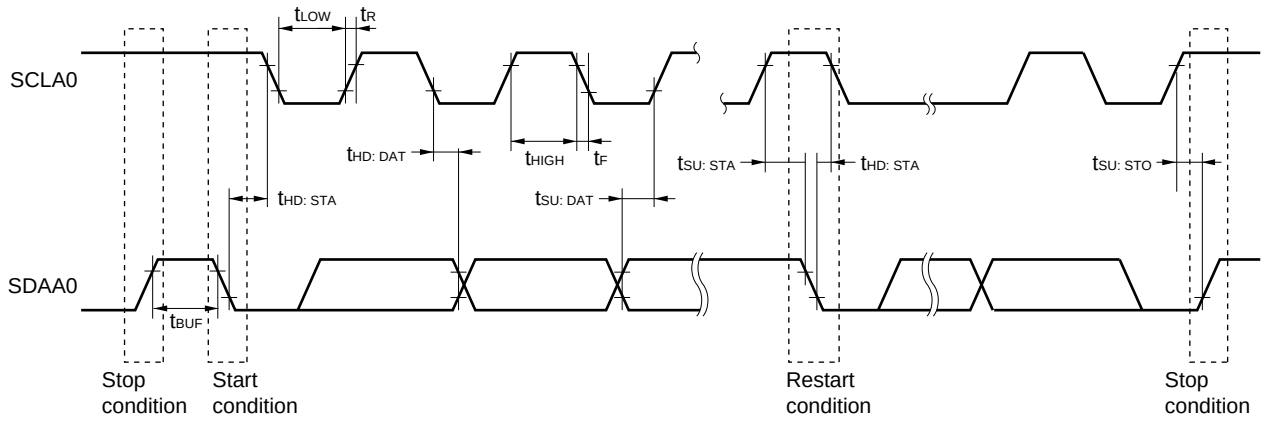
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK11}}$ cycle time	t_{KCY2}		400			ns
$\overline{\text{SCK11}}$ high-/low-level width	t_{KH2}, t_{KL2}		$t_{KCY2}/2$			ns
SI11 setup time (to $\overline{\text{SCK11}}\uparrow$)	t_{SIK2}		80			ns
SI11 hold time (from $\overline{\text{SCK11}}\uparrow$)	t_{KSI2}		50			ns
Delay time from $\overline{\text{SCK11}}\downarrow \rightarrow$to SO11 output	t_{KSO2}	C = 50 pF Note			120	ns

Note C is the load capacitance of the SO11 output line.

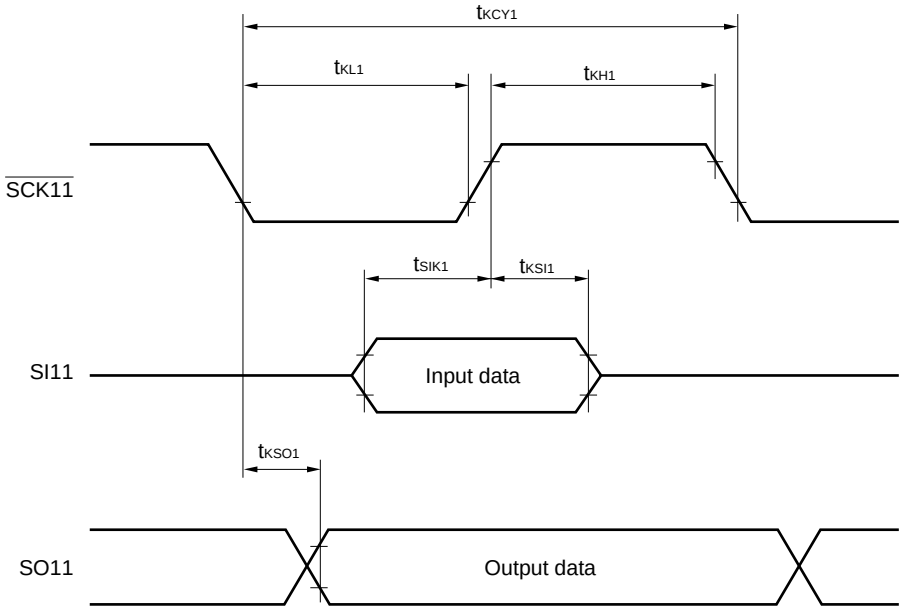
Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Serial Transfer Timing

IICA:



CSI11:



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Analog Characteristics

(1) A/D Converter

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Overall error[Notes 1, 2]	AINL	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
			2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Conversion time	tCONV	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V	3.3		66	μs
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V	4.4		66	μs
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V	6.6		66	μs
			2.7 V ≤ AV_{REF} < 4.0 V	13.2		66	μs
Zero-scale error[Notes 1, 2]	EZS	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
			2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Full-scale error[Notes 1, 2]	EFS	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±0.6	%FSR
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
			2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Integral non-linearity error[Note 1]	ILE	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±4.5	LSB
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
			2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
Differential non-linearity error[Note 1]	DLE	High-speed mode 1	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		High-speed mode 2	2.7 V ≤ AV_{REF} ≤ 5.5 V			±2.0	LSB
		Normal mode	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
			2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
Analog input voltage	VAIN			AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(2) Comparator

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

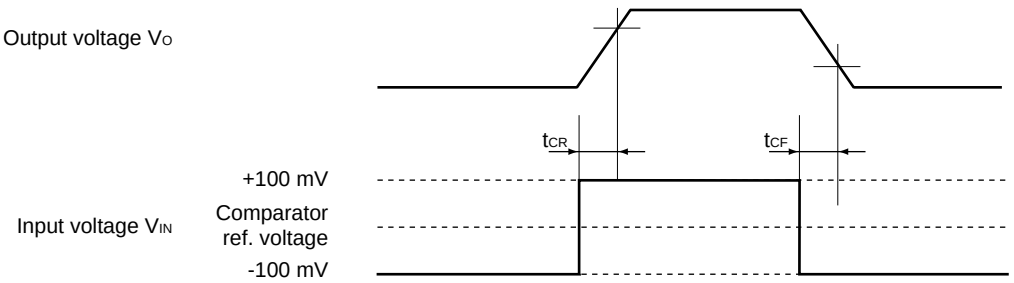
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOCMP}			±5	±40	mV
Input voltage range	V_{ICMP}	CMP0+, CMP1+, CMP2+	0		AV_{REF}	V
		CMPCOM	0.045		0.9 AV_{REF}	V
Internal reference voltage deviation	ΔV_{IREF}				60 Note 1	mV
Response time	t_{CR}, t_{CF}	Input amplitude ±100 mV		70	200	ns
Operation stabilization wait time Note 2	t_{CMP}				1	μs
Reference voltage stabilization wait time	t_{VR}	CVRE: 0 → 1 Note 3			20	μs
		CVRE = 1, When the reference voltage level is changed Note 4			5	μs

Notes 1. When the CnVRS4 to CnVRS0 bits (n = 0 to 2) are set to 1, 1, 1, 1, 1, the internal reference voltage range is 1.58 V ±60 mV.

2. Time required until a state is entered where the DC and AC specifications of the comparator are satisfied after the operation of the comparator has been enabled (CMPnEN bit = 1: n = 0 to 2)

3. Enable comparator output (CnOE bit = 1; n = 0 to 2) after enabling operation of the internal reference voltage generator (by setting the CVRE bit to 1) and waiting for the operation stabilization time to elapse.

4. Enable comparator output (CnOE bit = 1; n = 0 to 2) after enabling operation of the internal reference voltage generator (by setting the CVRE bit to 1), changing the internal reference voltage level, and waiting for the operation stabilization time to elapse.



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

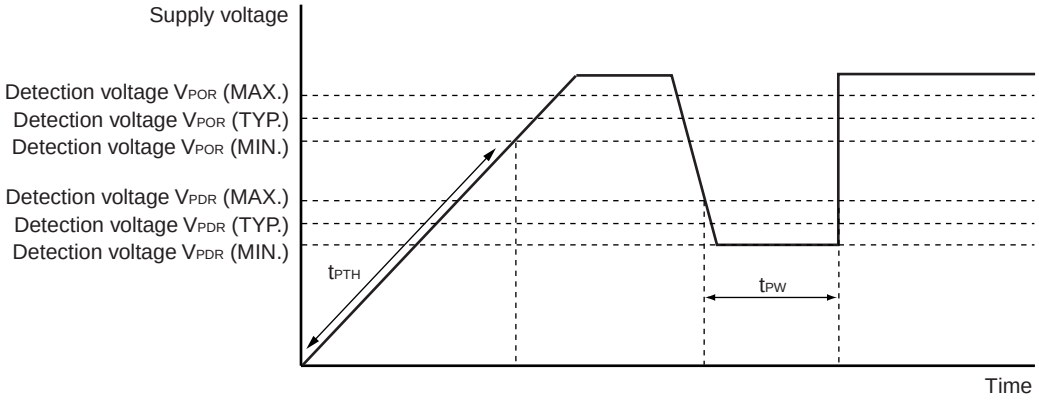
(3 POC

(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}		1.52	1.61	1.70	V
	V_{PDR}		1.50	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POR}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time					200	μs

Caution When 1.8 V ≤ V_{DD} < 2.7 V, the CPU can be operated at f_{IH} = 4 MHz (TYP.).

POC Circuit Timing



Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

(4) Supply Voltage Rise Time

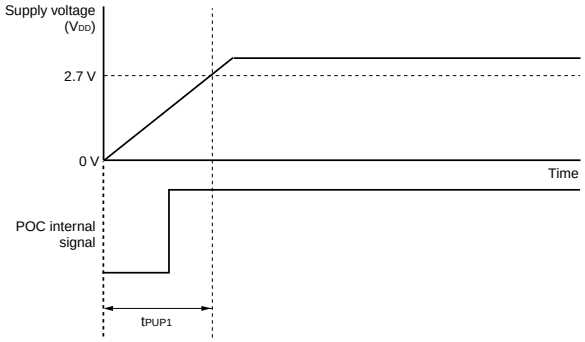
(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 2.7 V (V_{DD} (MIN.)) Note (V_{DD}: 0 V → 2.7 V)	t_{PUP1}	LVI default start function stopped is set (LVISTART (Option Byte) = 0), when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 2.7 V (V_{DD} (MIN.)) Note (releasing $\overline{\text{RESET}}$ input → V_{DD}: 2.7 V)	t_{PUP2}	LVI default start function stopped is set (LVISTART (Option Byte) = 0), when $\overline{\text{RESET}}$ input is used			1.9	ms

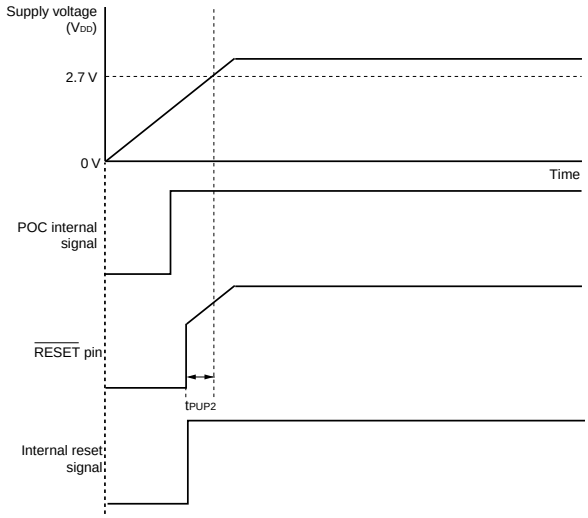
Note Make sure to raise the power supply in a shorter time than this.

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.

(5) LVI

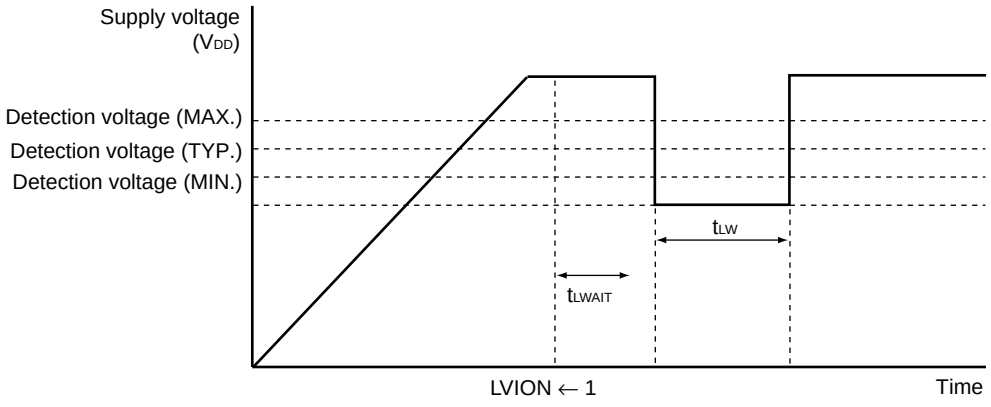
(T_A = –40 to +125°C, $V_{PDR} \le V_{DD} \le 5.5$ V, $AV_{REF} \le V_{DD}$, V_{SS} =0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.12	4.22	4.32	V
		V_{LVI1}		3.97	4.07	4.17	V
		V_{LVI2}		3.82	3.92	4.02	V
		V_{LVI3}		3.66	3.76	3.86	V
		V_{LVI4}		3.51	3.61	3.71	V
		V_{LVI5}		3.35	3.45	3.55	V
		V_{LVI6}		3.20	3.30	3.40	V
		V_{LVI7}		3.05	3.15	3.25	V
		V_{LVI8}		2.89	2.99	3.09	V
		V_{LVI9}		2.74	2.84	2.94	V
Minimum pulse width		t_{LW}		200			μs
Detection delay time						200	μs
Operation stabilization wait time[Note]		t_{LWAIT}				10	μs

Note 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 9

LVI Circuit Timing

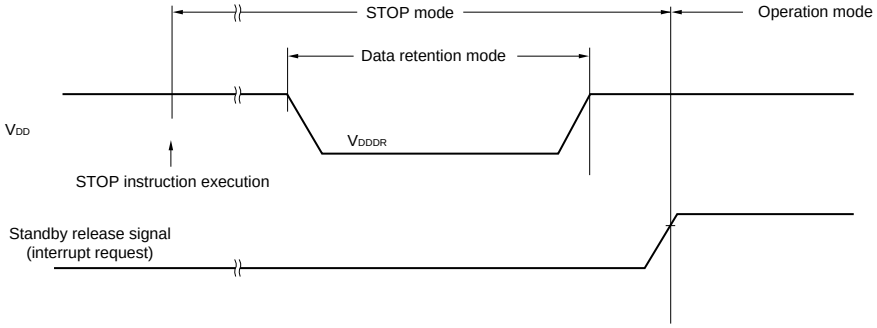


Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +125°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.50[Note]		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



Flash Memory Programming Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

• Basic characteristics

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}						4.5	16	mA
Number of rewrites per chip	C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note]	In normal power mode (RMC = 00H)	When a flash memory programmer is used, and the libraries provided by Renesas Electronics are used	Retention: 15 years	1000			Times
				When the EEPROM emulation libraries provided by Renesas Electronics are used, and the rewritable ROM size is 4 KB	Retention: 5 years	10000			Times
Operating temperature		When a flash memory programmer is used: 10 to 40 °C, during self-programming: –40 to +125 °C							
Operating voltage range		In normal power mode (RMC = 00H)		When a flash memory programmer is used		2.7 to 5.5 V@8 MHz (MAX.)			
				During self-programming		2.7 to 5.5 V@20 MHz (MAX.)			

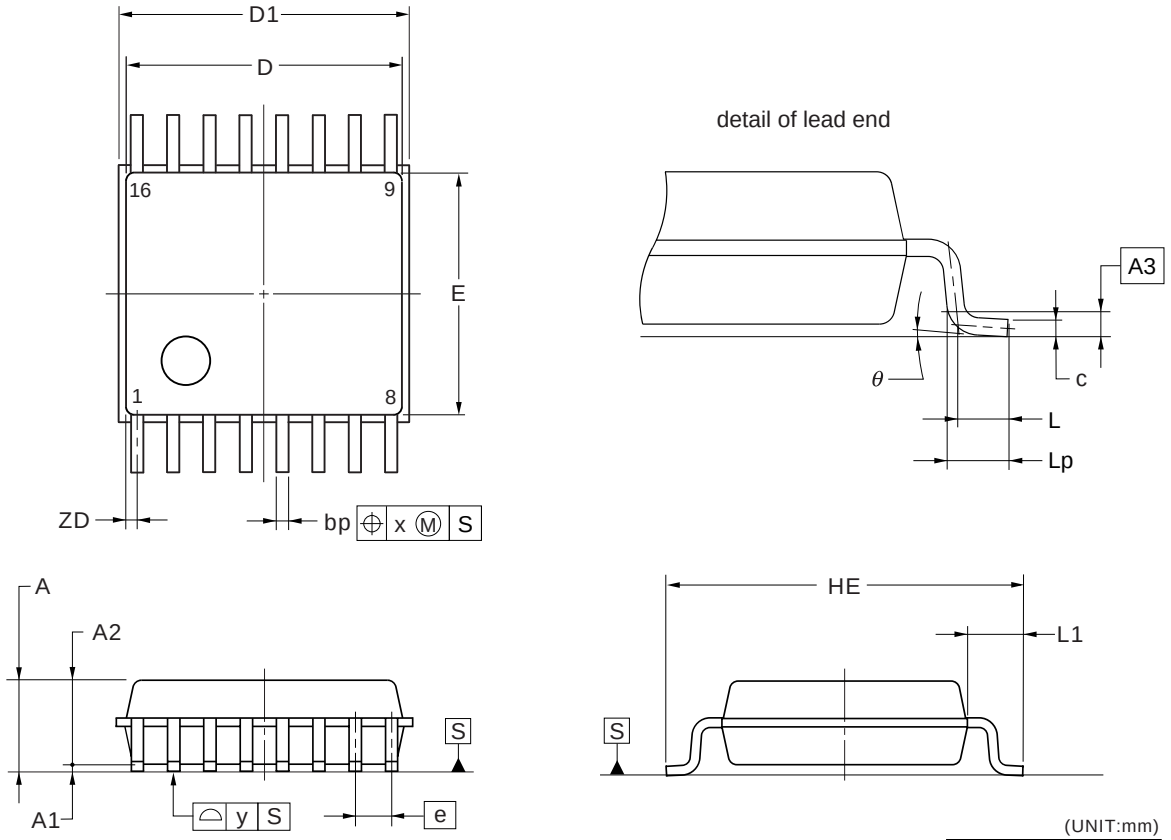
Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

CHAPTER 29 PACKAGE DRAWINGS

29.1 78K0/FY2-L

- μPD78F0854MAA-FAA-G, 78F0855MAA-FAA-G, 78F0856MAA-FAA-G, 78F0854MAA2-FAA-G, 78F0855MAA2-FAA-G, 78F0856MAA2-FAA-G

16-PIN PLASTIC SSOP (4.4x5.0)



(UNIT:mm)

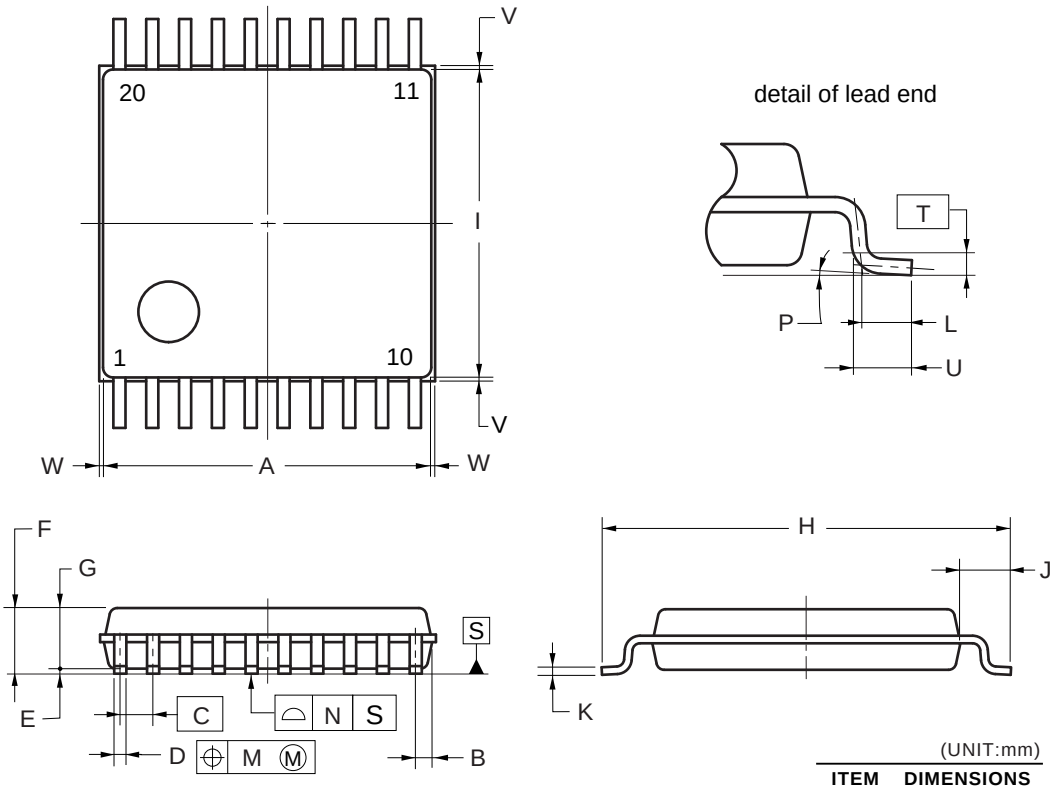
ITEM	DIMENSIONS
D	5.00±0.15
D1	5.20±0.15
E	4.40±0.20
HE	6.40±0.20
A	1.725 MAX.
A1	0.125±0.05
A2	1.50
A3	0.25
e	0.65
bp	$0.22^{+0.08}_{-0.07}$
c	$0.15^{+0.03}_{-0.04}$
L	0.50
Lp	0.60±0.10
L1	1.00±0.20
x	0.13
y	0.10
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
ZD	0.325

P16MA-65-FAA

29.2 78K0/FA2-L

- μPD78F0857MCA-CAA-G, 78F0858MCA-CAA-G, 78F0859MCA-CAA-G, 78F0857MCA2-CAA-G, 78F0858MCA2-CAA-G, 78F0859MCA2-CAA-G

20-PIN PLASTIC SSOP (7.62 mm (300))



(UNIT:mm)

ITEM	DIMENSIONS
A	6.50±0.10
B	0.325
C	0.65 (T.P.)
D	$0.22^{+0.10}_{-0.05}$
E	0.10±0.05
F	1.30±0.10
G	1.20
H	8.10±0.20
I	6.10±0.10
J	1.00±0.20
K	$0.15^{+0.05}_{-0.01}$
L	0.50
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25(T.P)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

P20MC-65-CAA

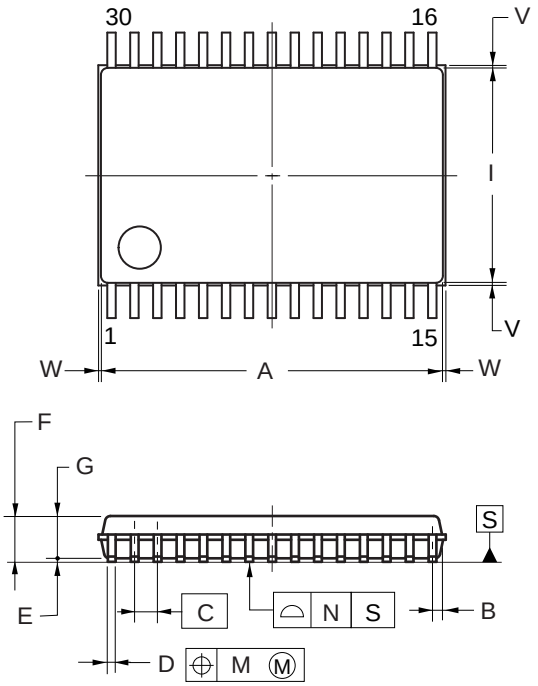
NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

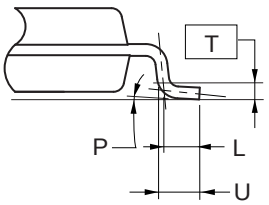
29.3 78K0/FB2-L

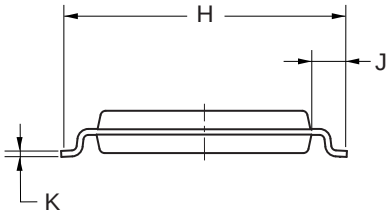
• μPD78F0864MCA-CAB-G, 78F0865MCA-CAB-G, 78F0864MCA2-CAB-G, 78F0865MCA2-CAB-G

30-PIN PLASTIC SSOP (7.62mm (300))



detail of lead end





NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

(UNIT:mm)

ITEM	DIMENSIONS
A	9.70±0.10
B	0.30
C	0.65 (T.P.)
D	$0.22^{+0.10}_{-0.05}$
E	0.10±0.05
F	1.30±0.10
G	1.20
H	8.10±0.20
I	6.10±0.10
J	1.00±0.20
K	$0.15^{+0.05}_{-0.01}$
L	0.50
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25(T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

P30MC-65-CAB

CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, please contact an Renesas Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www2.renesas.com/pkg/en/mount/index.html)

Table 30-1. Surface Mounting Type Soldering Conditions

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR60-107-3
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature), Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	WS60-107-1
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Cautions 1. Do not use different soldering methods together (except for partial heating).

2. The 78K0/Fx2-L microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

CHAPTER 31 CAUTIONS FOR WAIT

31.1 Cautions for Wait

This product has two internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with the low-speed peripheral hardware.

Because the clock of the CPU bus and the clock of the peripheral bus are asynchronous, unexpected illegal data may be passed if an access to the CPU conflicts with an access to the peripheral hardware.

When accessing the peripheral hardware that may cause a conflict, therefore, the CPU repeatedly executes processing, until the correct data is passed.

As a result, the CPU does not start the next instruction processing but waits. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks (for the number of wait clocks, refer to **Table 31-1**). This must be noted when real-time processing is performed.

31.2 Peripheral Hardware That Generates Wait

Table 31-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

Table 31-1. Registers That Generate Wait and Number of CPU Wait Clocks (1/2)

Peripheral Hardware	Register	Access	Number of Wait Clocks
16-bit timers Xn	TXnCCRn (during capture operation)	Read	3 clocks
	The above number of clocks is when the same source clock is selected for CPU and 16-bit timers Xn. The number of wait clocks can be calculated by the following expression and under the following conditions. <Calculating number of wait clocks> • Number of wait clocks = $\frac{3 f_{CPU}}{f_{TMXS}} + 1$ * Fraction is truncated if the number of wait clocks/f_{CPU} ≤ the low-level width of CPU clock and rounded up if the number of wait clocks/f_{CPU} > the low-level width of CPU clock. <Conditions for maximum/minimum number of wait clocks> • Maximum number of times: Maximum speed of CPU (20 MHz), lowest speed of TMXn clock (156.25 kHz) • Minimum number of times: Minimum speed of CPU (1.25 MHz), highest speed of TMXn clock (40 MHz: when f_{TMX} is used/20 MHz: when f_{PRS} is used) f_{TMXS}: 16-bit timers X0 and X1 selection clock frequency f_{TMX}: TMX control clock frequency f_{CPU}: CPU clock frequency f_{PRS}: Peripheral hardware clock frequency		

Caution When the peripheral hardware clock (f_{PRS}) is stopped, do not access the registers listed above using an access method in which a wait request is issued.

Remarks 1. n = 0 : 78K0/FY2-L, 78K0/FA2-L
n = 0, 1 : 78K0/FB2-L

2. The clock is the CPU clock (f_{CPU}).

Table 31-1. Registers That Generate Wait and Number of CPU Wait Clocks (2/2)

Peripheral Hardware	Register	Access	Number of Wait Clocks
Serial interface UART6	ASIS6	Read	1 clock (fixed)
Serial interface IICA	IICAS0	Read	1 clock (fixed)
A/D converter	ADM0	Write	1 to 5 clocks (when $f_{AD} = f_{PRS}/2$ is selected) 1 to 7 clocks (when $f_{AD} = f_{PRS}/3$ is selected) 1 to 9 clocks (when $f_{AD} = f_{PRS}/4$ is selected) 2 to 13 clocks (when $f_{AD} = f_{PRS}/6$ is selected) 2 to 17 clocks (when $f_{AD} = f_{PRS}/8$ is selected) 2 to 25 clocks (when $f_{AD} = f_{PRS}/12$ is selected)
	ADS	Write	
	ADPC0, ADPC1	Write	
	ADCR, ADCRH, ADCRL	Read	
	The above number of clocks is when the same source clock is selected for CPU and peripheral hardware. The number of wait clocks can be calculated by the following expression and under the following conditions. <Calculating number of wait clocks> • Number of wait clocks = $\frac{2 f_{CPU}}{f_{AD}} + 1$ * Fraction is truncated if the number of wait clocks ≤ 0.5 and rounded up if the number of wait clocks > 0.5. <Conditions for maximum/minimum number of wait clocks> • Maximum number of times: Maximum speed of CPU (f_{XP}), lowest speed of A/D conversion clock ($f_{PRS}/12$) • Minimum number of times: Minimum speed of CPU ($f_{XP}/16$), highest speed of A/D conversion clock (f_{PRS}) f_{AD}: A/D conversion clock frequency (f_{PRS} to $f_{PRS}/12$) f_{CPU}: CPU clock frequency f_{PRS}: Peripheral hardware clock frequency f_{XP}: Main system clock frequency		

Caution When the peripheral hardware clock (f_{PRS}) is stopped, do not access the registers listed above using an access method in which a wait request is issued.

Remark The clock is the CPU clock (f_{CPU}).

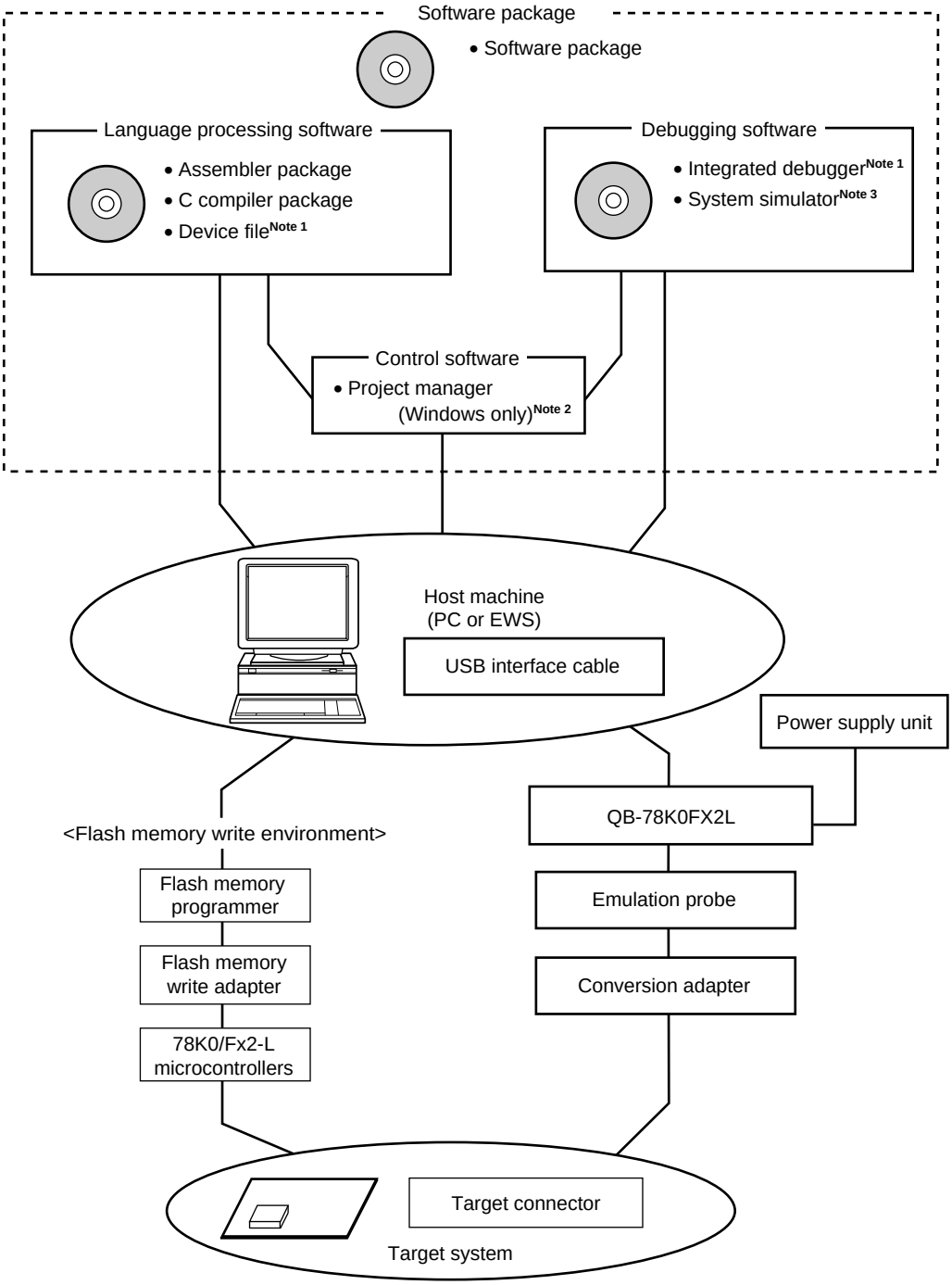
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0/Fx2-L 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-78K0FX2L



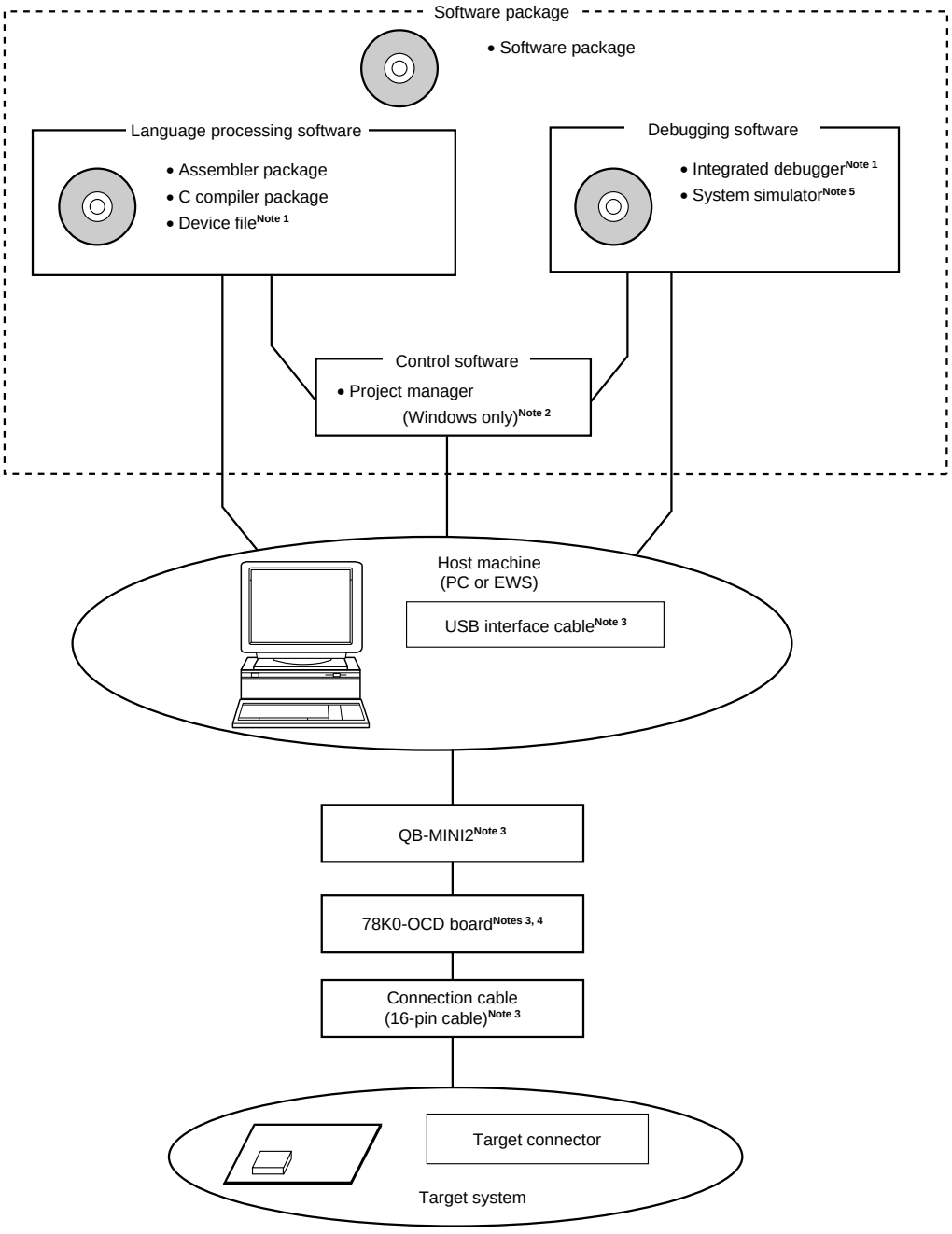
Notes 1. Download the device file for 78K0/Fx2-L microcontrollers (DF780865) and the integrated debugger ID78K0-QB from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

2. The project manager PM+ is included in the assembler package.
PM+ cannot be used other than with Windows™.

3. Instruction simulation version is 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/Fx2-L microcontrollers (DF780865) 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.
PM+ cannot be used other than with 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. Instruction simulation version is 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 (DF780865). **<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.
DF780865[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 system simulator). 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 DF780865 can be used in common with the RA78K0, CC78K0, ID78K0-QB, and system simulator. Download the DF780865 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

FL-PR5, PG-FP5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-78F0557MA-FAA-RX FA-78F0567MC-CAA-RX FA-78F0756MC-CAB-RX Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use. • FA-78F0557MA-FAA-RX : For 78K0/FY2-L • FA-78F0567MC-CAA-RX : For 78K0/FA2-L • FA-78F0756MC-CAB-RX : For 78K0/FB2-L

Remarks 1. FL-PR5, FA-78F0557MA-FAA-RX, FA-78F0567MC-CAA-RX, and FA-78F0756MC-CAB-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/Fx2-L 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-78K0Fx2L In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0/Fx2-L 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/Fx2-L microcontrollers. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. When using this as on-chip debug emulator, it should be used in combination with a connection cable (16-pin cable), 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[Note] 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 (DF780865).
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 (DF780865).

Note Download the ID78K0-QB from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

APPENDIX B REVISION HISTORY

B.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 21 LOW-VOLTAGE DETECTOR		
p.594	Modification of **When LVI default start function stopped is set (LVISTART = 0)**	(c)
p.597	Modification of **When LVI default start function stopped is set (LVISTART = 0)**	(c)
CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)		
p.655	Modification of **Pull-up resistor** in **DC Characteristics (3/5)**	(a)
CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)		
p.681	Modification of **Input leakage current, high** in **DC Characteristics (3/5)**	(a)
p.681	Modification of **Input leakage current, low** in **DC Characteristics (3/5)**	(a)

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

B.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/6)

Edition	Description	Chapter
Rev.2.00	Deletion of port function of P125/$\overline{\text{RESET}}$ pin Deletion of reset pin mode register (RSTMASK)	**Throughout**
	Modification of **Related Documents**	**INTRODUCTION**
	Change of description in **1.1 Features**	**CHAPTER 1 OUTLINE**
	Change of **Port Number]** and **[Example of Port Number]** in **1.2 Ordering Information**	
	Change of description in **1.5 Outline of Functions**	
	Change of description of $\overline{\text{RESET}}$ pin	**CHAPTER 2 PIN FUNCTIONS**
	Addition of **Caution 2** to **2.2.5 P70 (port 7)**	
	Addition of **Caution** to **2.2.6 P121 and P122 (port 12)**	
	Addition of **2.2.8 $\overline{\text{RESET}}$**	
	Change of description **in 2.2.9 REGC**	
	Change of, and deletion of **Note 2** and **Caution 2** in **Table 2-2. Pin I/O Circuit Types (78K0/FY2-L)**	
	Change of, and deletion of **Note 2** and **Caution 2** in **Table 2-3. Pin I/O Circuit Types (78K0/FA2-L)**	
	Change of , and deletion of **Note 2** and **Caution 2** in **Table 2-4. Pin I/O Circuit Types (78K0/FB2-L)**	
	Additon of Type 2 to **Figure 2-1. Pin I/O Circuit List ,** and deletion of Type 42	
	Change of **Table 3-6 Special Function Register List**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of **Table 4-5. Port Configuration**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of **Table 4-8 Setting Functions of P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins** and **Table 4-9 Setting Functions of P26/ANI6/CMPCOM Pin**	
	Change of **Figure 4-20. Block Diagram of P121 and P122**	
	Addition of **4.3 (2) Port register (Pxx)**	
	Addition of description to **4.3 (5) Port output mode register 6 (POM6)**	
	Addition of **Note 5** to **Table 4-12 Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FY2-L) (1/2)**	
	Addition of **Note 5** to **Table 4-13 Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FA2-L) (1/2)**	
	Addition of **Note 3** to **Table 4-14 Settings of Port Mode Register and Output Latch When Using Alternate Function (78K0/FB2-L) (2/2)**	
	Addition of **Note** to **Figure 5-1 Block Diagram of Clock Generator**	**CHAPTER 5 CLOCK GENERATOR**
	Addition of **Caution 1** to **Figure 5-5 Format of Main OSC Control Register (MOC)**	
	Change of **Figure 5-12 Clock Generator Operation When Power Supply Voltage Is Turned On, (When LVI Default Start Function Stopped Is Set (Option Byte: LVISTART = 0))** and **Figure 5-13 Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVISTART = 1))**	
	Change of, and addition of **Caution 2** to **Figure 5-14. CPU Clock Status Transition Diagram (When LVI Default Start Mode Function Stopped Is Set (Option Byte: LVISTART = 0))**	
	Addition of **Caution** to **Table 5-4. CPU Clock Transition and SFR Register Setting Examples (3/3)**	

(2/6)

Edition	Description	Chapter
Rev.2.00	Change of description to **6.1 Functions of 16-bit Timers X0 and X1,** addition of **(1) Interval time**r, and **(7) Timer output gating function (by interlocking with 8-bit timer H1)** to **(11) High-impedance output control function (by interlocking with comparator and INTP0)**	**CHAPTER 6 16-BIT TIMERS X0 AND X1**
	Change of **Figure 6-2 Block Diagram of 16-Bit Timer X1**	
	Addition of **Caution** to **6.2 (1) 16-bit timer Xn capture/compare register 0 (TXnCCR0)**	
	Change of, and addition of **Caution 3** to **Figure 6-8 Format of 16-Bit Timer X0 Operation Control Register 1 (TX0CTL1)**	
	Addition of **Caution 3** to **Figure 6-9 Format of 16-Bit Timer X1 Operation Control Register 1 (TX1CTL1) (78K0/FB2-L only)**	
	Addition of **Cautions 2, 3** to **Figure 6-10 Format of 16-Bit Timer X0 Operation Control Register 2 (TX0CTL2)** and **Figure 6-11 Format of 16-Bit Timer X1 Operation Control Register 2 (TX1CTL2) (78K0/FB2-L only)**	
	Change of **Figure 6-12 Format of 16-Bit Timer X0 Operation Control Register 3 (TX0CTL3)** to **Figure 6-14 Format of 16-Bit Timer X1 Operation Control Register 4 (TX1CTL4) (78K0/FB2-L only)**	
	Addition of **6.4 (1) Interval timer operation** to **(3) Capture function**	
	Change of **6.5 (1) PWM output operation (single output)** to **(4) PWM output operation (TMX0 and TMX1 synchronous start mode) (78K0/FB2-L only)**	
	Addition of **Notes 1, 2** to **Figure 6-43 Block Diagram of 16-Bit Timer X0 Output Configuration** to **Figure 6-45. Block Diagram of 16-Bit Timers X0 and X1 Output Configuration (78K0/FB2-L only)**	
	Change of **6.6 (1) Interlocking mode 1 (timer reset mode)** to **(3) Interlocking mode 3 (timer output reset mode)**	
	Change of **Figure 6-55 Format of High-impedance Output Function Control Register 0 (HZA0CTL0) (2/2)**	
	Change of **Figure 8-1 Block Diagram of 8-Bit Timer/Event Counter 51**	**CHAPTER 8 8-BIT TIMER/EVENT COUNTER 51**
	Modification of **Figure 11-1 Block Diagram of A/D Converter**	**CHAPTER 11 A/D CONVERTER**
	Addition of **11.2 (9) 10-bit A/D conversion result register for TMXn synchronization (ADCRXn)** and **(10) 8-bit A/D conversion result register L for TMXn synchronization (ADCRXnL)**	
	Addition of **10-bit A/D conversion result register for TMXn synchronization (ADCRXn)** and **8-bit A/D conversion result register L for TMXn synchronization (ADCRXnL)** to **11.3 Registers Used in A/D Converter**	
	Change of mode representation in **Figure 11-3. Timing Chart When Comparator Is Used**	
	Change of mode representation in **Table 11-2. A/D Conversion Time Selection**	
	Addition of **Note 3** to, and change of **Figure 11-10 Format of Analog Input Channel Specification Register (ADS)**	
	Change of **Table 11-4 Setting Functions of P23/ANI3/CMP2+, P24/ANI4/CMP+, P25/ANI5/CMP1+ Pins** and **Table 11-5 Setting Functions of P26/ANI6/CMPCOM Pin**	
	Addition of **Caution 3** to **11.4.1 Basic operation of A/D converter (software trigger mode)**	
	Addition of **11.4.2 Basic operation of A/D converter (timer trigger mode)**	
	Addition of **11.4.4 A/D converter trigger mode selection**	
	Addition of **Caution 5** to **11.4.5 (2) Setting of Software trigger mode** in **11.4.5 A/D converter operation mode**	
	Addition of **(3) Setting of Timer trigger mode** to **11.4.5 A/D converter operation mode**	
	Change of **Table 11-7 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	

(3/6)

Edition	Description	Chapter
Rev.2.00	Change of **12.1 Features of Comparators**	**CHAPTER 12 COMPARATORS**
	Change of **Figure 12-1 Block Diagram of Comparators**	
	Addition of **External interrupt rising edge enable registers (EGPCTL0, EGPCTL1), external interrupt falling edge enable registers (EGNCTL0, EGNCTL1)**	
	Change of **Figure 12-2 Format of Comparator 0 Control Register (C0CTL) (78K0/FA2-L, 78K0/FB2-L)** to **Figure 12-4 Format of Comparator 2 Control Register (C2CTL)**	
	Change of **Figure 12-5 Format of DA0 Internal Reference Voltage Selection Register (C0RVM)** to **Figure 12-7 Format of DA2 Internal Reference Voltage Selection Register (C2RVM)**	
	Change of **Table 12-2. Setting Functions of P23/ANI3/CMP2+, P24/ANI4/CMP0+, P25/ANI5/CMP1+ Pins** and **Table 12-3. Setting Functions of P26/ANI6/CMPCOM Pin**	
	Change of **Note** in **Figure 12-11 Example of Setting Procedure when Starting Comparator Operation (Using Internal Reference Voltage for Comparator Reference Voltage)**	
	Addition of **Note** to **Figure 12-13 Example of Setting Procedure when Starting Comparator Operation (Using Input Voltage from Comparator Common (CMPCOM) Pin for Comparator Reference Voltage (78K0/FB2-L only))**	
	Change of **Figure 12-14 Example of Setting Procedure when Stopping Comparator Operation**	
	Addition of description to **13.1 (2) Asynchronous serial interface (UART) mode**	**CHAPTER 13 SERIAL INTERFACE UART6**
	Addition of the port output mode register 6 (POM6) to **Table 13-1 Configuration of Serial Interface UART6**	
	Addition of **Note 2** to **Figure 13-4 Block Diagram of Serial Interface UART6**	
	Addition of **(9) Port output mode register 6 (POM6)** to **13.3 Registers Controlling Serial Interface UART6**	
	Addition of the port output mode register 6 (POM6) to **13.4.2 (1) Registers used**	
	Change of **Table 13-2 Relationship Between Register Settings and Pins**	
	Change of **Figure 14-1 Block Diagram of Serial Interface IICA**	**CHAPTER 14 SERIAL INTERFACE IICA**
	Addition of **Caution 3** to **Figure 14-3 Format of IICA Shift Register (IICA)**	
	Addition of **Note 3** to, and change of **Caution** in **Figure 14-5 Format of IICA Control Register 0 (IICACTL0) (1/4)**	
	Addition of description of the SPIE0 bit to **Figure 14-5 Format of IICA Control Register 0 (IICACTL0) (2/4)**	
	Change of description of the STT0 bit in **Figure 14-5 Format of IICA Control Register 0 (IICACTL0) (3/4)**	
	Change of **Caution** in **Figure 14-5 Format of IICA Control Register 0 (IICACTL0) (4/4)**	
	Change of **Figure 14-6 Format of IICA Status Register 0 (IICAS0) (2/3)**	
	Change of **Figure 14-8. Format of IICA Control Register 1 (IICACTL1) (1/2)**	
	Partial deletion of description in **14.3 (9) Port mode register 6 (PM6)**	
	Change of **14.4.2 Setting transfer clock by using IICWL and IICWH registers**	
	Addition of **Caution 2** to **18.1.1 Standby function**	**CHAPTER 18 STANDBY FUNCTION**
	Change of **Figure 18-4 HALT Mode Release by Reset**	
	Addition of **Caution 2** to **Table 18-3. Operating Statuses in STOP Mode**	
	Addition of **Note 1** to **Figure 18-5 Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)**	
	Addition of **Note 2** to **(3) When internal high-speed oscillation clock is used as CPU clock** in **Figure 18-6 STOP Mode Release by Interrupt Request Generation (2/2)**	
	Change of **Figure 18-7 STOP Mode Release by Reset**	

(4/6)

Edition	Description	Chapter
Rev.2.00	Change of **Figure 19-1 Block Diagram of Reset Function** to **Figure 19-4 Timing of Reset in STOP Mode by RESET Input**	**CHAPTER 19 RESET FUNCTION**
	Deletion of **Note** that “These are preliminary values and subject to change”.	**CHAPTER 20 POWER-ON-CLEAR CIRCUIT**
	Change of **Figure 20-2 Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector**	
	Deletion of **Note** that “These are preliminary values and subject to change”.	**CHAPTER 21 LOW-VOLTAGE DETECTOR**
	Change of **22.1 Regulator Overview**	**CHAPTER 22 REGULATOR**
	Change of, and addition of **Caution 4** to **Figure 22-1. Format of Regulator Mode Control Register (RMC)**	
	Deletion of **Table 22-1. Regulator Output Voltage Conditions**	
	Addition of **22.3 Cautions for Self Programming**	
	Change of **(4) 0083H/1083H** in **23.1 Functions of Option Bytes**	**CHAPTER 23 OPTION BYTE**
	Change of **Figure 23-1. Format of Option Byte (3/3)**	
	Change of **Table 24-2 Pin Connection**	**CHAPTER 24 FLASH MEMORY**
	Change of **24.4.1 TOOL pins**	
	Addition of **24.4.7 On-board writing when connecting crystal/ceramic resonator**	
	Change of **24.5.2 Flash memory programming mode**	
	Addition of **24.7 Processing Time for Each Command When PG-FP5 Is Used (Reference)**	
	Change of **Cautions 2** to **4** and **Remark** in **24.8 Flash Memory Programming by Self Programming**	
	Addition of **24.8.1 Register controlling self programming mode** and **24.8.2 Flow of self programming (Rewriting Flash Memory)**	
	Change of **Caution** in **24.8.3 Boot swap function**	
	Addition of **24.9 Creating ROM Code to Place Order for Previously Written Product**	
	Addition of **Caution 2** to **25.1 Connecting QB-MINI2 to 78K0/Fx2-L Microcontrollers**	**CHAPTER 25 ON-CHIP DEBUG FUNCTION**
	Modification of **Figure 25-1 Connection Example of QB-MINI2 and 78K0/Fx2-L Microcontrollers**	
	Change of **25.2 On-Chip Debug Security ID**	
	Addition of **25.3 Securing of User Resources**	
	Revision of chapter	**CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) (TARGET VALUES)**
	Addition of chapter	**CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS) (TARGET VALUES)**
	Addition of **wave soldering** to **Table 30-1 Surface Mounting Type Soldering Conditions**	**CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS (PRELIMINARY)**
	Change of **Table 31-1. Registers That Generate Wait and Number of CPU Wait Clocks (2/2)**	**CHAPTER 31 CAUTIONS FOR WAIT**
	Change of **Note 3** in **Figure A-1 Development Tool Configuration (1/2)**	**APPENDIX A DEVELOPMENT TOOLS**
	Change of **Note 5** in **Figure A-1 Development Tool Configuration (2/2)**	
	Addition of chapter	**APPENDIX B REVISION HISTORY**

(5/6)

Edition	Description	Chapter
Rev.2.01	Change URL of Renesas Electronics website	–
	Modification of, and addition of **Notes 2, 3** to **Table 2-2. Pin I/O Circuit Types (78K0/FY2-L)** to **Table 2-4. Pin I/O Circuit Types (78K0/FB2-L)**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of **Table 3-6. Special Function Register List**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of **Figure 6-3. Block Diagram of 16-Bit Timers X0 and X1**	**CHAPTER 6 16-BIT TIMERS X0 AND X1**
	Addition of **Note 3** to **Figure 6-8. Format of 16-bit Timer X0 Operation Control Register 1 (TX0CTL1)**	
	Addition of **Note 5** to **Figure 6-12. Format of 16-bit Timer X0 Operation Control Register 3 (TX0CTL3)**	
	Change of **(2) A/D conversion start timing signal output** in **6.4 Operation of 16-Bit Timers X0 and X1**	
	Change of **6.5 Operation of PWM output operation of 16-Bit Timers X0 and X1**	
	Modification of **Figure 6-47. Timing of Interlocking mode 1 (Timer reset mode)** and **Figure 6-49. Timing of Interlocking mode 2 (timer restart mode)**	
	Modification of **Figure 6-51. Timing of Interlocking mode 3 (Timer output reset mode)**	
	Addition of **(4) Priority when multiple interlocking modes occur** to **6.6 Interlocking Function with Comparator or INTP0**	
	Modification of **Figure 11-10. Format of Analog Input Channel Specification Register (ADS) (1/2)**	**CHAPTER 11 A/D CONVERTER**
	Modification of **Note** of **Figure 12-12. Example of Setting Procedure when Starting Comparator Operation (Using Internal Reference Voltage for Comparator Reference Voltage)**	**CHAPTER 12 COMPARATORS**
	Modification of **Table 18-3. Operating Statuses in STOP Mode**	**CHAPTER 18 STANDBY FUNCTION**
	Modification of **Note 2** of **Table 19-2. Hardware Statuses After Reset Acknowledgment (4/4)**	**CHAPTER 19 RESET FUNCTION**
	Modification of **Table 19-3. RESF Status When Reset Request Is Generated**	
	Modification of **Figure 20-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)**	**CHAPTER 20 POWER-ON-CLEAR CIRCUIT**
	Modification of **Note 1** of **Figure 21-2. Format of Low-Voltage Detection Register (LVIM)**	**CHAPTER 21 LOW-VOLTAGE DETECTOR**
	Modification of **Note** of **Figure 21-3. Format of Low-Voltage Detection Level Select Register (LVIS)**	
	Modification of **Remark** of **(1) Used as reset (LVIMD = 1)** in **21.4 Operation of Low-Voltage Detector**	
	Modification of **(2) When LVI default start function enabled is set (LVISTART = 1)** in **21.4.1 When used as reset**	
	Modification of **(2) When LVI default start function enabled is set (LVISTART = 1)** in **21.4.2 When used as interrupt**	

(6/6)

Edition	Description	Chapter
Rev.2.02	Modification of **Figure 6-22. (c) 16-bit timer X0 operation control register 1 (TX0CTL1)**	**CHAPTER 6 16-BIT TIMERS X0 AND X1**
	Modification of **Figure 6-22. (d) 16-bit timer X1 operation control register 1 (TX1CTL1) (78K0/FB2-L only)**	
	Modification of **6.4 (3) Capture function**	
	Modification of **Remarks 1.** and **Remarks 2.** in **6.4 (3) Capture function**	
	Change of **Figure 6-24. Basic Timing Example of Capture Function Operation**	
	Modification of **Remarks 1.** and **Remarks 2.** in **Figure 6-24. Basic Timing Example of Capture Function Operation**	
	Modification of **Figure 6-25. (b) 16-bit timer X0 operation control register 1 (TX0CTL1)**	
	Modification of **Figure 6-25. (c) 16-bit timer X1 operation control register 1 (TX1CTL1) (78K0/FB2-L only)**	
	Change of **Figure 6-26. Example of Software Processing for Capture Function Operation**	
	Modification of **<1> Count operation start flow** in **Figure 6-26. Example of Software Processing for Capture Function Operation**	
	Modification of **Remarks 1.** and **Remarks 2.** in **Figure 6-26. Example of Software Processing for Capture Function Operation**	

78K0/Fx2-L User’s Manual: Hardware

Publication Date: Rev.0.01 Sep 9, 2009
Rev.2.03 Jun 29, 2012

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/Fx2-L

Renesas Electronics Corporation

R01UH0068EJ0203

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X SC705C8AE0VFBE STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR LC87F0G08AUJA-AH CP8361BT STM8S207C6T3 CG8421AF