To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

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

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

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

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

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

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

RENESAS

User's Manual

78K0/FE2

8-Bit Single-Chip Microcontrollers

μPD78F0887(A) **μPD78F0887(A2)**
μPD78F0888(A) **μPD78F0888(A2)**
μPD78F0889(A) **μPD78F0889(A2)**
μPD78F0890(A) **μPD78F0890(A2)**

The 78K0/FE2 has an on-chip debug function.
Do not use this product for mass production after the on-chip debug function has been used because its reliability cannot be guaranteed, due to issues with respect to the number of times the flash memory can be rewritten. NEC Electronics does not accept complaints concerning when use this product for mass production after the on-chip debug function has been used.

Document No. U17554EJ4V0UD00 (4th edition)
Date Published March 2007 NS CP(K)

[MEMO]

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 V_{IL} (MAX) and V_{IH} (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 V_{IL} (MAX) and V_{IH} (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 V_{DD} 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.

EEPROM is trademark of NEC Electronics Corporation.
Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT is a trademark of International Business Machines Corporation.
HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Company.
SPARCstation is a trademark of SPARC International, Inc.
Solaris and SunOS are trademarks of Sun Microsystems, Inc.
SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

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

- **The information in this document is current as of March, 2007. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "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.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

[MEMO]

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0/FE2 and design and develop application systems and programs for these devices. The target products are as follows.

78K0/FE2: μPD78F0887 (A), 78F0888 (A), 78F0889 (A), 78F0890 (A)
78F0887 (A2), 78F0888 (A2), 78F0889 (A2), 78F0890 (A2)

Purpose

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

Organization

The 78K0/FE2 manual are separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

78K0/FE2 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.

- When using this manual as the manual for (A) and (A2) grade products:
  → Only the quality grade differs between (A) grade products and (A2) grade products.
  Read the part number as follows.
  - μPD78F0887→ μPD78F0887 (A), 78F0887 (A2)
  - μPD78F0888→ μPD78F0888 (A), 78F0888 (A2)
  - μPD78F0889→ μPD78F0889 (A), 78F0889 (A2)
  - μPD78F0890→ μPD78F0890 (A), 78F0890 (A2)
- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark <R> shows major revised points.
- How to interpret the register format:
  → For a bit number enclosed in brackets, the bit name is defined as a reserved word in the assembler, and is already defined in the header file named sfrbit.h in the C compiler.
- To check the details of a register when you know the register name:
  → Refer to **APPENDIX C REGISTER INDEX**.

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/FE2 User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E

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

Document Name		Document No.
RA78K0 Ver.3.80 Assembler Package	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
CC78K0 Ver.3.70 C Compiler	Operation	U17201E
	Language	U17200E
ID78K0-QB Ver. 2.90 Integrated Debugger	Operation	U17437E
PM plus Ver. 5.20		U16934E

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

Document Name	Document No.
QB-78K0FX2 In-Circuit Emulator	U17534E
QB-78K0MINI ON-CHIP DEBUG Emulator	U17029E
<R> QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP4 Flash Memory Programmer User's Manual	U15260E
<R> PG-FPL3 Flash Memory Programmer User's Manual	U17454E

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://www.necel.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.

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

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

Item / Part Number	Program Memory (ROM)		Data Memory	
			Internal High-Speed RAM[Note]	Internal Expansion RAM[Note]
μPD78F0887	Flash memory[Note]	48 KB	1024 bytes	2048 bytes
μPD78F0888		60 KB		
μPD78F0889		96KB		4096 bytes
μPD78F0890		128 KB		6144 bytes

Note The internal flash memory, internal high-speed RAM capacities, and internal expansion RAM capacities can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

- On-chip single-power-supply flash memory
- Self-programming (with boot swap function)
- On-chip debug function
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- Short startup is possible via the CPU default start using the on-chip internal high-speed oscillator
- On-chip watchdog timer (operable with on-chip internal low-speed oscillator clock)
- On-chip multiplier/divider
- On-chip clock output/buzzer output controller
- I/O ports: 55 (N-ch open drain: 4)
- Timer: 10 channels
- Serial interface: 4 channels
  (UART (LIN (Local Interconnect Network)-bus supported): 1 channel,
  CSI/UART [Note]: 1 channel, CSI: 1 channel, CAN: 1 channel)
- 10-bit resolution A/D converter: 12 channels
- Supply voltage: V_{DD} = 4.0 to 5.5 V when 20 MHz, V_{DD} = 2.7 to 5.5 V when 10 MHz, V_{DD} = 1.8 to 5.5 V when 5 MHz
  (with internal high-speed oscillator clock or subsystem clock: V_{DD} = 1.8 to 5.5 V)
- Operating ambient temperature: T_A = −40 to +85°C, −40 to +125°C

Note Select either of the functions of these alternate-function pins.

1.2 Applications

- Automotive electrical appliances (Body control, Door control, Front light control)
- Industrial equipment (Industrial robot, Building control)

1.3 Ordering Information

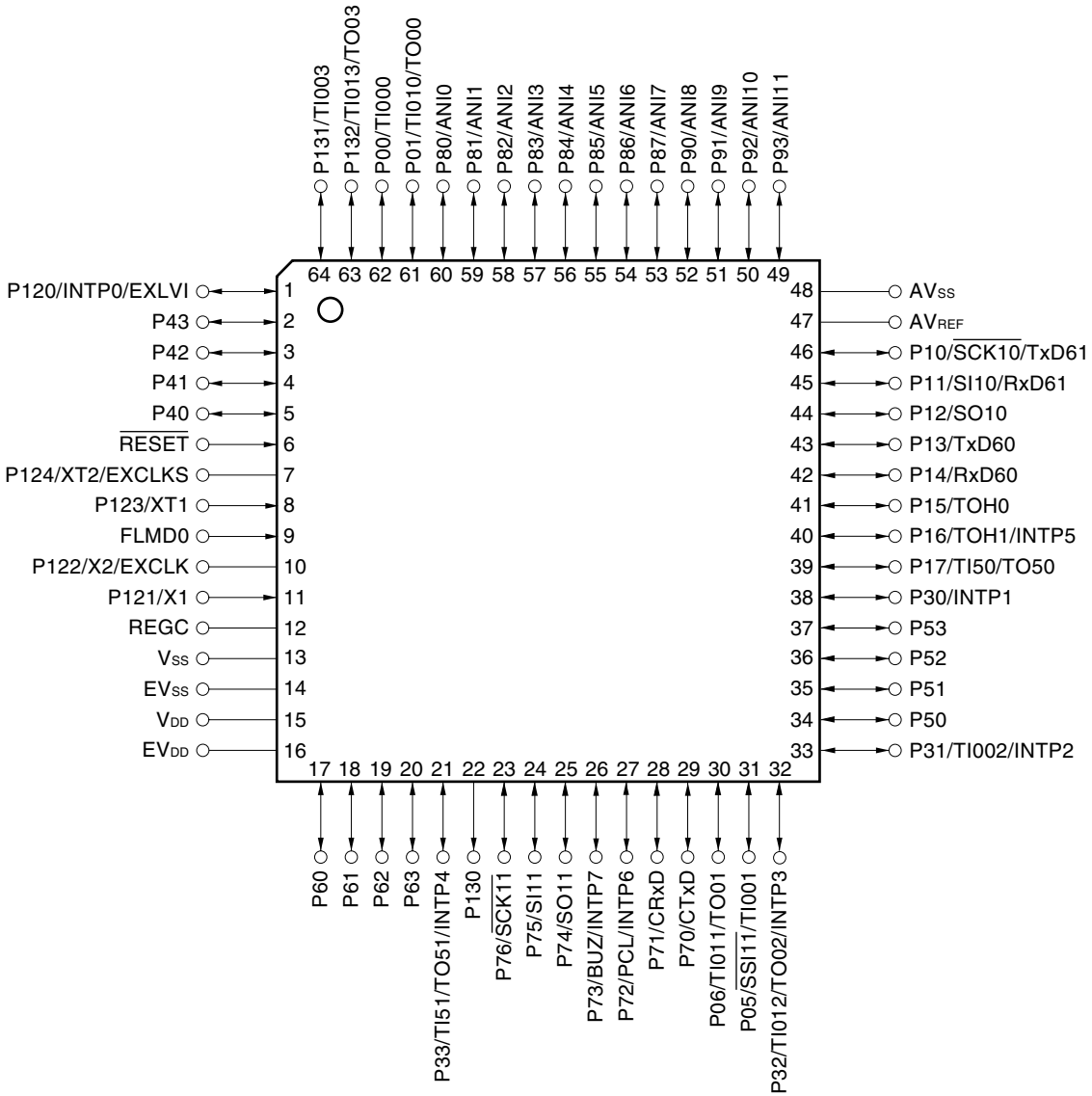
- **Flash memory version**

Part Number	Package	Quality Grade
μPD78F0887GK(A)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0887GK(A2)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0887GB(A)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special
μPD78F0887GB(A2)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special
μPD78F0888GK(A)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0888GK(A2)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0888GB(A)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special
μPD78F0888GB(A2)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special
μPD78F0889GK(A)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0889GK(A2)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0889GB(A)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special
μPD78F0889GB(A2)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special
μPD78F0890GK(A)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0890GK(A2)-GAJ-AX	64-pin plastic LQFP (12x12)	Special
μPD78F0890GB(A)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special
μPD78F0890GB(A2)-GAH-AX	64-pin plastic LQFP (Fine pitch) (10x10)	Special

Remark All these products are lead free products.

1.4 Pin Configuration (Top View)

- 64-pin plastic LQFP (12x12)
- 64-pin plastic LQFP (Fine pitch) (10x10)



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

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

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

4. ANI0/P80 to ANI11/P93 are set in the analog input mode after release of reset.

Pin Identification

ANI0 to ANI11:	Analog input	P90 to P93:	Port 9
AV_{REF}:	Analog reference voltage	P120 to P124:	Port 12
AV_{SS}:	Analog ground	P130 to P132:	Port 13
BUZ:	Buzzer output	PCL:	Programmable clock output
CRxD:	Receive data for CAN	REGC:	Regulator Capacitance
CTxD:	Transmit data for CAN	$\overline{\text{RESET}}$:	Reset
EV_{DD}:	Power supply for port	RxD60, RxD61:	Receive data
EV_{SS}:	Ground for port	$\overline{\text{SCK10}}$, $\overline{\text{SCK11}}$:	Serial clock input/output
EXCLK:	External clock input	SI10, SI11:	Serial data input
	(Main system clock)	SO10, SO11:	Serial data output
EXCLKS:	External clock input	$\overline{\text{SSI11}}$:	Serial interface chip select input
	(Subsystem clock)	TI000, TI010,	
EXLVI:	External potential input	TI001, TI011,	
	for low-voltage detector	TI002, TI012,	
FLMD0:	Flash programming mode	TI003, TI013,	
INTP0 to INTP7:	External interrupt input	TI50, TI51:	Timer input
P00, P01,		TO00, TO01,	
P05, P06:	Port 0	TO02, TO03	
P10 to P17:	Port 1	TO50, TO51,	
P30 to P33:	Port 3	TOH0, TOH1:	Timer output
P40 to P43:	Port 4	TxD60, TxD61:	Transmit data
P50 to P53:	Port 5	V_{DD}:	Power supply
P60 to P63:	Port 6	V_{SS}:	Ground
P70 to P76:	Port 7	X1, X2:	Crystal oscillator (high-speed system clock)
P80 to P87:	Port 8	XT1, XT2:	Crystal oscillator (subsystem clock)

1.5 Fx2 Series Lineup

1.5.1 78K0/Fx2 product lineup

- 44-pin LQFP (10 × 10 mm 0.8 mm pitch)

78K0/FC2

μPD78F0881	μPD78F0882	μPD78F0883
Single-power-supply flash memory: 32 KB, RAM: 2 KB	Single-power-supply flash memory: 48 KB, RAM: 3 KB	Single-power-supply flash memory: 60KB, RAM: 3 KB

- 48-pin LQFP (7 × 7 mm 0.5 mm pitch)

78K0/FC2

μPD78F0884	μPD78F0885	μPD78F0886
Single-power-supply flash memory: 32 KB, RAM: 2 KB	Single-power-supply flash memory: 48 KB, RAM: 3 KB	Single-power-supply flash memory: 60KB, RAM: 3 KB

- 64-pin LQFP (10 × 10 mm 0.5 mm pitch, 12 × 12 mm 0.65 mm pitch)

78K0/FE2

μPD78F0887	μPD78F0888	μPD78F0889	μPD78F0890
Single-power-supply flash memory: 48 KB, RAM: 3 KB	Single-power-supply flash memory: 60 KB, RAM: 3 KB	Single-power-supply flash memory: 96 KB, RAM: 5 KB	Single-power-supply flash memory: 128 KB, RAM: 7 KB

- 80-pin LQFP (12 × 12 mm 0.5 mm pitch, 14 × 14 mm 0.65 mm pitch)

78K0/FF2

μPD78F0891	μPD78F0892	μPD78F0893
Single-power-supply flash memory: 60 KB, RAM: 3 KB	Single-power-supply flash memory: 96 KB, RAM: 5 KB	Single-power-supply flash memory: 128 KB, RAM: 7 KB

Remark All product with on-chip debug function.

The list of functions in the 78K0/Fx2 is shown below.

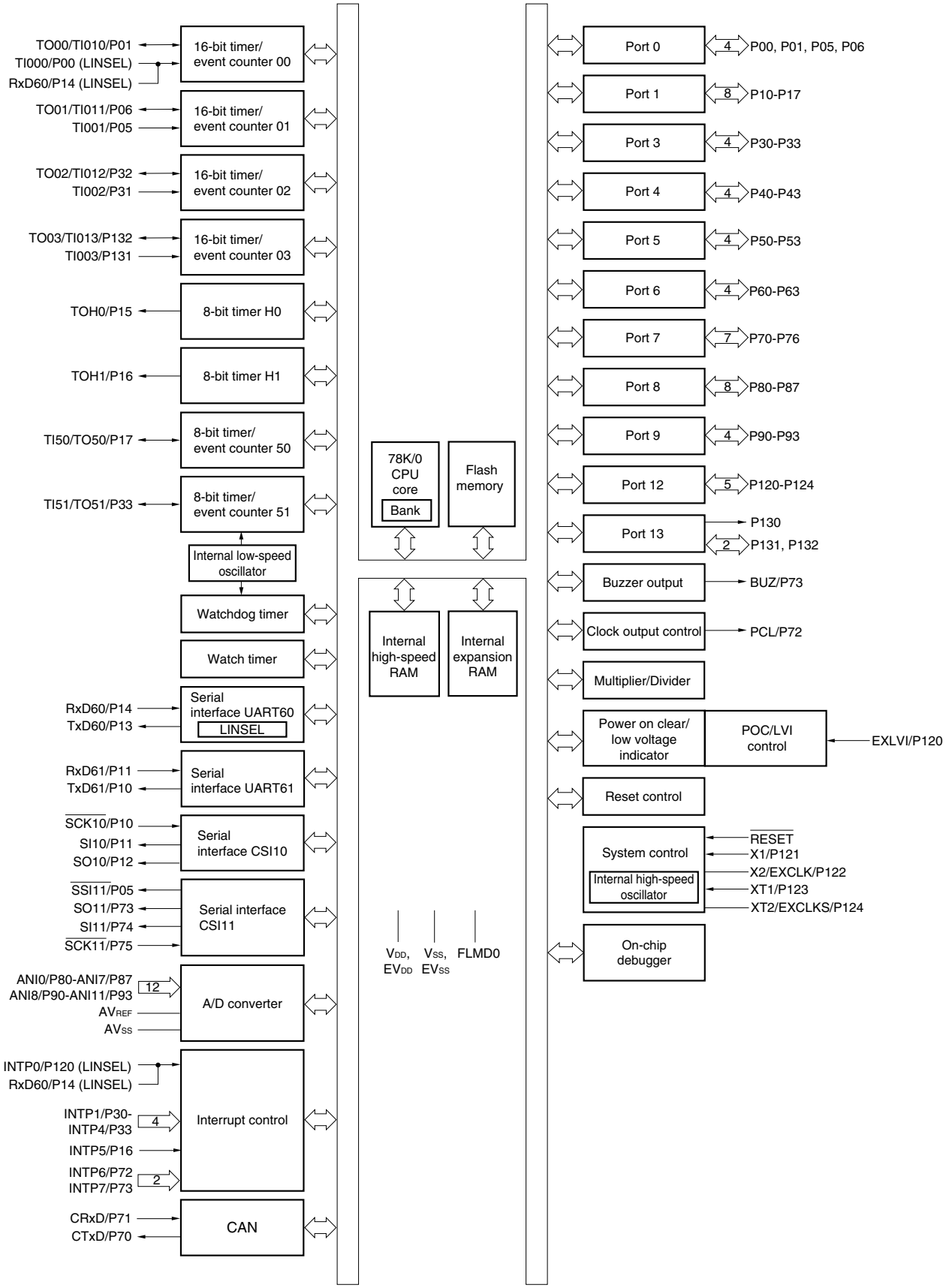
Item \ Part Number		78K0/FC2		78K0/FE2	78K0/FF2
Number of pins		44 pins	48 pins	64 pins	80 pins
Internal memory (bytes)	Flash memory	32 K/48 K/60 K		48 K/60 K/96 K/128 K	60 K/96 K/128 K
	RAM	2 K/3 K/3 K		3 K/3 K/5 K/7 K	3 K/5 K/7 K
Power supply voltage		V_{DD} = 4.0 to 5.5 V when 20 MHz, V_{DD} = 2.7 to 5.5 V when 10 MHz, V_{DD} = 1.8 to 5.5 V when 5 MHz			
Minimum instruction execution time		0.1 μs (when 20 MHz, V_{DD} = 4.0 to 5.5 V)			
Clock	Crystal/ceramic	4 to 20 MHz			
	Subclock	32.768 kHz			
	Internal low-speed oscillator	240 kHz (TYP.)			
	Internal high-speed oscillator	8 MHz (TYP., V_{DD} = 2.7 to 5.5 V)			
Ports	CMOS I/O	33	36	50	66
	CMOS output	1			
	N-ch open-drain I/O	3	4		
Timer	16 bits (TM0)	2 ch[Note]		4 ch	
	8 bits (TM5)	2 ch			
	8 bits (TMH)	2 ch			
	For watch	1 ch			
	WDT	1 ch			
Serial interface	CAN	1 ch			
	3-wire CSI	–		1 ch	
	LIN-UART	1 ch			
	LIN-UART/CSI	1 ch			
10-bit A/D converter		8 ch	9 ch	12 ch	16 ch
Interrupts	External	8			
<R>	Internal	24		29	
Reset	$\overline{\text{RESET}}$ pin	Provided			
	POC	1.59 V ±0.15 V (detection voltage is fixed)			
	LVI	4.24/4.09/3.93/3.78/3.62/3.47/3.32/3.16/3.01/2.85/2.70/2.55/2.39/2.24/2.08/1.93 V (selectable by software)			
	WDT	Provided			
Multiplier/divider		Provided			
Clock output/buzzer output		Provided			
Self-programming function		Provided			
On-chip debug function		Provided			
Standby function		HALT/STOP mode			
Operating ambient temperature		T_A = –40 to +85°C, –40 to +125°C			

Note Since TM01 does not have the following terminal at 78K0/FC2, the function is restricted in part.

μPD78F0881, 78F0882, and 78F0883: TI001, TI011, TO01

μPD78F0884, 78F0885, and 78F0886: TI001

1.6 Block Diagram



1.7 Outline of Functions

(1/2)

Item		μPD78F0887	μPD78F0888	μPD78F0889	μPD78F0890
Internal memory (bytes)	Flash memory (self-programming supported)[Note]	48 K	60 K	96 K	128 K
	Bank	–	–	4	6
	High-speed RAM[Note]	1 K			
	Expansion RAM[Note]	2 K	2 K	4 K	6 K
Memory space		64 KB			
High-speed system clock (oscillation frequency)		Crystal/ceramic oscillation (X_1), external main system clock input (EXCLK) (4 to 20 MHz: V_{DD} = 4.0 to 5.5 V, 4 to 10 MHz: V_{DD} = 2.7 to 5.5 V, 4 to 5 MHz: V_{DD} = 1.8 to 5.5 V)			
Internal high-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V)			
Internal low-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (240 kHz (TYP.))			
Subsystem clock (oscillation frequency)		Crystal oscillation (XT1), external subsystem clock input (EXCLKS) (32.768 kHz: V_{DD} = 1.8 to 5.5 V)			
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)			
Minimum instruction execution time		0.1 μs/0.2 μs/0.4 μs/0.8 μs/1.6 μs (high-speed system clock: @ f_{XP} = 20 MHz operation)			
		0.25 μs/0.5 μs/1.0 μs/2.0 μs/4.0 μs (TYP.) (internal oscillator clock: @ f_{RH} = 8 MHz (TYP.) operation)			
		122 μs (subsystem clock: when operating at f_{XT} = 32.768 kHz)			
Instruction set		• 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		Total: 55 CMOS I/O 50 CMOS output 1 N-ch open-drain I/O 4			
Timers		• 16-bit timer/event counter: 4 channels • 8-bit timer/event counter: 2 channels • 8-bit timer: 2 channels • Watch timer 1 channel • Watchdog timer: 1 channel			
	Timer outputs	8 (PWM output: 4)			
Clock output		• 78.125 kHz, 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (high-speed system clock: 10 MHz) • 32.768 kHz (subsystem clock: 32.768 kHz)			
Buzzer output		1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 kHz (high-speed system clock: 10 MHz)			
A/D converter		10-bit resolution × 12 channels			

Note The internal flash memory capacity, internal high-speed RAM capacity, and internal expansion RAM capacity can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

(2/2)

Item		μPD78F0887	μPD78F0888	μPD78F0889	μPD78F0890
Serial interface	CAN	1 ch			
	3-wire CSI	1 ch			
	LIN-UART	1 ch			
	LIN-UART/ CSI Note	1 ch			
Multiplier/divider		• 16 bit x 16 bit = 32 bit (Multiplication) • 32 bit ÷ 32 bit = 32 bit remainder of 16 bits (Division)			
<R> Vectored interrupt sources	Internal	29			
	External	8			
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector			
On-chip debug function		Provided			
Supply voltage		V_{DD} = 1.8 to 5.5 V			
Operating ambient temperature		T_A = −40 to +85°C, −40 to +125°C			
Package		• 64-pin plastic LQFP(10x10) • 64-pin plastic LQFP(12x12)			

Note Select either of the functions of these alternate-function pins.

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters 00 to 03				8-Bit Timer/ Event Counters 50 and 51		8-Bit Timers H0 and H1		Watch Timer	Watchdog Timer
		TM00	TM01	TM02	TM03	TM50	TM51	TMH0	TMH1		
Operation mode	Interval timer	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 channel Note	1 channel
	External event counter	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	–	–	–	–
Function	Timer output	1	1	1	1	1	1	1	1	–	–
	PPG output	1	1	1	1	–	–	–	–	–	–
	PWM output	–	–	–	–	1	1	1	1	–	–
	Pulse width measurement	2	2	2	2	–	–	–	–	–	–
	Square-wave output	1	1	1	1	1	1	1	1	–	–
	Interrupt source	2	2	2	2	1	1	1	1	1	–

Note In the watch timer, the watch timer function and interval timer function can be used simultaneously.

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

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

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

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

Power Supply	Corresponding Pins
AV_{REF}	P80 to P87, P90 to P93
EV_{DD}	Port pins other than P80 to P87, P90 to P93 and P121 to P124
V_{DD}	• P121 to P124 • Non-port pins

This section explains the names and functions of the pins of the 78K0/FE2.

(1) Port pins

Table 2-2. Port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI000
P01				TI010/TO00
P05				$\overline{SSI11}$/TI001
P06				TI011/TO01
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	$\overline{SCK10}$/TxD61
P11				SI10/RxD61
P12				SO10
P13				TxD60
P14				RxD60
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP1
P31				INTP2/TI002
P32				INTP3/TI012/TO02
P33				INTP4/TI51/TO51
P40 to P43	I/O	Port 4. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	–

Table 2-2. Port pins (2/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P50 to P53	I/O	Port 5. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	–
P60 to P63	I/O	Port 6. 4-bit I/O port Input/output can be specified in 1-bit units.	N-ch open drain I/O port.	Input	–
P70	I/O	Port 7. 7-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.		Input	CTxD
P71					CRxD
P72					PCL/INTP6
P73					BUZ/INTP7
P74					SO11
P75					SI11
P76					$\overline{\text{SCK11}}$
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units.		Input	ANI0 to ANI7
P90 to P93	I/O	Port 9. 4-bit I/O port. Input/output can be specified in 1-bit units.		Input	ANI8 to ANI11
P120	I/O	Port 12. 5-bit I/O port. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.		Input	INTP0/EXLVI
P121					X1
P122					X2/EXCLK
P123					XT1
P124					XT2/EXCLKS
P130	Output	Port 13. P130 is 1-bit output-only port. P131 and P132 are 2-bit I/O port. P131 and P132 use of an on-chip pull-up resistor can be specified by a software setting.		Output	–
P131	I/O			Input	
P132					

(2) Non-port pins

Table 2-3. Non-port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P120/EXLVI
INTP1				P30
INTP2				P31/TI002
INTP3				P32/TI012/TO02
INTP4				P33/TI51/TO51
INTP5				P16/TOH1
INTP6				P72/PCL
INTP7				P73/BUZ
SI10	Input	Serial data input to serial interface	Input	P11/RxD61
SI11				P75
SO10	Output	Serial data output from serial interface	Input	P12
SO11				P74
$\overline{\text{SCK10}}$	I/O	Clock input/output for serial interface	Input	P10/TxD61
$\overline{\text{SCK11}}$				P76
$\overline{\text{SSI11}}$	Input	Serial interface chip select input	Input	P05/TI001
RxD60	Input	Serial data input to asynchronous serial interface	Input	P14
RxD61				P11/SI10
TxD60	Output	Serial data output from asynchronous serial interface	Input	P13
TxD61				P10/$\overline{\text{SCK10}}$
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	P00
TI001		External count clock input to 16-bit timer/event counter 01 Capture trigger input to capture registers (CR001, CR011) of 16-bit timer/event counter 01		P05/$\overline{\text{SSI11}}$
TI002		External count clock input to 16-bit timer/event counter 02 Capture trigger input to capture registers (CR002, CR012) of 16-bit timer/event counter 02		P31/INTP2
TI003		External count clock input to 16-bit timer/event counter 03 Capture trigger input to capture registers (CR003, CR013) of 16-bit timer/event counter 03		P131
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TI011		Capture trigger input to capture register (CR001) of 16-bit timer/event counter 01		P06/TO01
TI012		Capture trigger input to capture register (CR002) of 16-bit timer/event counter 02		P32/TO02/INTP3
TI013		Capture trigger input to capture register (CR003) of 16-bit timer/event counter 03		P132/TO03
TO00	Output	16-bit timer/event counter 00 output	Input	P01/TI010
TO01		16-bit timer/event counter 01 output		P06/TI011
TO02		16-bit timer/event counter 02 output		P32/TI012/INTP3

Table 2-3. Non-port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
TO03	Output	16-bit timer/event counter 03 output	Input	P132/TI013
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P17/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P33/TO51/INTP4
TO50	Output	8-bit timer/event counter 50 output	Input	P17/TI50
TO51		8-bit timer/event counter 51 output		P33/TI51/INTP4
TOH0		8-bit timer H0 output		P15
TOH1		8-bit timer H1 output		P16/INTP5
PCL	Output	Clock output (for trimming of high-speed system clock, subsystem clock)	Input	P72/INTP6
BUZ	Output	Buzzer output	Input	P73/INTP7
ANI0 to ANI11	Input	A/D converter analog input	Input	P80 to P87 P90 to P93
CTxD	Input	CAN transmit data output	Input	P70
CRxD	Output	CAN receive data input	Input	P71
AV_{REF}	Input	A/D converter reference voltage input and positive power supply for port 2	–	–
AV_{SS}	–	A/D converter ground potential. Make the same potential as EV_{SS} or V_{SS}.	–	–
$\overline{RESET}$	Input	System reset input	–	–
X1	Input	Connecting resonator for high-speed system clock	Input	P121
X2	–		Input	P122/EXCLK
XT1	Input	Connecting resonator for subsystem clock	Input	P123
XT2	–		Input	P124/EXCLKS
EXCLK	Input	External clock input for main system clock	Input	P122/X2
EXCLKS	Input	External clock input for subsystem clock	Input	P124/XT2
EXLVI	Input	Potential input for external low-voltage detection	Input	P120/INTP0
V_{DD}	–	Positive power supply (except for ports)	–	–
EV_{DD}	–	Positive power supply for ports	–	–
V_{SS}	–	Ground potential (except for ports)	–	–
EV_{SS}	–	Ground potential for ports	–	–
FLMD0	–	Flash memory programming mode setting.	–	–
REGC	–	This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF: recommended).	–	–

2.2 Description of Pin Functions

2.2.1 P00, P01, P05, P06 (port 0)

P00, P01, P05 and P06 function as a 4-bit I/O port. These pins also function as timer I/O and serial interface chip select input.

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

(1) Port mode

P00, P01, P05 and P06 function as 4-bit I/O port. P00, P01, P05 and P06 can be set to input or output 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, P01, P05 and P06 function as timer I/O, and serial interface chip select input.

(a) TI000, TI001

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

(b) TI010, TI011

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

(c) TO00, TO01

These are timer output pins.

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

This is the serial interface chip select input pin.

2.2.2 P10 to P17 (port 1)

P10 to P17 function as an 8-bit I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

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

(1) Port mode

P10 to P17 function as an 8-bit I/O port. P10 to P17 can be set to input or output in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

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

(a) SI10

This is a serial interface serial data input pin.

(b) SO10

This is a serial interface serial data output pin.

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

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

(d) RxD60, RxD61

These are the serial data input pins of the asynchronous serial interface.

(e) TxD60, TxD61

These are the serial data output pins of the asynchronous serial interface.

(f) TI50

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

(g) TO50, TOH0, and TOH1

These are timer output pins.

(h) INTP5

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

2.2.3 P30 to P33 (port 3)

P30 to P33 function as a 4-bit I/O port. These pins also function as pins for external interrupt request input and timer I/O.

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

(1) Port mode

P30 to P33 function as a 4-bit I/O port. P30 to P33 can be set to input or output in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30 to P33 function as external interrupt request input pins and timer I/O pins.

(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) TI002

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

(c) TI012

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

(d) TO02

This is a timer output pin.

(e) TI51

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

(f) TO51

This is a timer output pin.

Cautions 1. Be sure to pull the P31/TI002/INTP2 pin down before a reset release, to prevent malfunction.

2. Connect P31/TI002/INTP2 as follows when writing the flash memory with a flash programmer.
- P31/TI002/INTP2: Connect to EV_{SS} via a resistor (10 kΩ: recommended).
The above connection is not necessary when writing the flash memory by means of self programming.

Remark P31/TI002/INTP2 and P32/TI012/TO02/INTP3 can be used as on-chip debug mode setting pins when the on-chip debug function is used. For details, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

2.2.4 P40 to P43 (port 4)

P40 to P43 function as a 4-bit I/O port. P40 to P43 can be set to input or output in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

2.2.5 P50 to P53 (port 5)

P50 to P53 function as a 4-bit I/O port. P50 to P53 can be set to input or output in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

2.2.6 P60 to P63 (port 6)

P60 to P63 function as a 4-bit I/O port. P60 to P63 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

P60 to P63 are N-ch open-drain pins.

2.2.7 P70 to P76 (port 7)

P70 to P76 function as a 7-bit I/O port. These pins also function as external interrupt request input, clock output pins, buzzer output pins, CAN I/F I/O, serial interface data I/O and clock I/O.

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

(1) Port mode

P70 to P76 function as a 7-bit I/O port. P70 to P76 can be set to input or output in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

P70 to P77 function as external interrupt request input, output pins, buzzer output pins, CAN I/F I/O, serial interface data I/O and clock I/O.

(a) INTP6, INTP7

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) CRxD

This is the CAN serial receive data input pin.

(c) CTxD

This is the CAN serial transmit data output pin.

(d) PCL

This is a clock output pin.

(e) BUZ

This is a buzzer output pin.

(f) SI11

This is a serial interface serial data input pin.

(g) SO11

This is a serial interface serial data output pin.

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

This is the serial interface serial clock I/O pin.

2.2.8 P80 to P87 (port 8)

P80 to P87 function as an 8-bit I/O port. These pins also function as pins for A/D converter analog input.

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

(1) Port mode

P80 to P87 function as an 8-bit I/O port. P80 to P87 can be set to input or output in 1-bit units using port mode register 8 (PM8).

(2) Control mode

P80 to P87 function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, see **(5) P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11** in **13.6 Cautions for A/D Converter**.

Caution P80/ANI0 to P87/ANI7 is set in the analog input mode after release of reset.

2.2.9 P90 to P93 (port 9)

P90 to P93 function as an 4-bit I/O port. These pins also function as pins for A/D converter analog input.

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

(1) Port mode

P90 to P93 function as an 4-bit I/O port. P90 to P93 can be set to input or output in 1-bit units using port mode register 9 (PM9).

(2) Control mode

P90 to P93 function as A/D converter analog input pins (ANI8 to ANI11). When using these pins as analog input pins, see **(5) P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11** in **13.6 Cautions for A/D Converter**.

Caution P90/ANI8 to P93/ANI11 is set in the analog input mode after release of reset.

2.2.10 P120 to P124 (port 12)

P120 to P124 function as a 5-bit I/O port. These pins also function as pins for external interrupt request input, external clock input for main system clock, external clock input for subsystem clock and potential input for external low-voltage detection. The following operation modes can be specified in 1-bit units.

(1) Port mode

P120 to P124 function as a 5-bit I/O port. P120 to P124 can be set to input or output using port mode register 12 (PM12). Only for P120, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

(2) Control mode

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

(a) INTP0

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

(b) EXLVI

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

(c) X1, X2

These are the pins for connecting a resonator for high-speed system clock.

When supplying an external clock, input a signal to the X1 pin and input the inverse signal to the X2 pin.

Caution Connect P121/X1 as follows when writing the flash memory with a flash programmer.
- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).
The above connection is not necessary when writing the flash memory by means of self programming.

Remark The X1 and X2 pins can be used as on-chip debug mode setting pins when the on-chip debug function is used. For details, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

(d) EXCLK

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

(e) XT1, XT2

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

When supplying an external clock, input a signal to the XT1 pin and input the inverse signal to the XT2 pin.

(f) EXCLKS

This is an external clock input pin for subsystem clock.

2.2.11 P130 to P132 (port 13)

P130 functions as a 1-bit output-only port. P131 and P132 function as a 2-bit I/O port. These pins also function as pins for timer I/O. The following operation modes can be specified in 1-bit units.

(1) Port mode

P131 and P132 can be set to input or output in 1 bit units using port mode register 13 (PM13). P131 and P132 use of an on-chip pull-up resistor can be specified by pull-up resistor option register 13 (PU13).

(2) Control mode

P130, P131 and P132 function as timer I/O and serial interface chip select input.

(a) TI003

This is the pin for inputting an external count clock to 16-bit timer/event counter 03 and is also for inputting a capture trigger signal to the capture registers (CR003, CR013) of 16-bit timer/event counter 03.

(b) TI013

This is the pin for inputting a capture trigger signal to the capture register (CR003) of 16-bit timer/event counter 03.

(c) TO03

This is a timer output pin.

2.2.12 AV_{REF}

This is the A/D converter reference voltage input pin.

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

Note Connect port 8 and port 9 directly to EV_{DD} when it is used as a digital port.

2.2.13 AV_{SS}

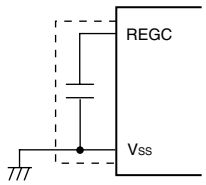
This is the A/D converter ground potential pin. Even when the A/D converter is not used, always use this pin with the same potential as the EV_{SS} pin or V_{SS} pin.

2.2.14 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

2.2.15 REGC

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



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

2.2.16 V_{DD} and EV_{DD}

V_{DD} is the positive power supply pin for other than ports.

EV_{DD} is the positive power supply pin for ports.

2.2.17 V_{SS} and EV_{SS}

V_{SS} is the ground potential pin for other than ports.

EV_{SS} is the ground potential pin for ports.

2.2.18 FLMD0

This is a pin for setting flash memory programming mode.

Connect to EV_{SS} or V_{SS} in the normal operation mode. In flash memory programming mode, be sure to connect this pin to the flash programmer.

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

Table 2-4 shows 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 2-4. Pin I/O Circuit Types (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01/TI010/TO00			
P05/$\overline{SSI11}$/TI001			
P06/TI011/TO01			
P10/$\overline{SCK10}$/TxD61			
P11/SI10/RxD61			
P12/SO10	5-H		
P13/TxD60			
P14/RxD60	5-AH		
P15/TOH0	5-H		
P16/TOH1/INTP5	5-AH		
P17/TI50/TO50			
P30/INTP1			
P31/TI002/INTP2 Note			
P32/TI012/TO02/INTP3			
P33/TI51/TO51/INTP4			
P40 to P43	5-H		
P50 to P53			
P60 to P63	13-P		Input: Connect to EV_{SS}. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P70/CTxD	5-H		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P71/CRxD	5-AH		
P72/PCL/INTP6			
P73/BUZ/INTP7			
P74/SO11	5-H		
P75/SI11	5-AH		
P76/$\overline{SCK11}$			

Note Connect P31/TI002/INTP2 as follows when writing the flash memory with a flash programmer.

- P31/TI002/INTP2: Connect to EV_{SS} via a resistor (10 kΩ: recommended).

The above connection is not necessary when writing the flash memory by means of self programming.

Table 2-4. Pin I/O Circuit Types (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P80/ANI0 to P87/ANI7[Note 1]	11-G	I/O	<Analog setting> Connect to AV_{REF} or AV_{SS}. <Digital setting> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P90/ANI8 to P93/ANI11[Note 1]			
P120/INTP0/EXLVI	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P121/X1[Note 2, 3]	37	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P122/X2/EXCLK[Note 2]			
P123/XT1[Note 2]			
P124/XT2/EXCLKS[Note 2]			
P130	3-C	Output	Leave open.
P131/TI003	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P132/TI013/TO03			
$\overline{RESET}$	2	Input	Connect to EV_{DD} or V_{DD}.
AV_{REF}	–	–	Connect directly to EV_{DD} or V_{DD}[Note 4].
AV_{SS}			Connect directly to EV_{SS} or V_{SS}.
FLMD0			Connect to EV_{SS} or V_{SS}.

Notes 1. P80/ANI0 to P87/ANI7 and P90/ANI8 to P93/ANI11 are set in the analog input mode after release of reset.

2. Use the recommended connection above in I/O port mode (see **Figure 6-6 Format of Clock Operation Mode Select Register (OSCCTL)**) when these pins are not used.

3. Connect P121/X1 as follows when writing the flash memory with a flash programmer.
- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).

The above connection is not necessary when writing the flash memory by means of self programming.

4. Connect port 8 and port 9 directly to EV_{DD} when it is used as a digital port.

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

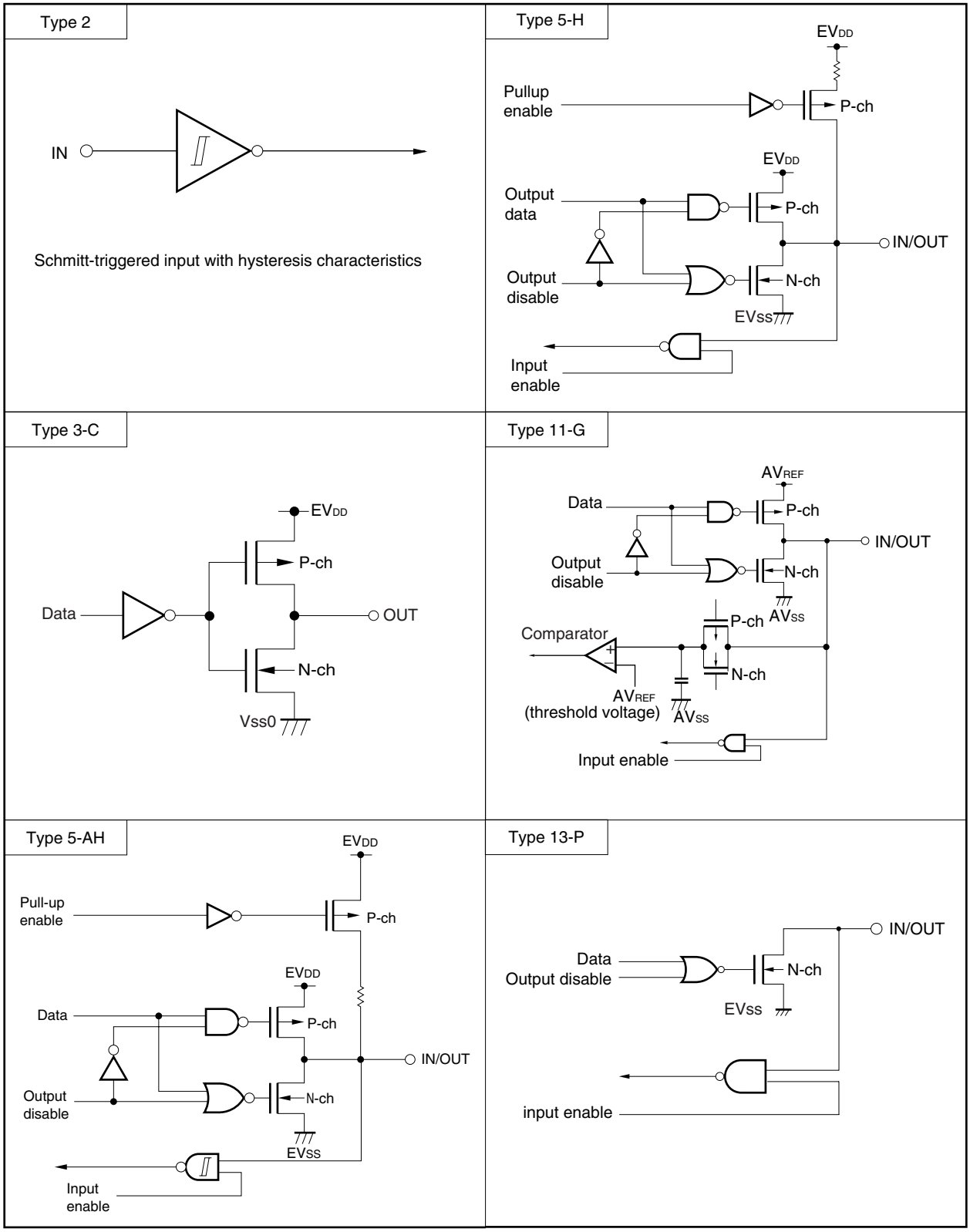
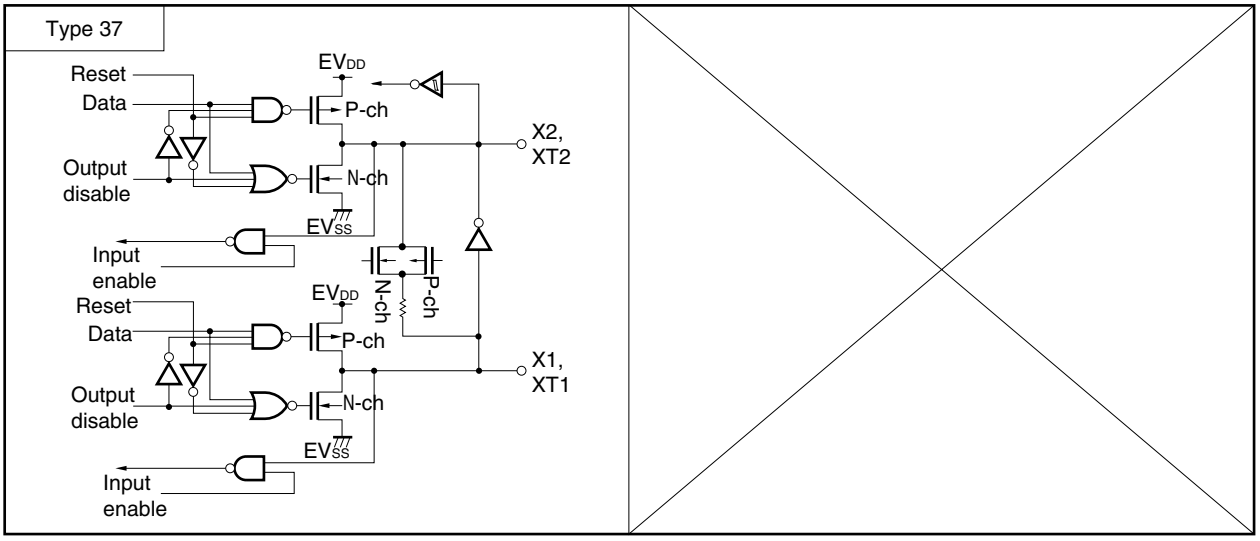


Figure 2-1. Pin I/O Circuit List (2/2)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the 78K0/FE2 can each access a 64 KB memory space. Figures 3-1 to 3-4 show the memory map.

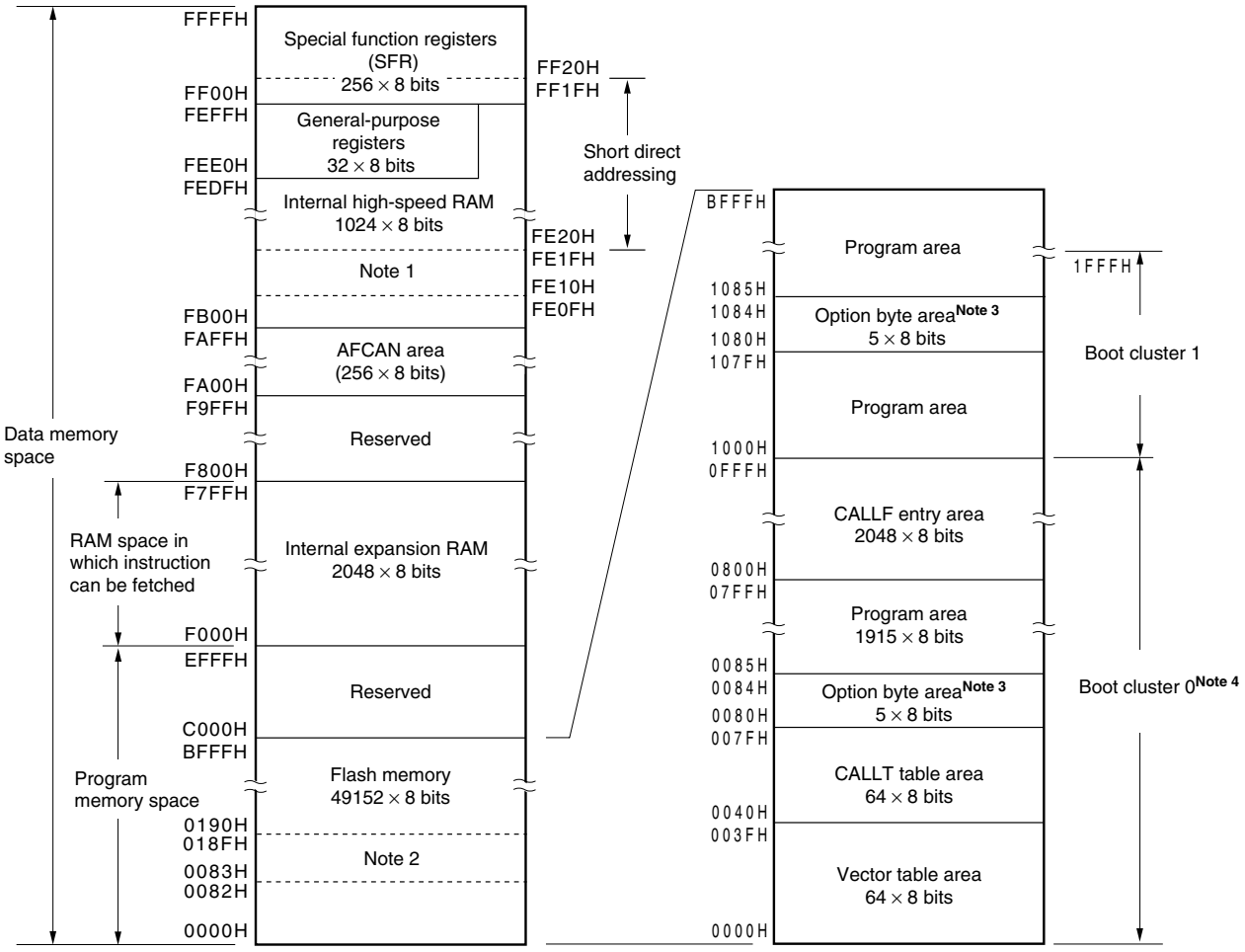
Caution Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of the 78K0/FE2 is fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

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

Flash Memory Version	IMS	IXS
μPD78F0887	CCH	08H
μPD78F0888	CFH	08H
μPD78F0889	CCH[Note]	04H
μPD78F0890	CCH[Note]	00H

Note The μPD78F0889 and μPD78F0890 have internal ROMs of 96 KB and 128 KB, respectively. However, the set value of IMS of these devices is the same as those of the 48 KB product because banks are used. For how to set the banks, see **4.3 Memory Bank Select Register (BANK)**.

Figure 3-1. Memory Map (μPD78F0887)



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.8 Security Setting**).

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

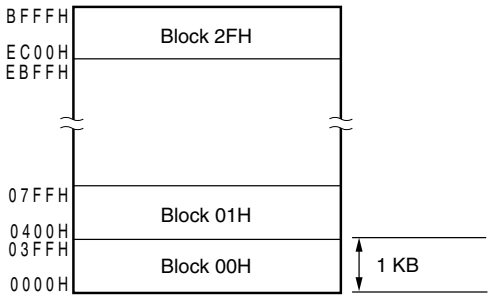
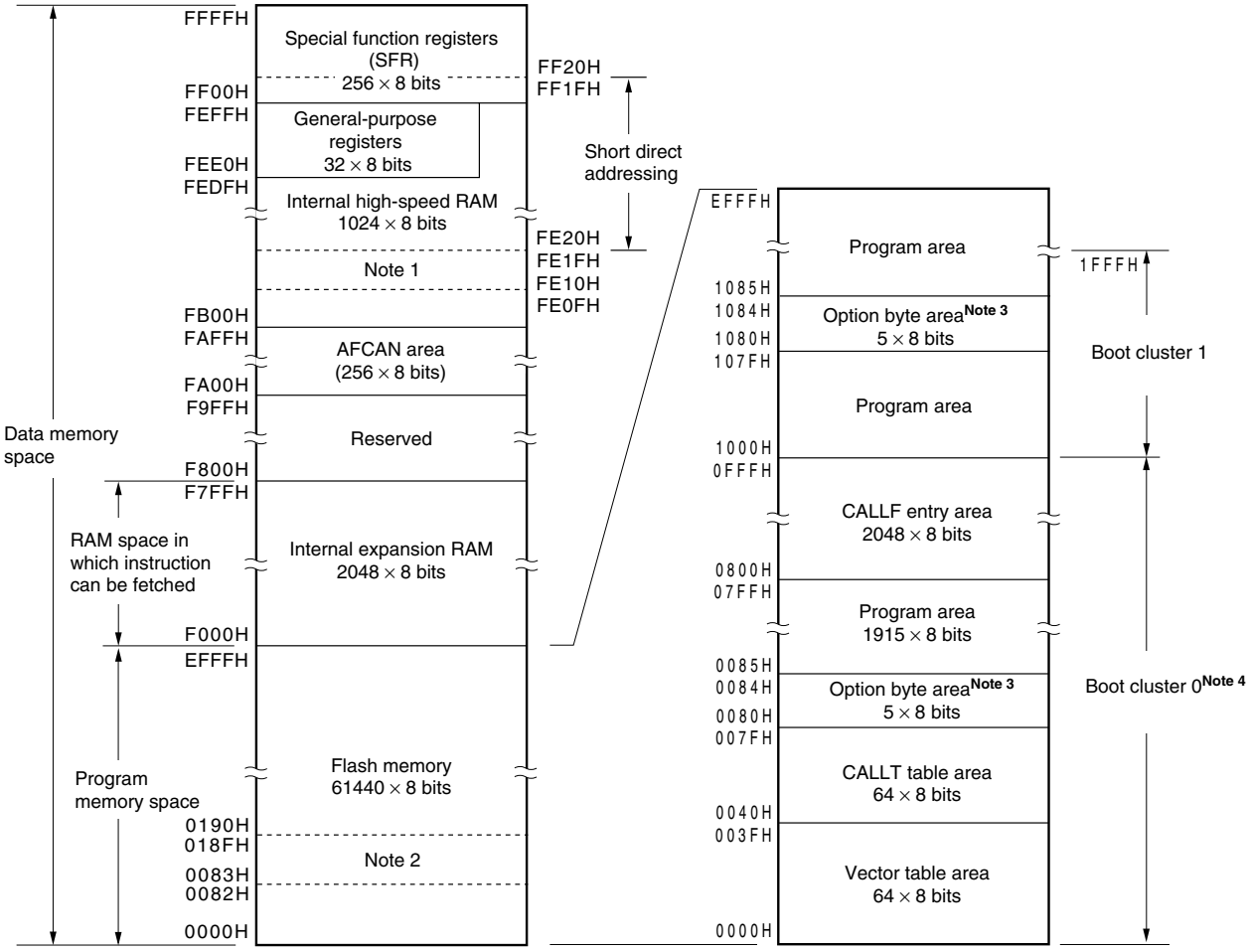


Figure 3-2. Memory Map (μPD78F0888)



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.8 Security Setting**).

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

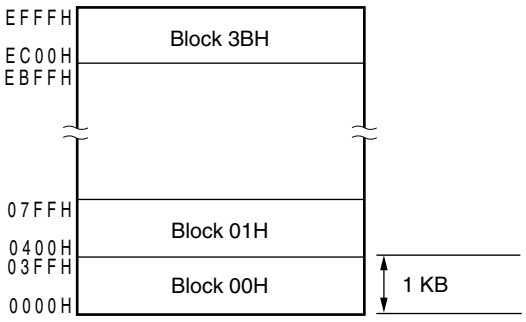
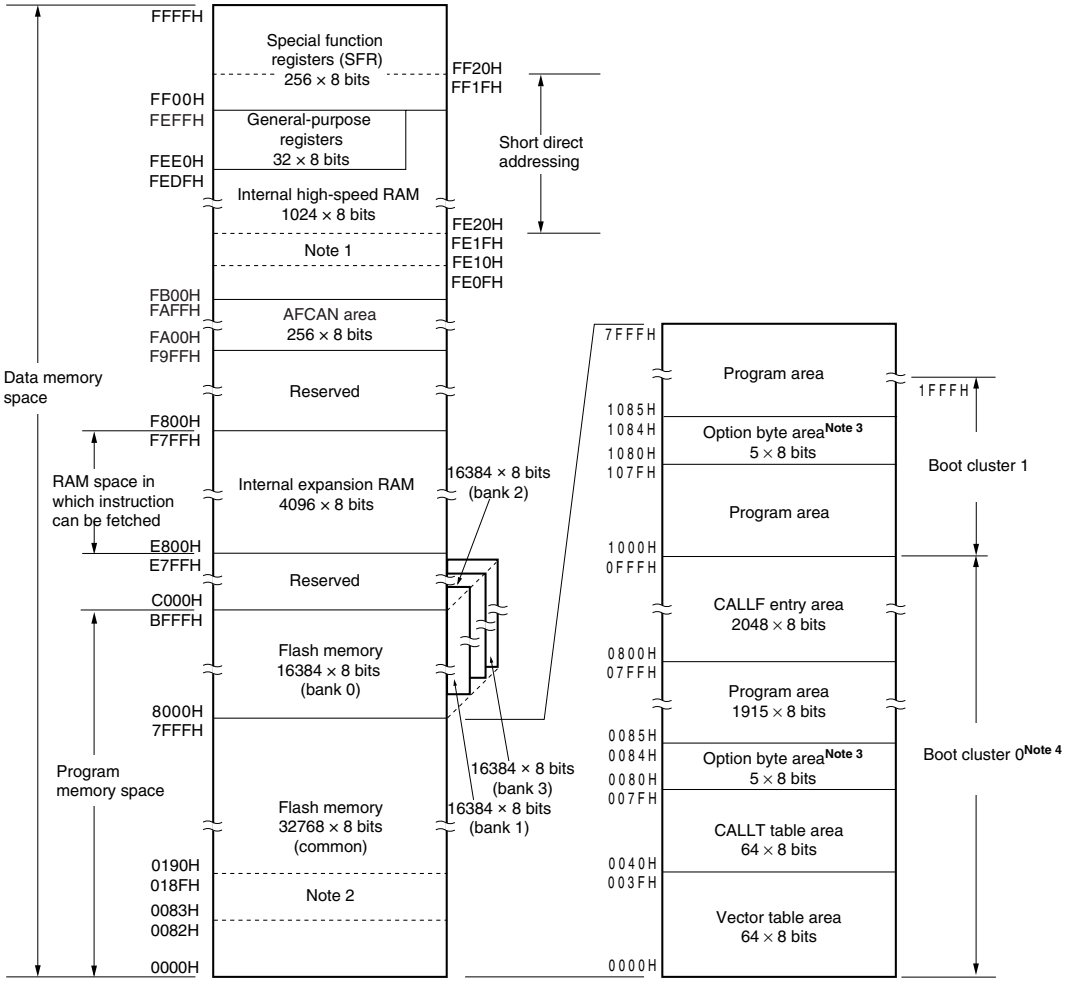


Figure 3-3. Memory Map (μPD78F0889)



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.8 Security Setting**).

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

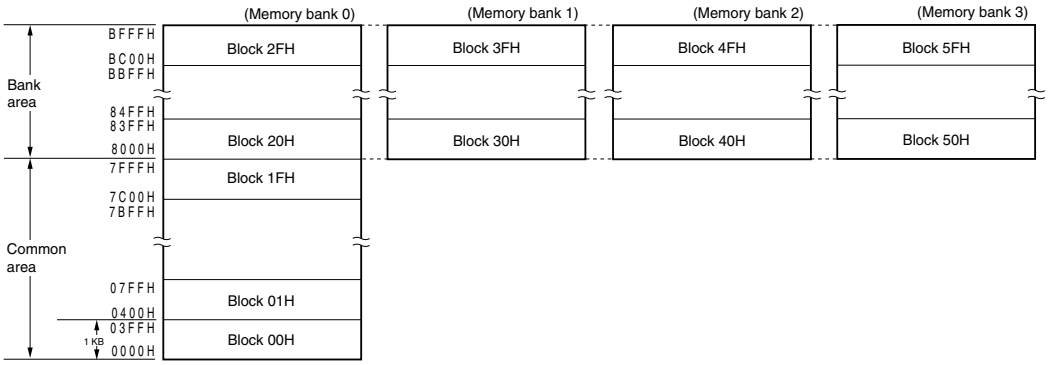
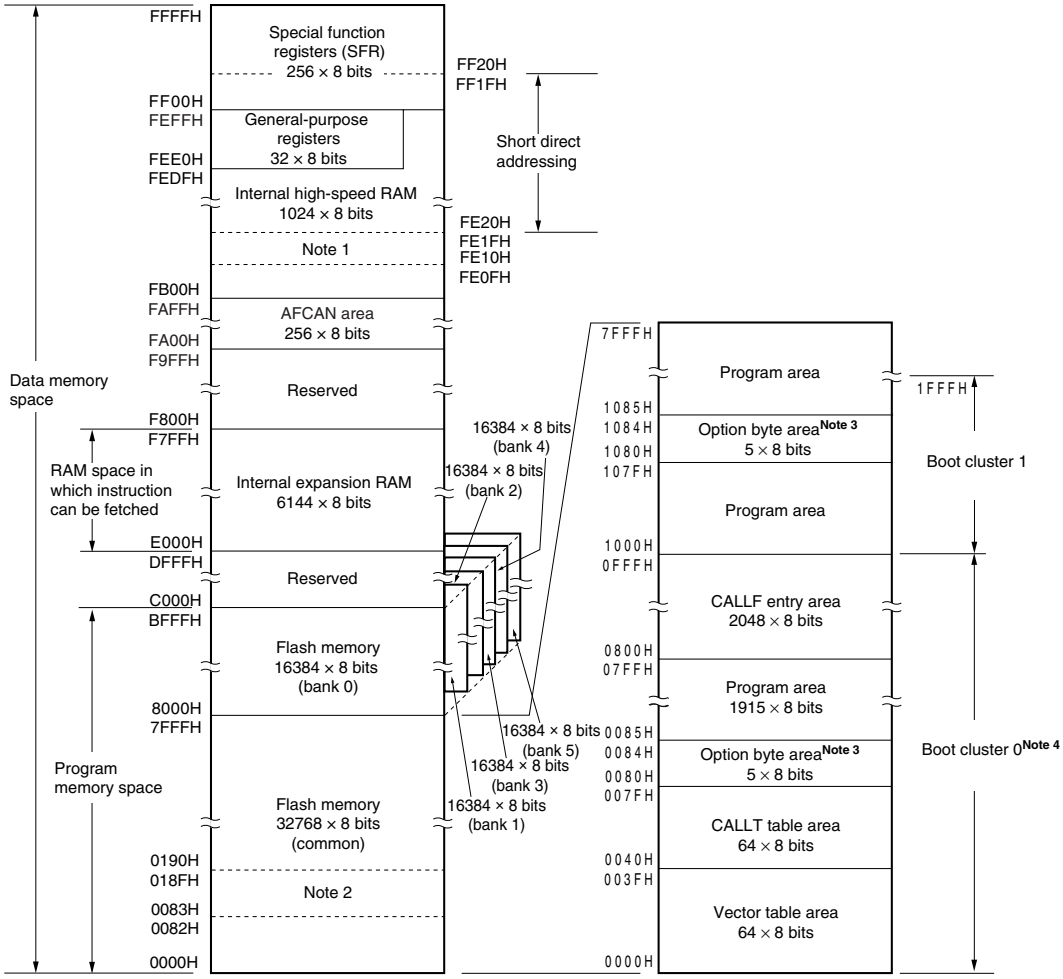


Figure 3-4. Memory Map (μPD78F0890)



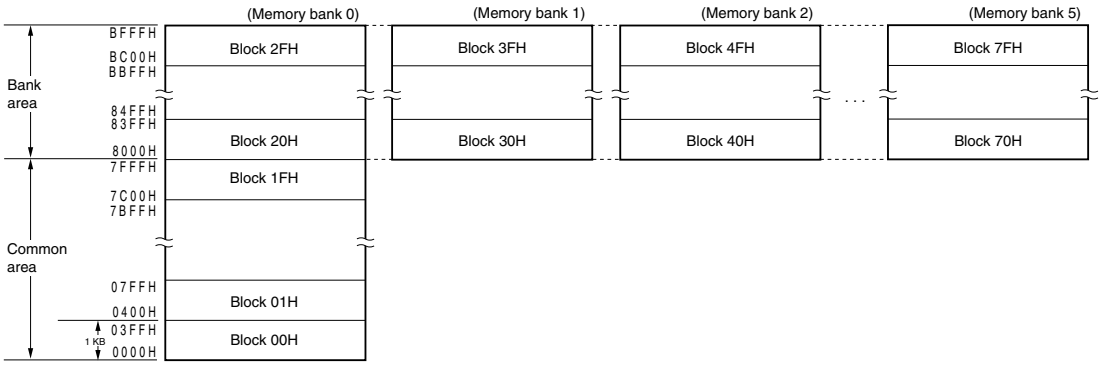
Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.8 Security Setting**).

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



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

Table 3-2. Correspondence Between Address Values and Block Numbers in Flash Memory (1/2)

(1) μPD78F0887, 78F0888

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

Remark μPD78F0887: Block numbers 00H to 2FH
μPD78F0888: Block numbers 00H to 3BH

Table 3-2. Correspondence Between Address Values and Block Numbers in Flash Memory (2/2)

(2) μPD78F0889, 78F0890

Address Value	Block Number	Address Value	Memory Bank	Block Number	Address Value	Memory Bank	Block Number	Address Value	Memory Bank	Block Number
0000H to 03FFH	00H	8000H to 83FFH	0	20H	8000H to 83FFH	2	40H	8000H to 83FFH	4	60H
0400H to 07FFH	01H	8400H to 87FFH		21H	8400H to 87FFH		41H	8400H to 87FFH		61H
0800H to 0BFFH	02H	8800H to 8BFFH		22H	8800H to 8BFFH		42H	8800H to 8BFFH		62H
0C00H to 0FFFH	03H	8C00H to 8FFFH		23H	8C00H to 8FFFH		43H	8C00H to 8FFFH		63H
1000H to 13FFH	04H	9000H to 93FFH		24H	9000H to 93FFH		44H	9000H to 93FFH		64H
1400H to 17FFH	05H	9400H to 97FFH		25H	9400H to 97FFH		45H	9400H to 97FFH		65H
1800H to 1BFFH	06H	9800H to 9BFFH		26H	9800H to 9BFFH		46H	9800H to 9BFFH		66H
1C00H to 1FFFH	07H	9C00H to 9FFFH		27H	9C00H to 9FFFH		47H	9C00H to 9FFFH		67H
2000H to 23FFH	08H	A000H to A3FFH		28H	A000H to A3FFH		48H	A000H to A3FFH		68H
2400H to 27FFH	09H	A400H to A7FFH		29H	A400H to A7FFH		49H	A400H to A7FFH		69H
2800H to 2BFFH	0AH	A800H to ABFFH		2AH	A800H to ABFFH		4AH	A800H to ABFFH		6AH
2C00H to 2FFFH	0BH	AC00H to AFFFH		2BH	AC00H to AFFFH		4BH	AC00H to AFFFH		6BH
3000H to 33FFH	0CH	B000H to B3FFH		2CH	B000H to B3FFH		4CH	B000H to B3FFH		6CH
3400H to 37FFH	0DH	B400H to B7FFH		2DH	B400H to B7FFH		4DH	B400H to B7FFH		6DH
3800H to 3BFFH	0EH	B800H to BBFFH		2EH	B800H to BBFFH		4EH	B800H to BBFFH		6EH
3C00H to 3FFFH	0FH	BC00H to BFFFH		2FH	BC00H to BFFFH		4FH	BC00H to BFFFH		6FH
4000H to 43FFH	10H	8000H to 83FFH	1	30H	8000H to 83FFH	3	50H	8000H to 83FFH	5	70H
4400H to 47FFH	11H	8400H to 87FFH		31H	8400H to 87FFH		51H	8400H to 87FFH		71H
4800H to 4BFFH	12H	8800H to 8BFFH		32H	8800H to 8BFFH		52H	8800H to 8BFFH		72H
4C00H to 4FFFH	13H	8C00H to 8FFFH		33H	8C00H to 8FFFH		53H	8C00H to 8FFFH		73H
5000H to 53FFH	14H	9000H to 93FFH		34H	9000H to 93FFH		54H	9000H to 93FFH		74H
5400H to 57FFH	15H	9400H to 97FFH		35H	9400H to 97FFH		55H	9400H to 97FFH		75H
5800H to 5BFFH	16H	9800H to 9BFFH		36H	9800H to 9BFFH		56H	9800H to 9BFFH		76H
5C00H to 5FFFH	17H	9C00H to 9FFFH		37H	9C00H to 9FFFH		57H	9C00H to 9FFFH		77H
6000H to 63FFH	18H	A000H to A3FFH		38H	A000H to A3FFH		58H	A000H to A3FFH		78H
6400H to 67FFH	19H	A400H to A7FFH		39H	A400H to A7FFH		59H	A400H to A7FFH		79H
6800H to 6BFFH	1AH	A800H to ABFFH		3AH	A800H to ABFFH		5AH	A800H to ABFFH		7AH
6C00H to 6FFFH	1BH	AC00H to AFFFH		3BH	AC00H to AFFFH		5BH	AC00H to AFFFH		7BH
7000H to 73FFH	1CH	B000H to B3FFH		3CH	B000H to B3FFH		5CH	B000H to B3FFH		7CH
7400H to 77FFH	1DH	B400H to B7FFH		3DH	B400H to B7FFH		5DH	B400H to B7FFH		7DH
7800H to 7BFFH	1EH	B800H to BBFFH		3EH	B800H to BBFFH		5EH	B800H to BBFFH		7EH
7C00H to 7FFFH	1FH	BC00H to BFFFH		3FH	BC00H to BFFFH		5FH	BC00H to BFFFH		7FH

Remark μPD78F0889: Block numbers 00H to 5FH
μPD78F0890: Block numbers 00H to 7FH

3.1.1 Internal program memory space

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

78K0/FE2 products incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F0887	Flash memory	49152 × 8 bits (0000H to BFFFH)
μPD78F0888		61440 × 8 bits (0000H to EFFFH)
μPD78F0889		98304 × 8 bits (0000H to 7FFFH (common area: 32 KB) + 8000H to BFFFH (bank area: 16 KB) × 4)
μPD78F0890		131072 × 8 bits (0000H to 7FFFH (common area: 32 KB) + 8000H to BFFFH (bank area: 16 KB) × 6)

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

(1) Vector code area

The 64-byte area 0000H to 003FH is reserved as a Vector code area. The program start addresses for branch upon reset signal input or generation of each interrupt request are stored in the Vector code 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 Code

Vector Code Address	Interrupt Source	Vector Code Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	0020H	INTCSI10/INTSRE61
		0022H	INTP6/INTSR61
0004H	INTLVI	0024H	INTP7/INTST61
0006H	INTP0	0026H	INTTMH1
0008H	INTP1	0028H	INTTMH0
000AH	INTP2/INTTM002	002AH	INTTM50
000CH	INTP3/INTTM012	002CH	INTTM000
000EH	INTP4/INTTM003	002EH	INTTM010
0010H	INTP5/INTTM013	0030H	INTAD
0012H	INTC0ERR	0032H	INTWTI/INTDMU
0014H	INTC0WUP	0034H	INTTM51
0016H	INTC0REC	0036H	INTWT
0018H	INTC0TRX	0038H	INTCSI11
001AH	INTSRE60	003AH	INTTM001
001CH	INTSR60	003CH	INTTM011
001EH	INTST60	003EH	BRK

(2) CALLT instruction table area

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

(3) Option byte area

The option byte area is assigned to the 1-byte area of 0080H. Refer to **CHAPTER 23 OPTION BYTE** for details.

(4) CALLF instruction entry area

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

(5) On-chip debug security ID setting area

A 10-byte area of 0085H to 008EH and 1085H to 108EH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 0085H to 008EH when the boot swap is not used and at 0085H to 008EH and 1085H to 108EH when the boot swap is used. For details, see **CHAPTER 25 ON-CHIP DEBUG FUNCTION**.

3.1.2 Bank area (μPD78F0889 and 78F0890 only)

The μPD78F0889 has bank areas 0 to 3 and the μPD78F0890 has bank areas 0 to 5 as illustrated below.

The banks are selected by a bank select register (BANK) (see **4.3 Memory Bank Select Register (BANK)**).

Cautions 1. Instructions cannot be fetched between different memory banks.

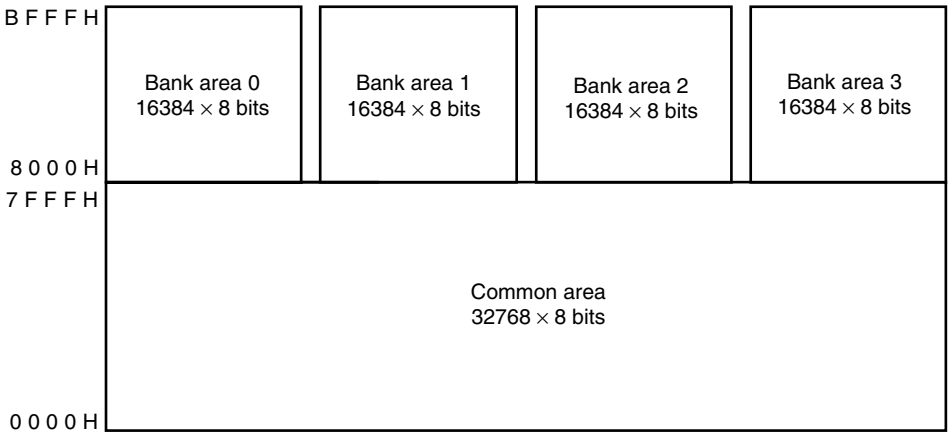
2. Branch and access cannot be directly executed between different memory banks. Execute branch or access between different memory banks via the common area.

3. Allocate interrupt servicing in the common area.

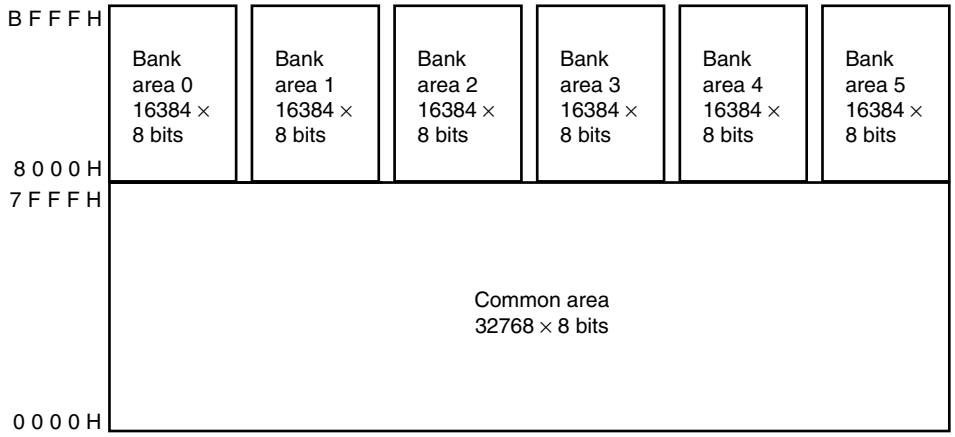
4. An instruction that extends from 7FFFH to 8000H can only be executed in memory bank 0.

Figure 3-5. Internal ROM (Flash Memory) Configuration

(a) μPD78F0889



(b) μPD78F0890



The following table shows the relations among bank numbers, CPU addresses, and real addresses of the flash memory.

Table 3-5. Bank Numbers, CPU Addresses, and Real Addresses of Flash Memory

(a) μPD78F0889

Bank No.	CPU Address	Real Address of Flash Memory
–	0000H to 7FFFH (common area)	00000H to 07FFFH
0	8000H to BFFFH	08000H to 0BFFFH
1		0C000H to 0FFFFH
2		10000H to 13FFFH
3		14000H to 17FFFH
4 or more	Setting prohibited	

(b) μPD78F0890

Bank No.	CPU Address	Real Address of Flash Memory
–	0000H to 7FFFH (common area)	00000H to 07FFFH
0	8000H to BFFFH	08000H to 0BFFFH
1		0C000H to 0FFFFH
2		10000H to 13FFFH
3		14000H to 17FFFH
4		18000H to 1BFFFH
5		1C000H to 1FFFFH
6 or more	Setting prohibited	

3.1.3 Internal data memory space

78K0/FE2 products incorporate the following RAM.

(1) Internal high-speed RAM

Table 3-6. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD78F0887	1024 × 8 bits (FB00H to FEFFH)
μPD78F0888	
μPD78F0889	
μPD78F0890	

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

(2) Internal expansion RAM

Table 3-7. Internal Expansion RAM Capacity

Part Number	Internal Expansion RAM
μPD78F0887	2048 × 8 bits (F000H to F7FFH)
μPD78F0888	
μPD78F0889	4096 × 8 bits (E800H to F7FFH)
μPD78F0890	6144 × 8 bits (E000H to F7FFH)

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

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

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Table 3-8 Special Function Register List** in **3.2.3 Special Function Registers (SFRs)**).

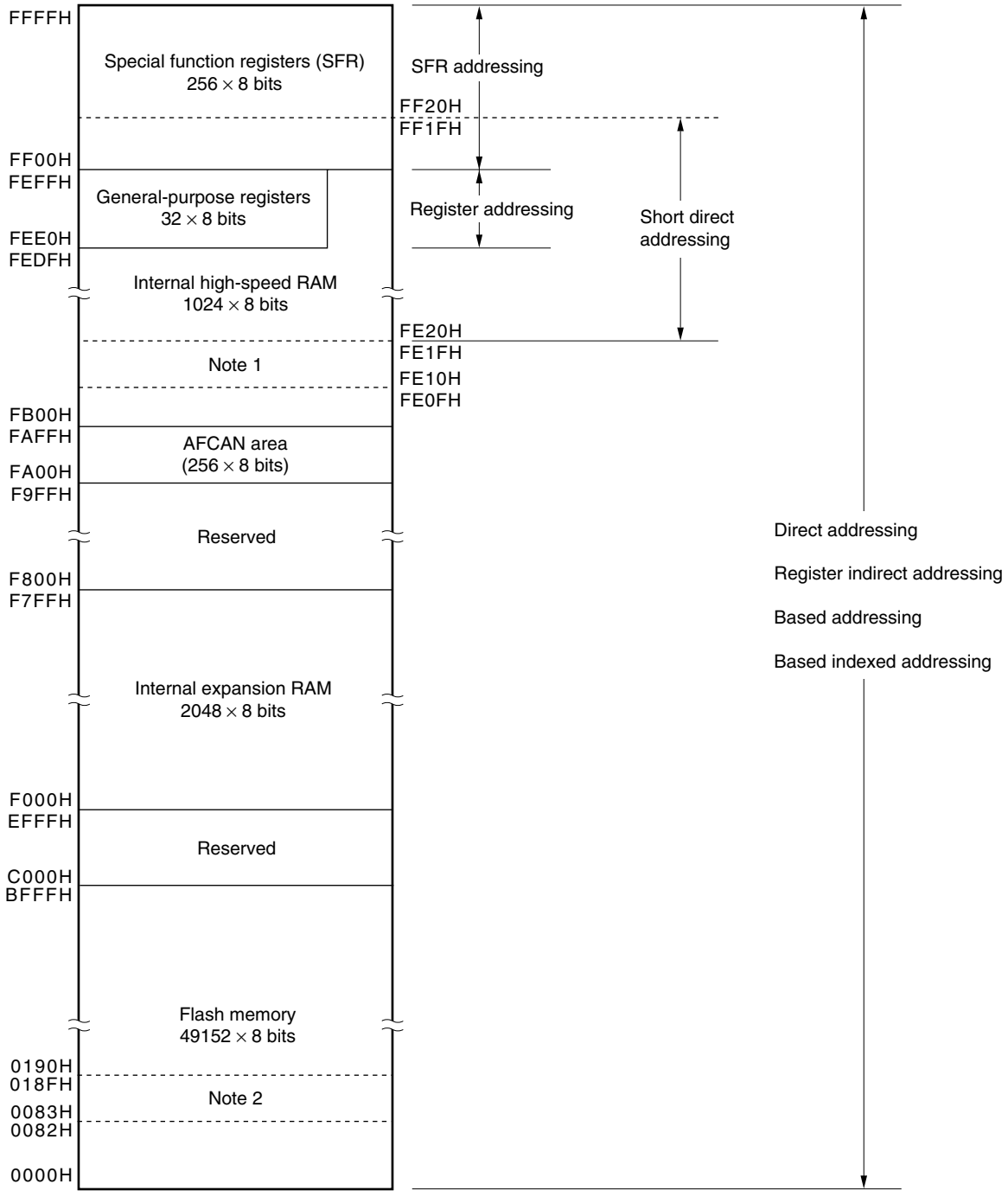
Caution Do not access addresses to which SFRs are not assigned.

3.1.5 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/FE2, 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. Figure 3-6 to 3-9 show correspondence between data memory and addressing. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

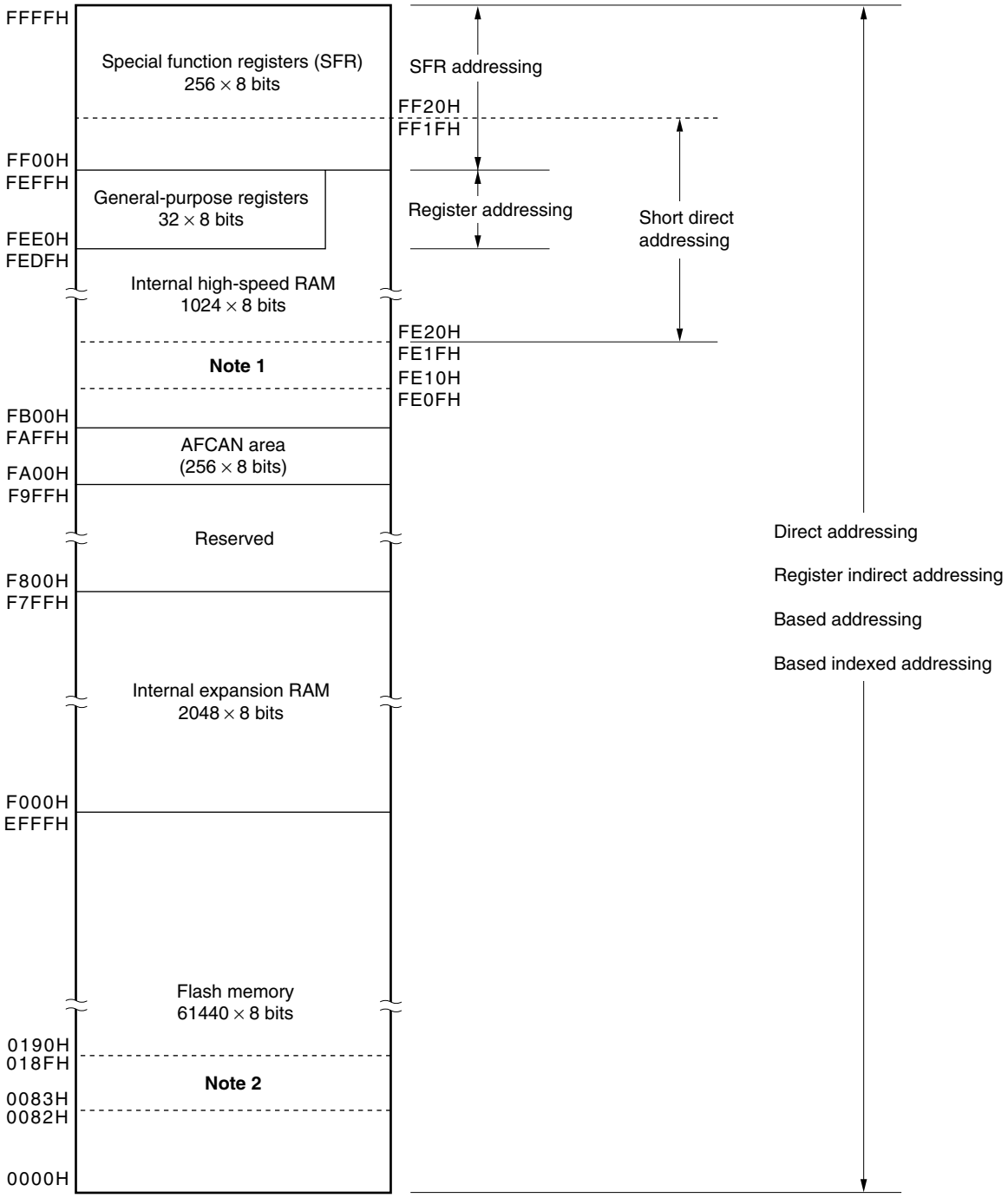
Figure 3-6. Correspondence Between Data Memory and Addressing (μPD78F0887)



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

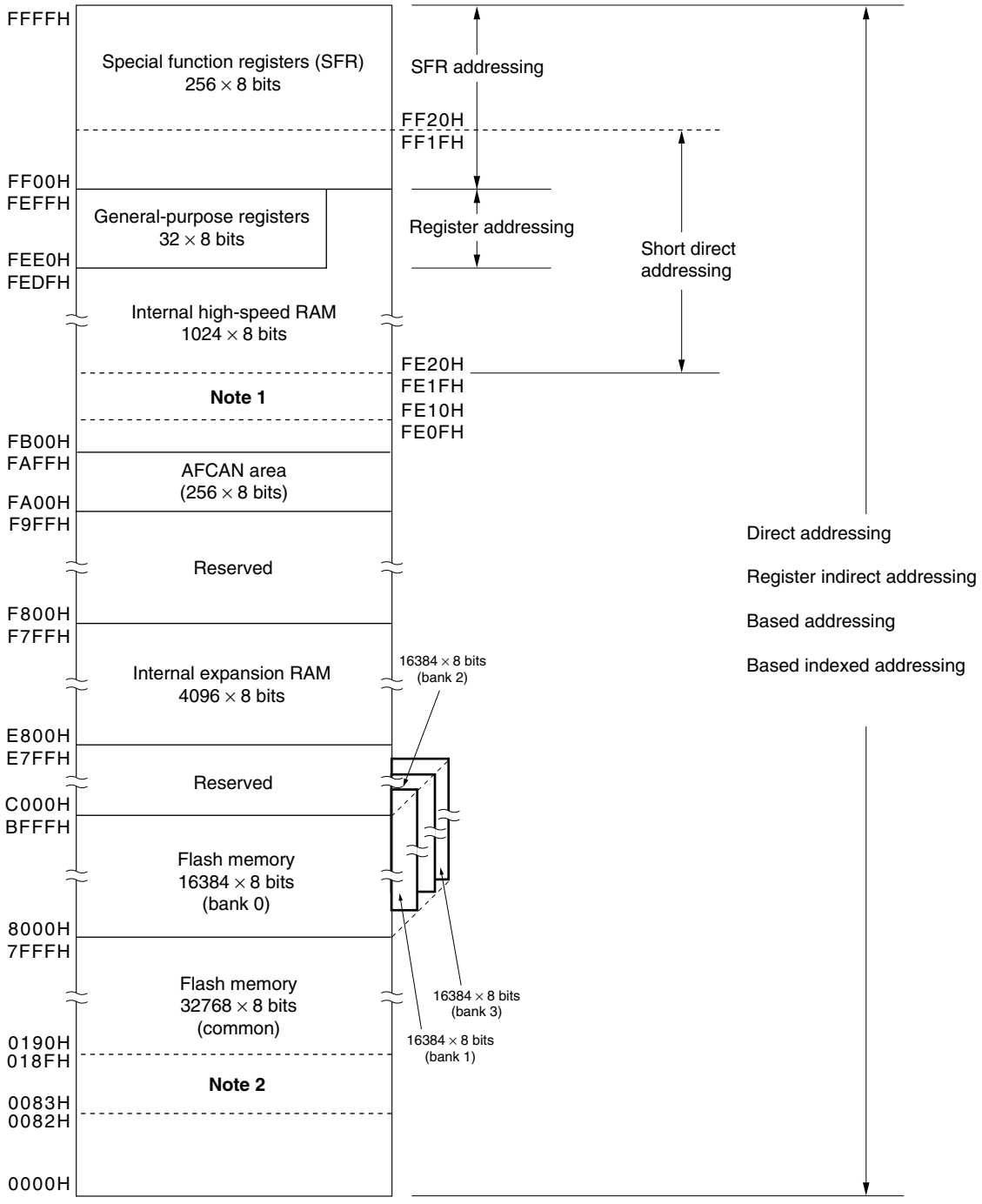
Figure 3-7. Correspondence Between Data Memory and Addressing (μPD78F0888)



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

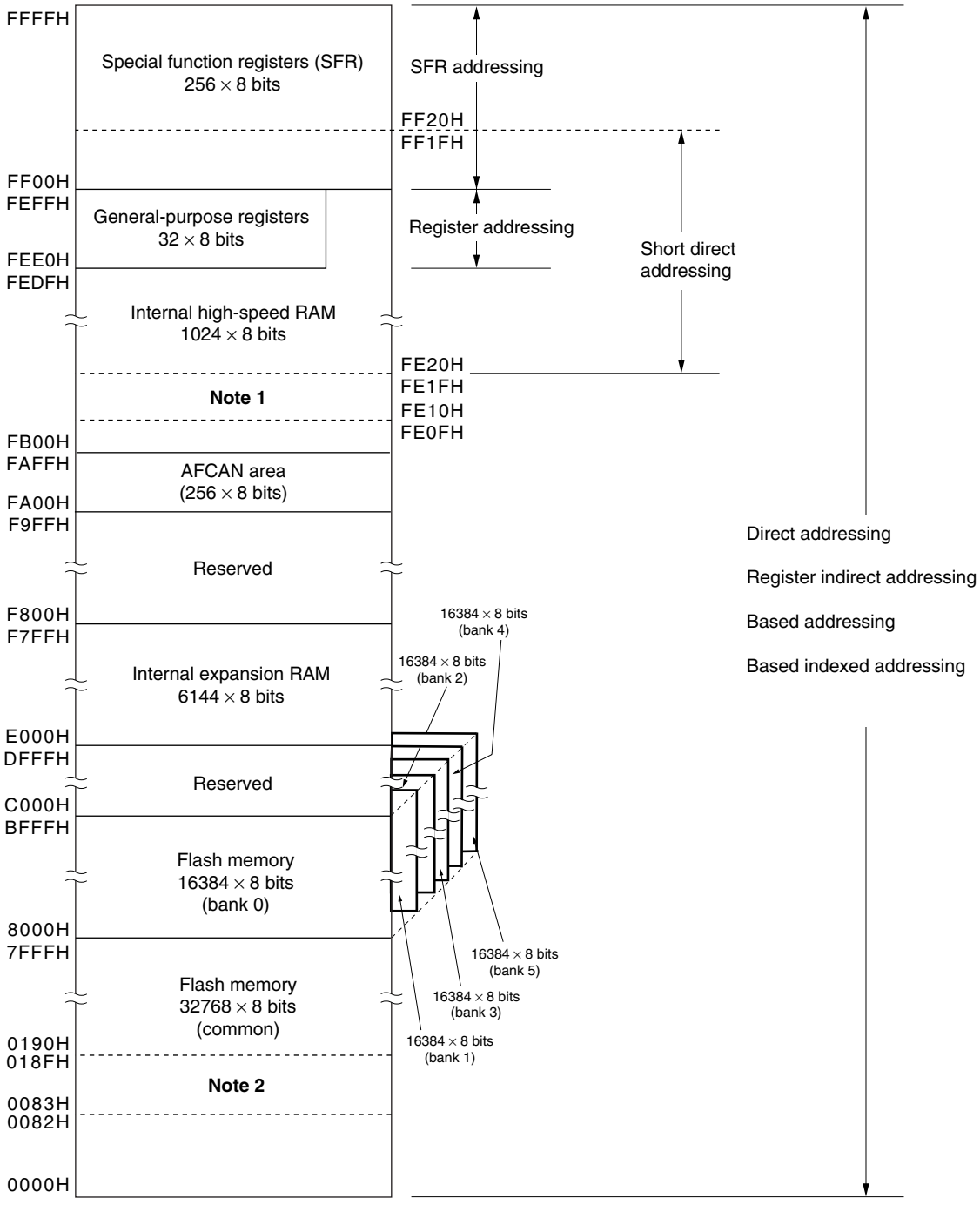
Figure 3-8. Correspondence Between Data Memory and Addressing (μPD78F0889)



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

Figure 3-9. Correspondence Between Data Memory and Addressing (μPD78F0890)



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3.2 Processor Registers

78K0/FE2 products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 16-bit register that holds the address information of the next program to be executed.

In normal operation, the 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 code values at addresses 0000H and 0001H to the program counter.

Figure 3-10. 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 automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets the PSW to 02H.

Figure 3-11. 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. Other interrupt requests are all disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgement 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 acknowledgement 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 acknowledgement 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 3-12 Format of Stack Pointer

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

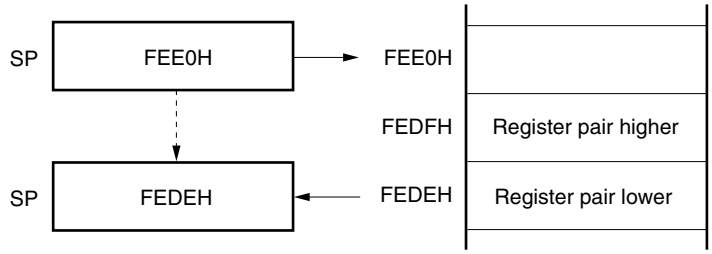
The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-13 and 3-14.

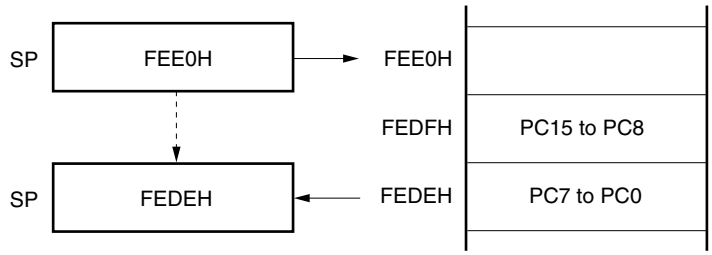
Caution Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-13. 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)

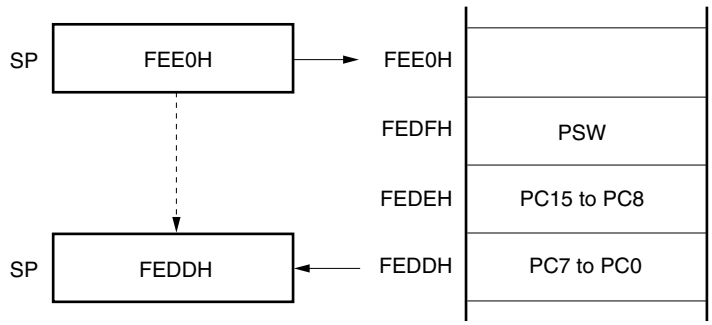
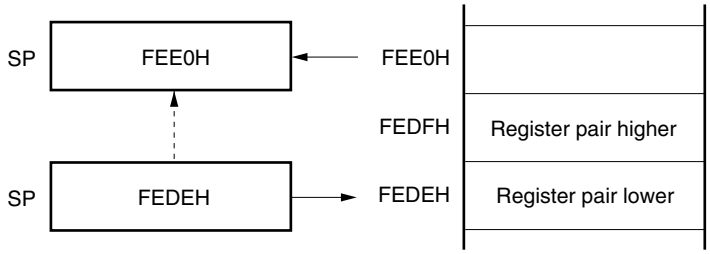
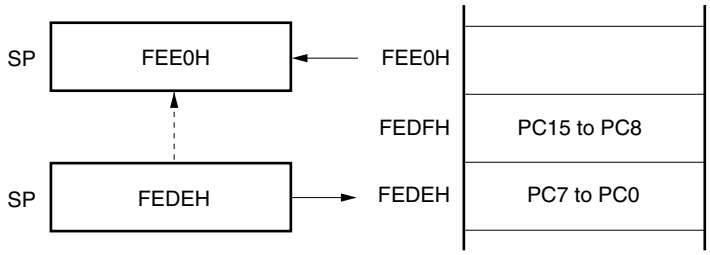


Figure 3-14. Data to Be Restored from Stack Memory

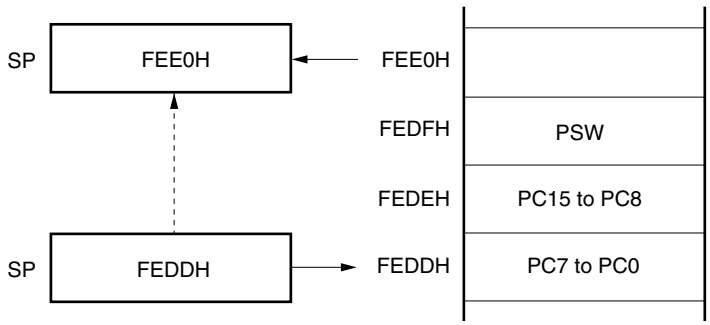
(a) POP rp instruction (when SP = FEDEH)



(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).

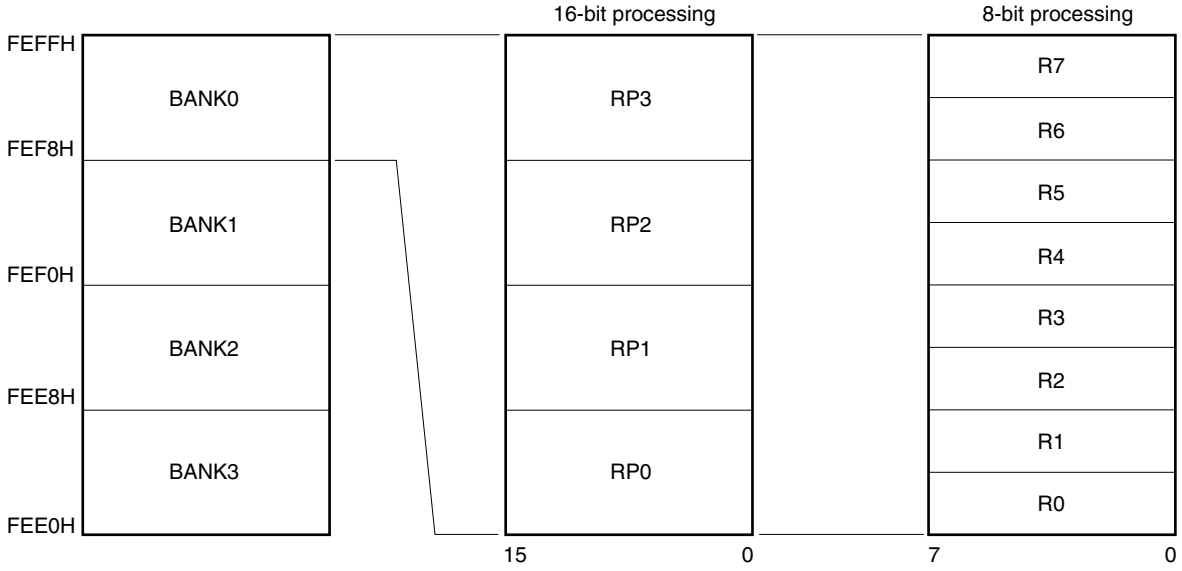
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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).

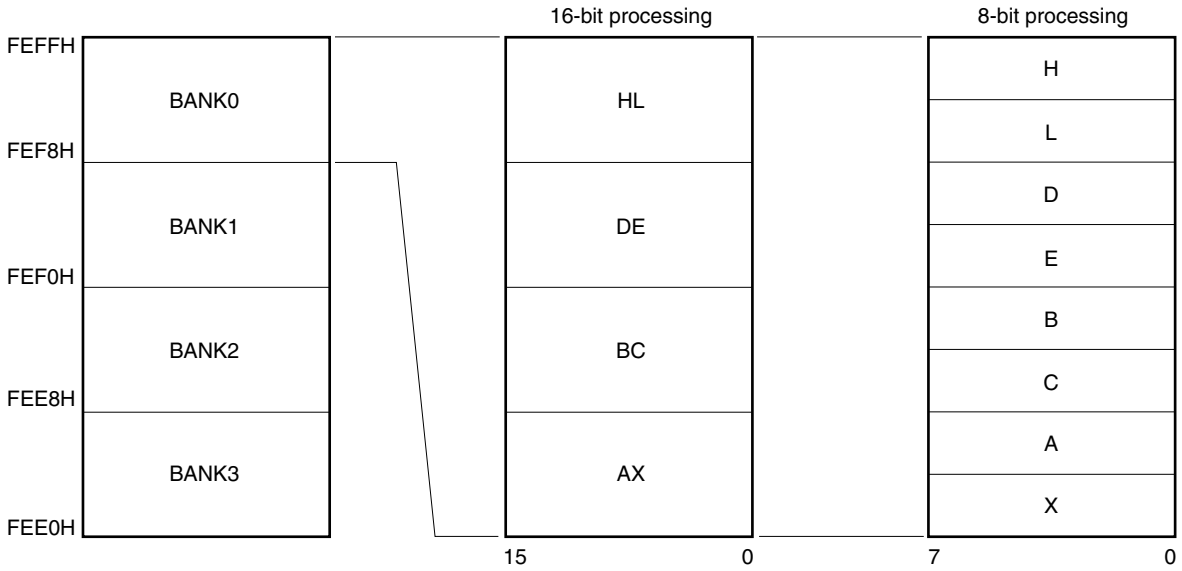
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-15. Configuration of General-Purpose Registers

(a) Absolute name



(b) Function 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 3-8 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 by the header file “sfrbit.h” in the CC78K0. When using the RA78K0, ID78K0-NS, ID78K0, or SM78K0, 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 3-8. Special Function Register List (1/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port register 0	P0	R/W	√	√	–	00H
FF01H	Port register 1	P1	R/W	√	√	–	00H
FF02H	8-bit timer H compare register 00	CMP00	R/W	–	√	–	00H
FF03H	Port register 3	P3	R/W	√	√	–	00H
FF04H	Port register 4	P4	R/W	√	√	–	00H
FF05H	Port register 5	P5	R/W	√	√	–	00H
FF06H	Port register 6	P6	R/W	√	√	–	00H
FF07H	Port register 7	P7	R/W	√	√	–	00H
FF08H	Port register 8	P8	R/W	√	√	–	00H
FF09H	Port register 9	P9	R/W	√	√	–	00H
FF0AH	Receive buffer register 60	RXB60	R	–	√	–	FFH
FF0BH	Transmit buffer register 60	TXB60	R/W	–	√	–	FFH
FF0CH	Port register 12	P12	R/W	√	√	–	00H
FF0DH	Port register 13	P13	R/W	√	√	–	00H
FF0EH	8-bit timer H compare register 10	CMP10	R/W	–	√	–	00H
FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H
FF10H	16-bit timer counter 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16-bit timer capture/compare register 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H
FF17H	8-bit timer compare register 50	CR50	R/W	√	√	–	00H
FF18H	10-bit A/D conversion result register	ADCR	R	–	–	√	0000H
FF19H	8-bit A/D conversion result register	ADCRH	R	–	√	–	00H
FF1AH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	00H
FF1BH	8-bit timer H compare register 11	CMP11	R/W	–	√	–	00H
FF1FH	8-bit timer counter 51	TM51	R	–	√	–	00H
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1	R/W	√	√	–	FFH
FF22H	A/D port configuration register	ADPC	R/W	√	√	–	00H
FF23H	Port mode register 3	PM3	R/W	√	√	–	FFH
FF24H	Port mode register 4	PM4	R/W	√	√	–	FFH
FF25H	Port mode register 5	PM5	R/W	√	√	–	FFH
FF26H	Port mode register 6	PM6	R/W	√	√	–	FFH
FF27H	Port mode register 7	PM7	R/W	√	√	–	FFH
FF28H	Port mode register 8	PM8	R/W	√	√	–	FFH
FF29H	Port mode register 9	PM9	R/W	√	√	–	FFH

Table 3-8. Special Function Register List (2/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF2AH	A/D converter mode register	ADM	R/W	√	√	–	00H
FF2BH	Analog input channel specification register	ADS	R/W	√	√	–	00H
FF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH
FF2DH	Port mode register 13	PM13	R/W	√	√	–	FEH
FF2EH	Asynchronous serial interface operation mode register 61	ASIM61	R/W	√	√	–	01H
FF2FH	Asynchronous serial interface reception error status register 61	ASIS61	R	–	√	–	00H
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
FF34H	Pull-up resistor option register 4	PU4	R/W	√	√	–	00H
FF35H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
FF37H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H
FF38H	Asynchronous serial interface transmission status register 61	ASIF61	R	–	√	–	00H
FF39H	Clock selection register 61	CKSR61	R/W	–	√	–	00H
FF3AH	Receive buffer register 61	RXB61	R/W	–	√	–	FFH
FF3BH	Transmit buffer register 61	TXB61	R/W	–	√	–	FFH
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H
FF3DH	Pull-up resistor option register 13	PU13	R/W	√	√	–	00H
FF3EH	Baud rate generator control register 61	BRGC61	R/W	–	√	–	FFH
FF3FH	Asynchronous serial interface control register 61	ASICL61	R/W	√	√	–	16H
FF40H	Clock output selection register	CKS	R/W	√	√	–	00H
FF41H	8-bit timer compare register 51	CR51	R/W	√	√	–	00H
FF42H	Multiplier/divider control register 0	DMUC0	R/W	√	√	–	00H
FF43H	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H
FF44H	Remainder data register 0	SDR0 SDR0L	R/W	–	√	√	0000H
FF45H		SDR0H					
FF47H	Serial I/O shift register 11	SIO11	R	–	√	–	00H
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H
FF4AH	Multiplication/Division Data Register A0L	MDA0L	R/W	–	√	√	0000H
FF4BH							
FF4CH	Multiplication/Division Data Register A0H	MDA0H	R/W	–	√	√	0000H
FF4DH							
FF4EH	Transmit buffer register 11	SOTB11	R/W	–	√	–	00H
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H
FF50H	Asynchronous serial interface operation mode register 60	ASIM60	R/W	√	√	–	01H

Table 3-8. Special Function Register List (3/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF51H	Prescaler mode register 03	PRM03	R/W	√	√	–	00H
FF52H	Capture/compare control register 03	CRC03	R/W	√	√	–	00H
FF53H	Asynchronous serial interface reception error status register 60	ASIS60	R	–	√	–	00H
FF54H	16-bit timer mode control register 02	TMC02	R/W	√	√	–	00H
FF55H	Asynchronous serial interface transmission status register 60	ASIF60	R	–	√	–	00H
FF56H	Clock selection register 60	CKSR60	R/W	–	√	–	00H
FF57H	Baud rate generator control register 60	BRGC60	R/W	–	√	–	FFH
FF58H	Asynchronous serial interface control register 60	ASICL60	R/W	√	√	–	16H
FF59H	Prescaler mode register 02	PRM02	R/W	√	√	–	00H
FF5AH	16-bit timer counter 02	TM02	R	–	–	√	0000H
FF5BH							
FF5CH	Capture/compare control register 02	CRC02	R/W	√	√	–	00H
FF60H	Module Receive History List Get Pointer Register	C0RGPT	R/W	–	–	√	xx02H
FF61H							
FF62H	Module Transmission History List Get Pointer Register	C0TGPT	R/W	–	–	√	xx02H
FF63H							
FF64H	CAN Global Macro Clock Selection	C0GMCTRL	R/W	–	–	√	0000H
FF65H							
FF66H	CAN Global Macro Automatic Block Transmission Delay Register	C0GMABT	R/W	–	–	√	0000H
FF67H							
FF68H	Module Last Out Pointer Register	C0LOPT	R	–	√	–	Undefined
FF69H	8-bit timer H mode register 0	TMHMD0	R/W	√	√	–	00H
FF6AH	Timer clock selection register 50	TCL50	R/W	√	√	–	00H
FF6BH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF6CH	16 bit capture/compare register 002	CR002	R/W	–	–	√	0000H
FF6DH							
FF6EH	CAN Global Macro Clock Selection Register	C0GMCS	R/W	–	√	–	0FH
FF6FH	CAN Global Macro Automatic Block Transmission Register	C0GMABTD	R/W	–	√	–	00H
FF70H	CAN Module Mask 1 Register L	C0MASK1L	R/W	–	–	√	Undefined
FF71H							
FF72H	CAN Module Mask 1 Register H	C0MASK1H	R/W	–	–	√	Undefined
FF73H							
FF74H	CAN Module Mask 2 Register L	C0MASK2L	R/W	–	–	√	Undefined
FF75H							
FF76H	CAN Module Mask 2 Register H	C0MASK2H	R/W	–	–	√	Undefined
FF77H							
FF78H	CAN Module Mask 3 Register L	C0MASK3L	R/W	–	–	√	Undefined
FF79H							

Table 3-8. Special Function Register List (4/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF7AH	CAN Module Mask 3 Register H	C0MASK3H	R/W	–	–	√	Undefined
FF7BH							
FF7CH	CAN Module Mask 4 Register L	C0MASK4L	R/W	–	–	√	Undefined
FF7DH							
FF7EH	CAN Module Mask 4 Register H	C0MASK4H	R/W	–	–	√	Undefined
FF7FH							
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	00H
FF88H	Serial operation mode register 11	CSIM11	R/W	√	√	–	00H
FF89H	Serial clock selection register 11	CSIC11	R/W	√	√	–	00H
FF8AH	CAN module time stamp register	C0TS	R/W	–	–	√	0000H
FF8BH							
FF8CH	Timer clock selection register 51	TCL51	R/W	√	√	–	00H
FF8FH	Watch timer operation mode register	WTM	R/W	√	√	–	00H
FF90H	CAN Module Control Register	C0CTRL	R/W	–	–	√	0000H
FF91H							
FF92H	CAN Module Last Error Code Register	C0LEC	R/W	–	√	–	00H
FF93H	CAN Module Information Register	C0INFO	R	–	√	–	00H
FF94H	CAN Module Error Counters	C0ERC	R	–	–	√	0000H
FF95H							
FF96H	CAN Module Interrupt Enable Register	C0IE	R/W	–	–	√	0000H
FF97H							
FF98H	CAN Module Interrupt Pending Register	C0INTS	R/W	–	–	√	0000H
FF99H							
FF9BH	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/9AH[Note1]
FF9CH	CAN Module Bit Rate Register	C0BTR	R/W	–	–	√	370FH
FF9DH							
FF9EH	CAN Module bit rate Prescaler register	C0BRP	R/W	–	√	–	FFH
FF9FH	CAN Module Last In Pointer Register	C0LIPT	R	–	√	–	Undefined
FFA0H	Internal oscillator mode register	RCM	R/W	√	√	–	00H[Note2]
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	16-bit timer output control register 02	TOC02	R/W	√	√	–	00H
FFA6H	16-bit timer counter 03	TM03	R	–	–	√	0000H
FFA7H							

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 internal high-speed oscillator has been stabilized.

Tables 3-8. Special Function Register List (5/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFA8H	16-bit timer capture/compare register 003	CR003		R/W	–	–	√	0000H
FFA9H								
FFAAH	16-bit timer capture/compare register 013	CR013		R/W	–	–	√	0000H
FFABH								
FFACH	Reset control flag register	RESF		R	–	√	–	00H[Note 1]
FFADH	16-bit timer mode control register 03	TMC03		R/W	√	√	–	00H
FFAEH	Multiplier/divider data register B0	MDB0	MDB0L	R/W	–	√	√	0000H
FFAFH			MDB0H					
FFB0H	16-bit timer counter 01	TM01		R	–	–	√	0000H
FFB1H								
FFB2H	16-bit timer capture/compare register 001	CR001		R/W	–	–	√	0000H
FFB3H								
FFB4H	16-bit timer capture/compare register 011	CR011		R/W	–	–	√	0000H
FFB5H								
FFB6H	16-bit timer mode control register 01	TMC01		R/W	√	√	–	00H
FFB7H	Prescaler mode register 01	PRM01		R/W	√	√	–	00H
FFB8H	Capture/compare control register 01	CRC01		R/W	√	√	–	00H
FFB9H	16-bit timer output control register 01	TOC01		R/W	√	√	–	00H
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
FFBFH	Low-voltage detection level selection register	LVIS		R/W	√	√	–	00H
FFC2H	Flash status register	PFS		R/W	√	√	–	00H
FFC4H	Flash programming mode control register	FLPMC		R/W	√	√	–	08H/0CH[Note 2]
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	√	√		DFH
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH

Notes 1. This value varies depending on the reset source.

2. Varies depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

Tables 3-8. Special Function Register List (6/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFEAH	Priority specification flag register 1L	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	Priority specification flag register 1H		PR1H	R/W	√	√		FFH
FFECH	16-bit timer capture/compare register 012	CR012		R/W	–	–	√	0000H
FFEDH								
FFEEH	8-bit timer H carrier control register 1	TMCYC1		R/W	√	√	–	00H
FFEFH	Clock operation mode select register	OSCCTL		R/W	√	√	–	00H
FFF0H	Internal memory size switching register[Note]	IMS		R/W	–	√	–	CFH
FFF4H	Internal expansion RAM size switching register[Note]	IXS		R/W	–	√	–	0CH
FFF9H	16-bit timer output control register 03	TOC03		R/W	√	√	–	00H
FFFAH	8-bit timer H mode register 1	TMHMD1		R/W	√	√	–	00H
FFFBH	Processor clock control register	PCC		R/W	√	√	–	01H

Note Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of the 78K0/FE2 is fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each as indicated below.

Flash Memory Version	IMS	IXS
μPD78F0887	CCH	08H
μPD78F0888	CFH	08H
μPD78F0889	CCH	04H
μPD78F0890	CCH	00H

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents 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 the PC and branched by the following addressing (for details of instructions, refer to **78K/0 Series Instructions User's Manual (U12326E)**.

3.3.1 Relative addressing

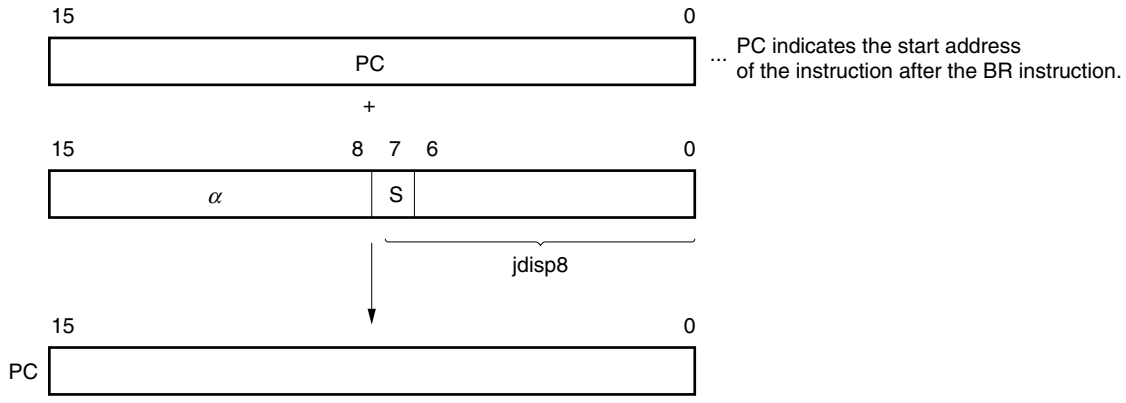
[Function]

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

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

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

[Illustration]



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

3.3.2 Immediate addressing

[Function]

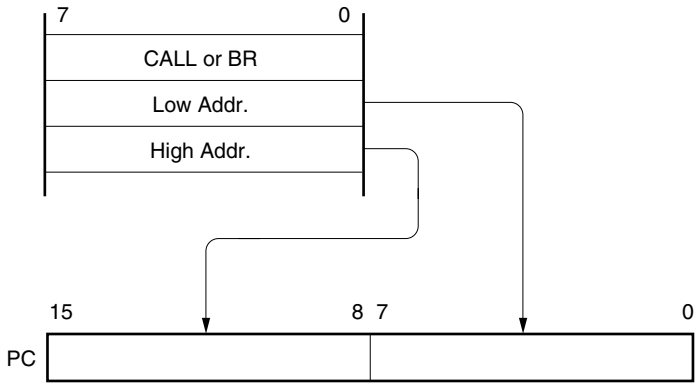
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

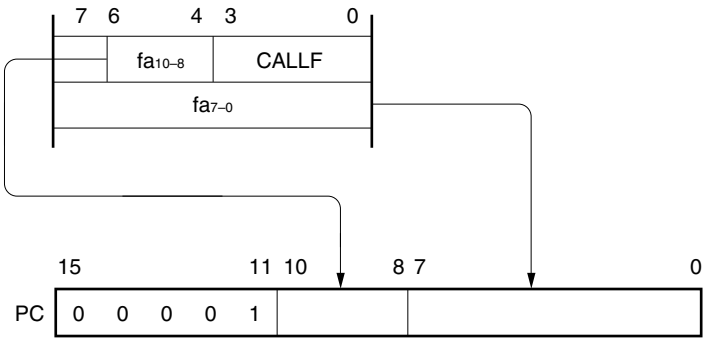
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

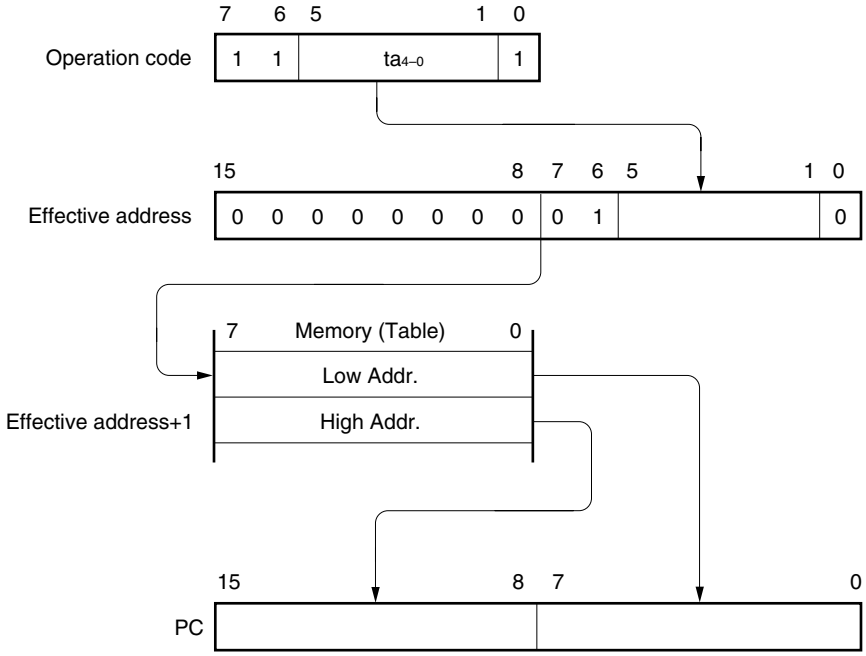
[Function]

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

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

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

[Illustration]



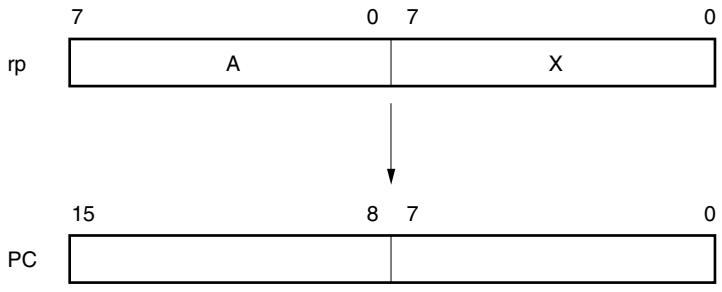
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

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

Of the 78K0/FE2 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 employed 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 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 (Rn and RPn) of an operation code.

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

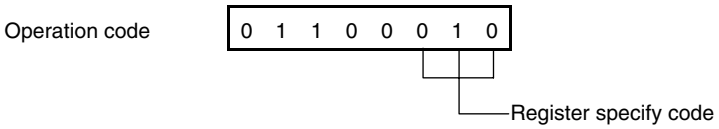
[Operand format]

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

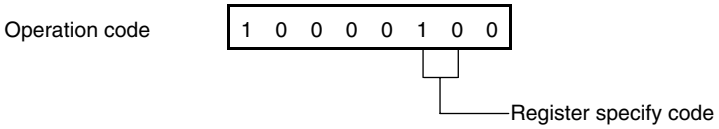
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

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

[Operand format]

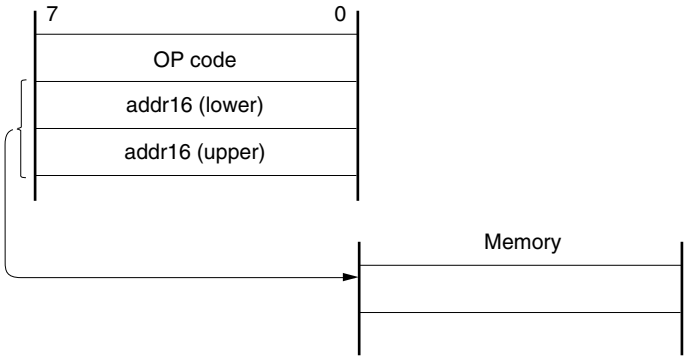
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

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

This addressing is applied to the 256-byte space FE20H to FF1FH. Internal 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.

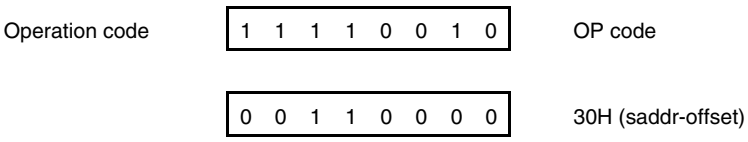
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to the **[Illustration]** shown below.

[Operand format]

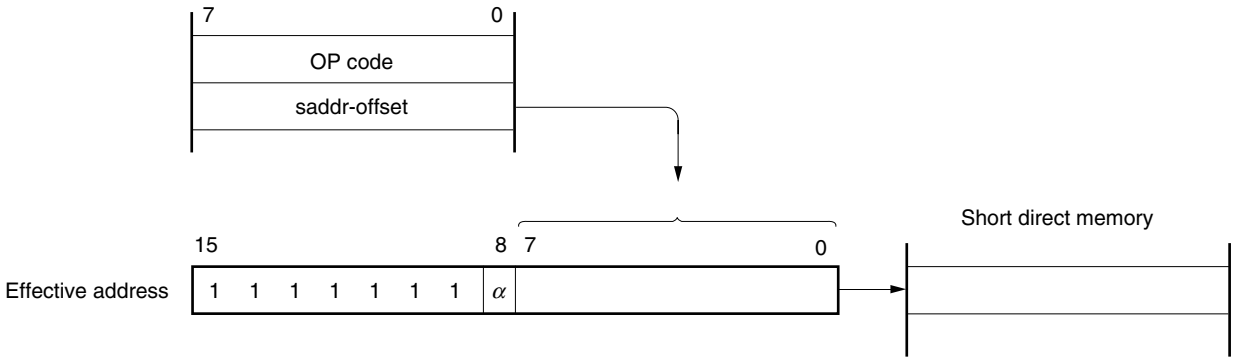
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

MOV 0FE30H, A; when transferring value of A register to saddr (FE30H)



[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]

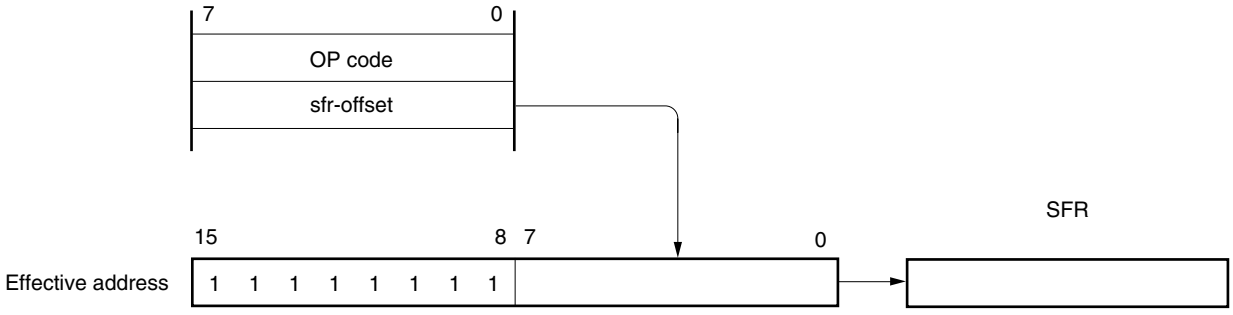
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

Operation code	1 1 1 1 0 1 1 0	OP code
	0 0 1 0 0 0 0 0	20H (sfr-offset)

[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 the memory spaces.

[Operand format]

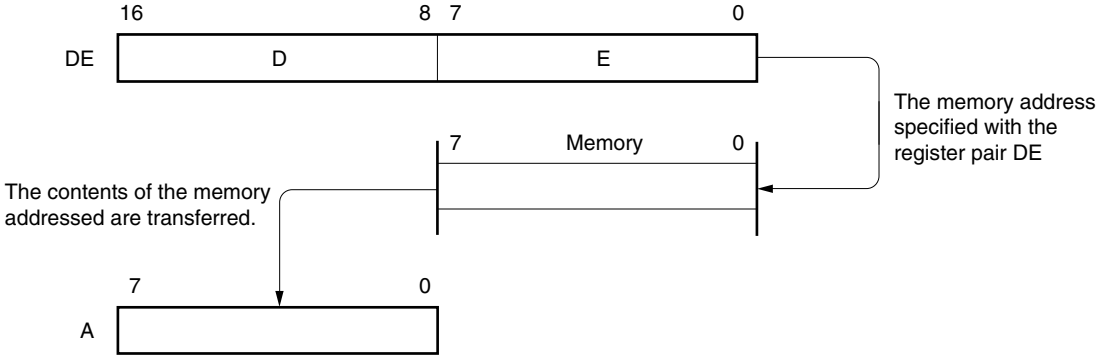
Identifier	Description
–	[DE], [HL]

[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 the memory spaces.

[Operand format]

Identifier	Description
–	[HL + byte]

[Description example]

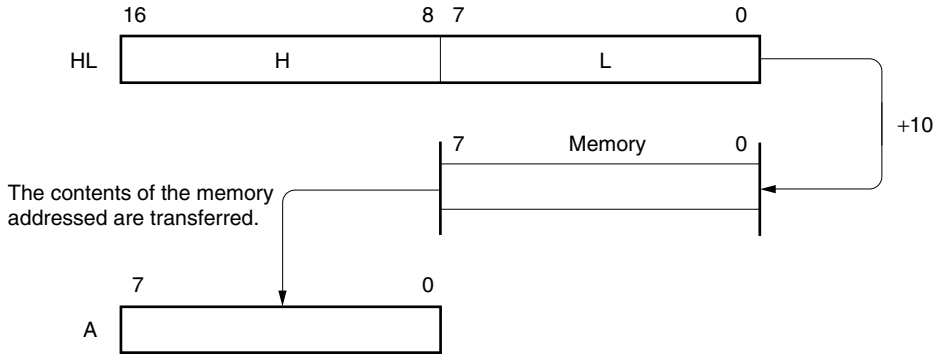
MOV A, [HL + 10H]; when setting byte to 10H

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 the memory spaces.

[Operand format]

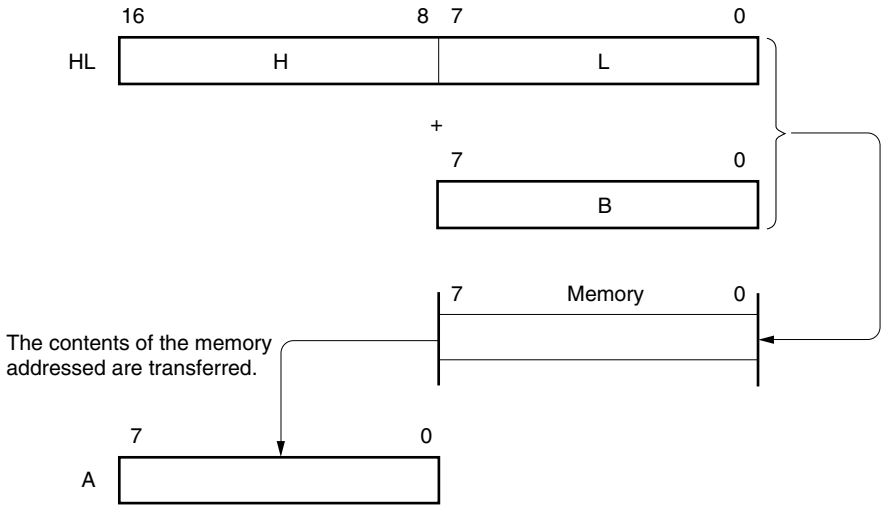
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

In the case of MOV A, [HL + B]; (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.

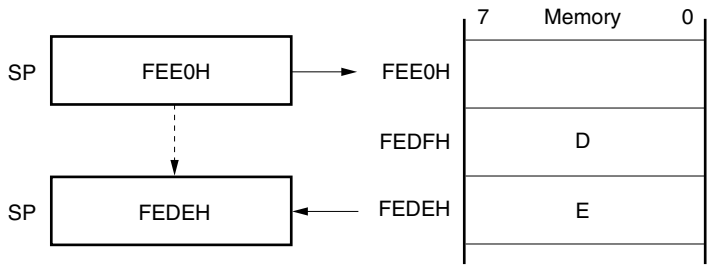
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

In the case of PUSH DE; (saving DE register)

Operation code	1	0	1	1	0	1	0	1

[Illustration]



CHAPTER 4 MEMORY BANK SELECT FUNCTION (μPD78F0889, 78F0890 ONLY)

4.1 Memory Bank

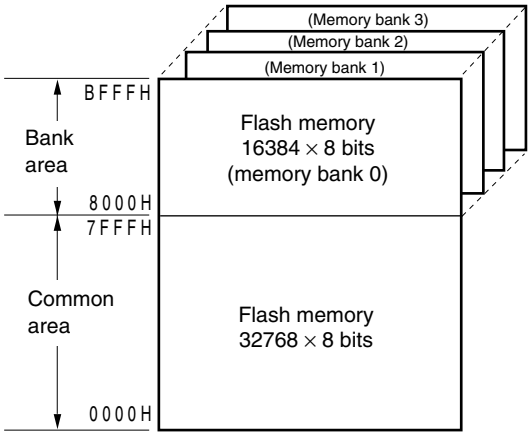
The μPD78F0889, 78F0890 implement a ROM capacity of 96 KB or 128 KB by selecting a memory bank from a memory space of 8000H to BFFFH.

The μPD78F0889 has memory banks 0 to 3, and the μPD78F0890 have memory banks 0 to 5, as shown below.

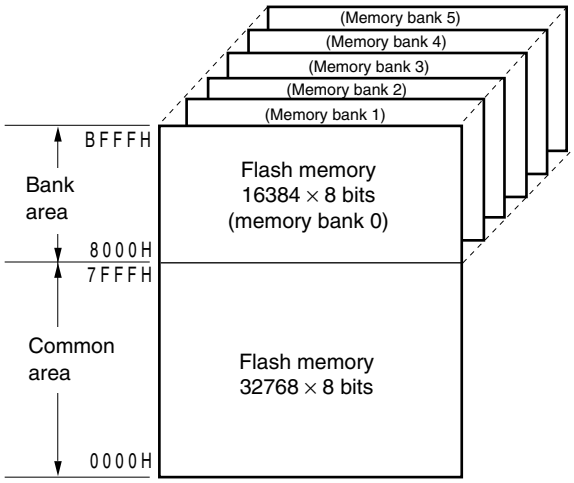
The memory banks are selected by using a memory bank select register (BANK).

Figure 4-1. Internal ROM (Flash Memory) Configuration

(a) μPD78F0889



(b) μPD78F0890



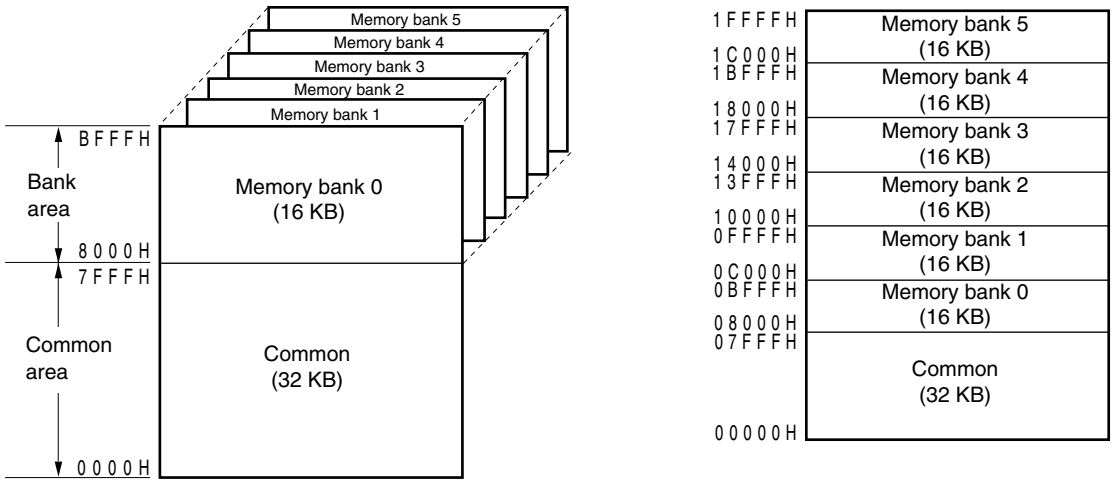
4.2 Difference in Representation of Memory Space

With the 78K0/FE2 products which support the memory bank, addresses can be viewed in the following two different ways.

- Memory bank number + CPU address
- Flash memory real address (HEX FORMAT [BANK])

Figure 4-2. Address View

(a) Memory bank number + CPU address **(b) Flash memory real address (HEX FORMAT [BANK])**



"Memory bank number + CPU address" is represented with a vacancy in the address space, while the flash memory real address is shown with no vacancy in the address space.

"Memory bank number + CPU address" is used for addressing in the user program. For on-board programming and self programming not using the self programming sample library[Note 1], the flash memory real address is used.

Note that the HEX file that is output by the assembler (RA78K0) by default uses the flash memory real address. For address representation of the other tools such as the simulator and the debugger[Note 2], see **Table 4-1**.

Notes 1. "Memory bank number + CPU address" can be used when performing self programming, using the self programming sample library, because the addresses are automatically translated.

2. SM+ for 78K0/Fx2, ID78K0-QB

Table 4-1. Memory Bank Address Representation

Memory Bank Number	CPU Address	Flash Memory Real Address	Address Representation in Simulator and Debugger[Note 1]
Memory bank 0	08000H-0BFFFH[Note 2]	08000H-0BFFFH	08000H-0BFFFH
Memory bank 1		0C000H-0FFFFH	18000H-1BFFFH
Memory bank 2		10000H-13FFFH	28000H-2BFFFH
Memory bank 3		14000H-17FFFH	38000H-3BFFFH
Memory bank 4		18000H-1BFFFH	48000H-4BFFFH
Memory bank 5		1C000H-1FFFFH	58000H-5BFFFH

Notes 1. SM+ for 78K0/Fx2, ID78K0-QB

2. Set the memory bank to be used by the memory bank select register (BANK) (see **Figure 4-3**).

For details, see the **RA78K0 Ver. 3.80 Assembler Package Operation User's Manual (U17199E)**.

4.3 Memory Bank Select Register (BANK)

The memory bank select register (BANK) is used to select a memory bank to be used.
BANK can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears BANK to 00H.

Figure 4-3. Format of Memory Bank Select Register (BANK)

Address: FFF3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BANK	0	0	0	0	0	BANK2	BANK1	BANK0

BANK2	BANK1	BANK0	Bank setting	
			μPD78F0889	μPD78F0890
0	0	0	Common area (32 K) + memory bank 0 (16 K)	
0	0	1	Common area (32 K) + memory bank 1 (16 K)	
0	1	0	Common area (32 K) + memory bank 2 (16 K)	
0	1	1	Common area (32 K) + memory bank 3 (16 K)	
1	0	0	Setting prohibited	Common area (32 K) + memory bank 4 (16 K)
1	0	1		Common area (32 K) + memory bank 5 (16 K)
Other than above			Setting prohibited	

Caution Be sure to change the value of the BANK register in the common area (0000H to 7FFFH). If the value of the BANK register is changed in the bank area (8000H to BFFFH), an inadvertent program loop occurs in the CPU. Therefore, never change the value of the BANK register in the bank area.

4.4 Selecting Memory Bank

The memory bank selected by the memory bank select register (BANK) is reflected on the bank area and can be addressed. Therefore, to access a memory bank different from the one currently selected, that memory bank must be selected by using the BANK register.

The value of the BANK register must not be changed in the bank area (8000H to BFFFH). Therefore, to change the memory bank, branch an instruction to the common area (0000H to 7FFFH) and change the value of the BANK register in that area.

Cautions 1. Instructions cannot be fetched between different memory banks.

2. Branching and accessing cannot be directly executed between different memory banks. Execute branching or accessing between different memory banks via the common area.

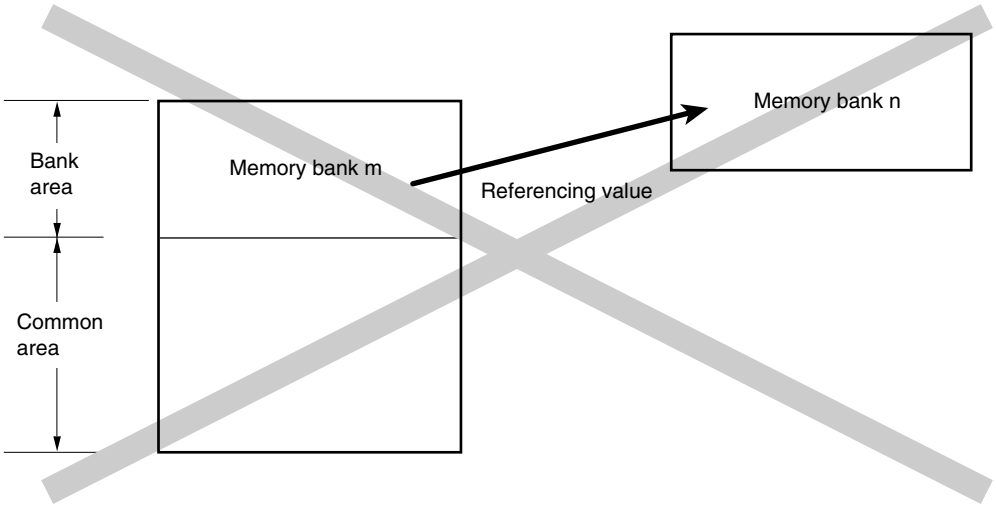
3. Allocate interrupt servicing in the common area.

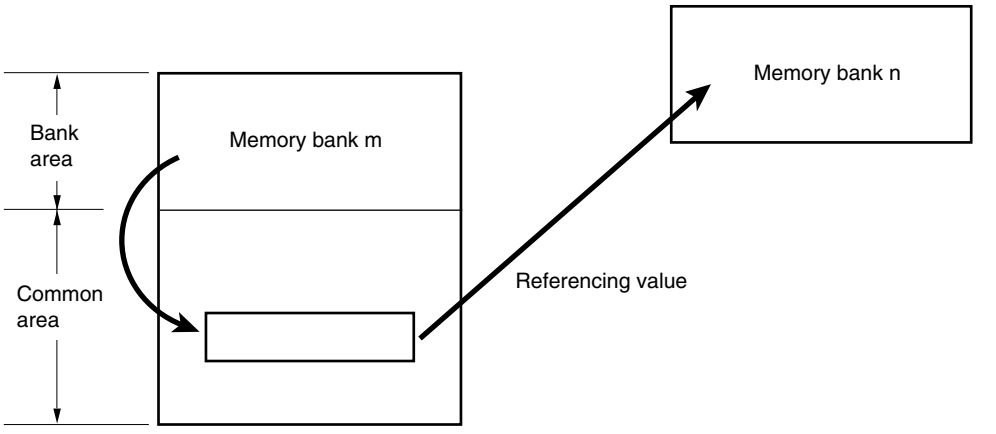
4. An instruction that extends from 7FFFH to 8000H can only be executed in memory bank 0.

4.4.1 Referencing values between memory banks

Values cannot be directly referenced from one memory bank to another.

To access another memory bank from one memory bank, branch once to the common area (0000H to 7FFFH), change the setting of the BANK register there, and then reference a value.





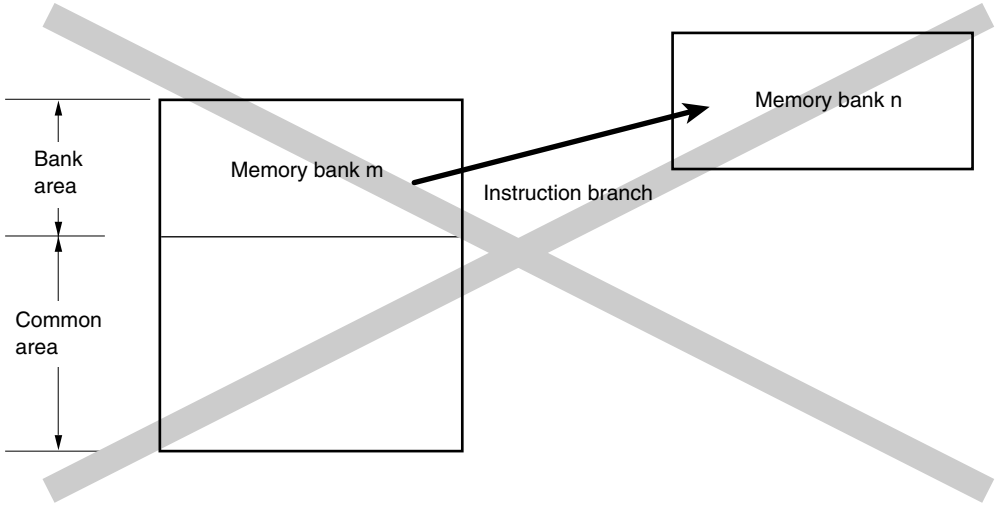
- Software example (to store a value to be referenced in register A)

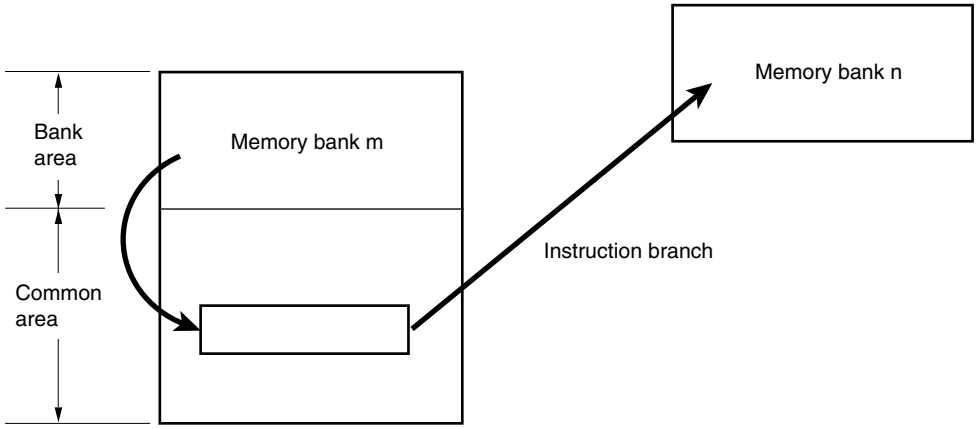
```
RAMD      DSEG    SADDR
R_BNKA:   DS      2                                ; Secures RAM for specifying an address at the reference destination.
R_BNKN:   DS      1                                ; Secures RAM for specifying a memory bank number at the reference destination.
R_BNKRN:  DS      1                                ; Secures RAM for saving a memory bank number at the reference source.

---------------------------------------------------------------------------------------------------------------------------
ETRC      CSEG    UNIT
ENTRY:

          MOV     R_BNKN,#BANKNUM  DATA1           ; Stores the memory bank number at the reference destination.
          MOVW    R_BNKA,#DATA1                    ; Stores the address at the reference destination.
          CALL    !BNKRD                           ; Calls a subroutine for referencing between memory banks.
                    :
                    :
---------------------------------------------------------------------------------------------------------------------------
BNKC      CSEG    AT       7000H

BNKRD:                                             ; Subroutine for referencing between memory banks.
          PUSH    HL                               ; Saves the contents of the HL register.
          MOV     A,R_BNKN                         ; Acquires the memory bank number at the reference destination.
          XCH     A,BANK                           ; Swaps the memory bank number at the reference source for that at the reference
                                                   ; destination
          MOV     R_BNKRN,A                        ; Saves the memory bank number at the reference source.
          XCHW    AX,HL                            ; Saves the contents of the X register.
          MOVW    AX,R_BNKA                        ; Acquires the address at the reference destination.
          XCHW    AX,HL                            ; Specifies the address at the reference destination.
          MOV     A,[HL]                           ; Reads the target value.
          XCH     A,R_BNKRN                        ; Acquires the memory bank number at the reference source.
          MOV     BANK,A                           ; Specifies the memory bank number at the reference source.
          MOV     A,R_BNKRN                        ; Write the target value to the A register.
          POP     HL                               ; Restores the contents of the HL register.

          RET                                      ; Return
---------------------------------------------------------------------------------------------------------------------------
DATA      CSEG    BANK3
DATA1:    DB      0AAH

END
```

4.4.2 Branching instruction between memory banks

Instructions cannot branch directly from one memory bank to another.

To branch an instruction from one memory bank to another, branch once to the common area (0000H to 7FFFH), change the setting of the BANK register there, and then execute the branch instruction again.





- Software example 1 (to branch from all areas)

```
RAMD     DSEG    SADDR
R_BNKA:  DS      2                          ; Secures RAM for specifying a memory bank at the branch destination.
R_BNKN:  DS      1                          ; Secures RAM for specifying a memory bank number at the branch destination.
RSAVEAX: DS      2                          ; Secures RAM for saving the AX register.

ETRC     CSEG    UNIT
ENTRY:
         MOV     R_BNKN,#BANKNUM  TEST      ; Stores the memory bank number at the branch destination in RAM.
         MOVW    R_BNKA,#TEST               ; Stores the address at the branch destination in RAM.
         BR      !BNKBR                     ; Branches to inter-memory bank branch processing.
           :
           :

BNKC     CSEG    AT       7000H             ;
BNKBR:
         MOVW    RSAVEAX,AX                 ; Saves the AX register.
         MOV     A,R_BNKN                   ; Acquires the memory bank number at the branch destination.
         MOV     BANK,A                     ; Specifies the memory bank number at the branch destination.
         MOVW    AX,R_BNKA                  ; Specifies the address at the branch destination.
         PUSH    AX                         ; Sets the address at the branch destination to stack.
         MOVW    AX,RSAVEAX                 ; Restores the AX register.
         RET                                ; Branch

BN3      CSEG    BANK3
TEST:
         MOV ···
           :
           :
END
```

- Software example 2 (to branch from common area to any bank area)

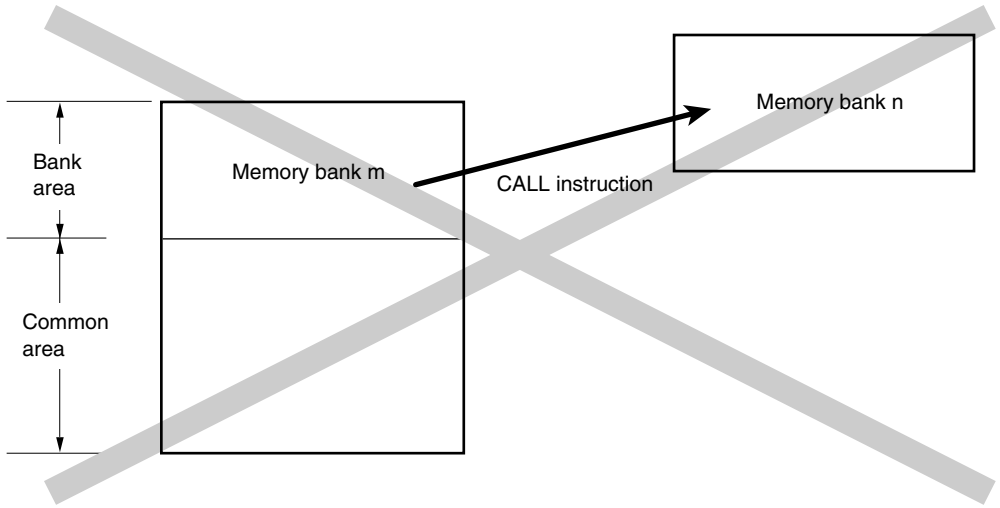
```
ETRC     CSEG    AT       2000H
ENTRY:
         MOV     R_BNKN,#BANKNUM TEST       ; Stores the memory bank number at the branch destination in RAM.
         BR      !TEST                      ; Stores the address at the branch destination in RAM.

BN3      CSEG    BANK3
TEST:
         MOV ···
           :
           :
END
```

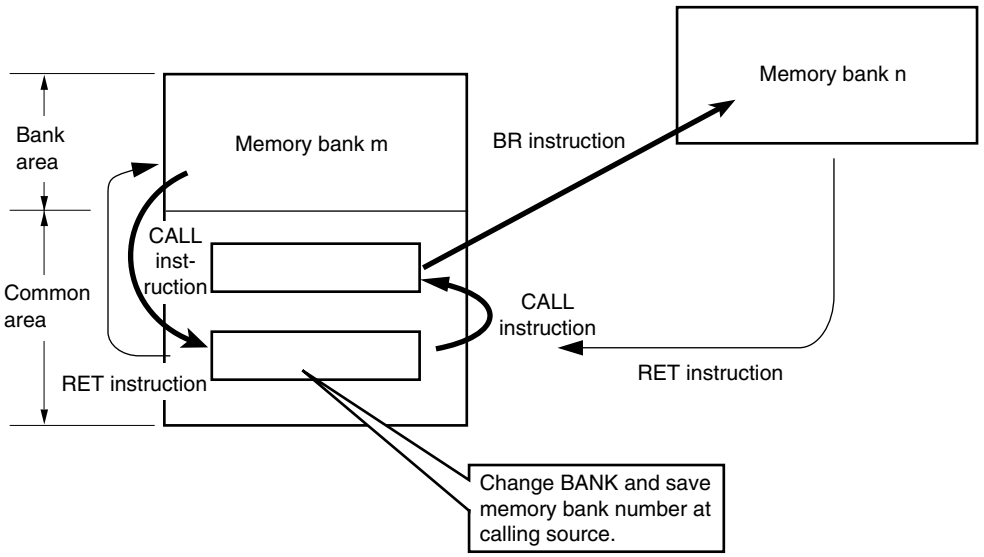
4.4.3 Subroutine call between memory banks

Subroutines cannot be directly called between memory banks.

To call a subroutine between memory banks, branch once to the common area (0000H to 7FFFH), specify the memory bank at the calling destination by using the BANK register there, execute the CALL instruction, and branch to the call destination by that instruction.

At this time, save the current value of the BANK register to RAM. Restore the value of the BANK register before executing the RET instruction.





- Software example

```
RAMD      DSEG    SADDR
R_BNKA:   DS      2                          ; Secures RAM for specifying an address at the calling destination.
R_BNKN:   DS      1                          ; Secures RAM for specifying a memory bank number at the calling destination.
R_BNKRN:  DS      1                          ; Secures RAM for saving a memory bank number at the calling source.
RSAVEAX:  DS      2                          ; Secures RAM for saving the AX register.
-------------------------------------------------------------------------------------------------------------------
ETRC      CSEG    UNIT
ENTRY:
          MOV     R_BNKN,#BANKNUM  TEST      ; Store the memory bank number at the calling destination in RAM.
          MOVW    R_BNKA,#TEST               ; Stores the address at the calling destination in RAM.
          CALL    !BNKCAL                    ; Branches to an inter-memory bank calling processing routine.
                    :
                    :
-------------------------------------------------------------------------------------------------------------------
BNKC      CSEG    AT       7000H
BNKCAL:                                      ; Inter-memory bank calling processing routine
          MOVW    RSAVEAX,AX                 ; Saves the AX register.
          MOV     A,R_BNKN                   ; Acquires the memory bank number at the calling destination.
          XCH     A,BANK                     ; Changes the bank and acquires the memory bank number at the calling source.
          MOV     R_BNKRN,A                  ; Saves the memory bank number at the calling source to RAM.
          CALL    !BNKCALS                   ; Calls a subroutine to branch to the calling destination.

          MOVW    RSAVEAX,AX                 ; Saves the AX register.
          XCH     A,R_BNKRN                  ; Acquires the memory bank number at the calling source.
          MOV     BANK,A                     ; Specifies the memory bank number at the calling source.
          MOVW    RSAVEAX,AX                 ; Restores the AX register.
          RET                                ; Returns to the calling source.

BNKCALS:
          MOVW    AX,R_BNKA                  ; Specifies the address at the calling destination.
          PUSH    AX                         ; Sets the address at the calling destination to stack.
          MOVW    AX,RSAVEAX                 ; Restores source AX register.
          RET     AX                         ; Branches to the calling destination.
-------------------------------------------------------------------------------------------------------------------
BN3       CSEG    BANK3
TEST:                                        ;
          MOV ···
            :
            :
          RET
END
```

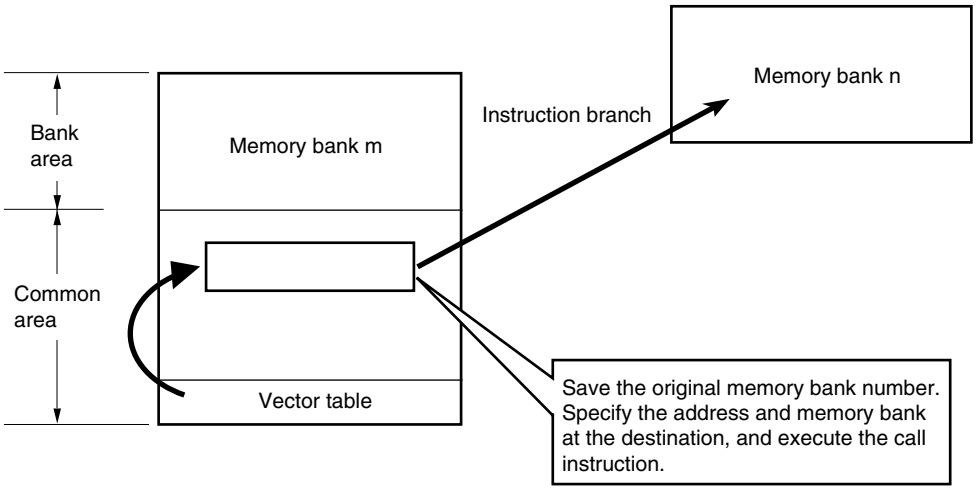
Remark In the software example above, multiplexed processing is not supported.

4.4.4 Instruction branch to bank area by interrupt

When an interrupt occurs, instructions can branch to the memory bank specified by the BANK register by using the vector table, but it is difficult to identify the BANK register when the interrupt occurs.

Therefore, specify the branch destination address specified by the vector table in the common area (0000H to 7FFFH), specify the memory bank at the branch destination by using the BANK register in the common area, and execute the CALL instruction. At this time, save the BANK register value before the change to RAM, and restore the value of the BANK register before executing the RETI instruction.

Remark Allocate interrupt servicing that requires a quick response in the common area.



- Software example (when using interrupt request of 16-bit timer/event counter 00)

```
VCTBL   CSEG    AT      0020H
        DW      BNKITM000               ; Specifies an address at the timer interrupt destination.

RAMD    DSEG    SADDR
R_BNKRN: DS     1                       ; Secures RAM for saving the memory bank number before the interrupt occurs.

-------------------------------------------------------------------------------------------------------

BNKC    CSEG    AT      7000H

BNKITM000:                              ; Inter-memory bank interrupt servicing routine
        PUSH    AX                      ; Saves the contents of the AX register.

        MOV     A,BANK
        MOV     R_BNKRN,A               ; Saves the memory bank number before the interrupt to RAM.
        MOV     BANK,#BANKNUM TEST      ; Specifies the memory bank number of the interrupt routine.
        CALL    !TEST                   ; Calls the interrupt routine.
        MOV     A,R_BNKRN               ; Restores the memory bank number before the interrupt.
        MOV     BANK,A

        POP     AX                      ; Restores the contents of the AX register.

        RETI

-------------------------------------------------------------------------------------------------------

BN3     CSEG    BANK3
TEST:                                   ; Interrupt servicing routine
        MOV ···
          :
          :
        RET
END
```

Remark Note the following points to use the memory bank select function efficiently.

- Allocate a routine that is used often in the common area.
- If a value that is planned to be referenced is placed in RAM, it can be referenced from all of the areas.
- If the reference destination and the branch destination of the routine placed in a memory bank are placed in the same memory bank, then the code size and processing are more efficient.
- Allocate interrupt servicing that requires a quick response in the common area.

CHAPTER 5 PORT FUNCTIONS

5.1 Port Functions

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

Table 5-1. Pin I/O Buffer Power Supplies

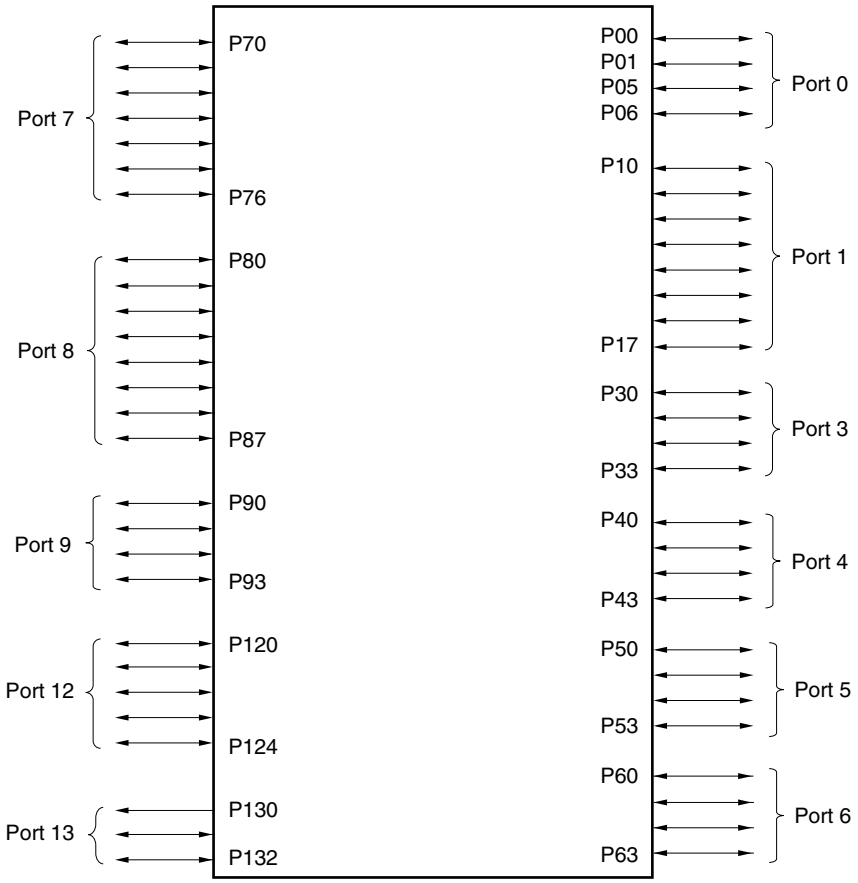
Power Supply	Corresponding Pins
AV_{REF}	P80 to P87, P90 to P93
EV_{DD}	Port pins other than P80 to P87, P90 to P93 and P121 to P124
V_{DD}	• P121 to P124 • Non-port pins

78K0/FE2 products are provided with the ports shown in Figure 5-1, which enable variety of control operations.

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

The 78K0/FE2 has a total of 55 I/O ports, ports 0, 1, 3 to 9, 12 and 13. The port configuration is shown below.

Figure 5-1. Port Types



5.2 Port Configuration

Ports include the following hardware.

Table 5-2. Port Configuration

Item	Configuration
Control registers	Port mode register (PM0, PM1, PM3 to PM9, PM12, PM13) Port register (P0, P1, P3 to P9, P12, P13) Pull-up resistor option register (PU0, PU1, PU3 to PU5, PU7, PU12, PU13)
Port	Total: 55 (CMOS I/O: 50, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	Total: 34

5.2.1 Port 0

Port 0 is a 4-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00, P01, P05 and P06 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

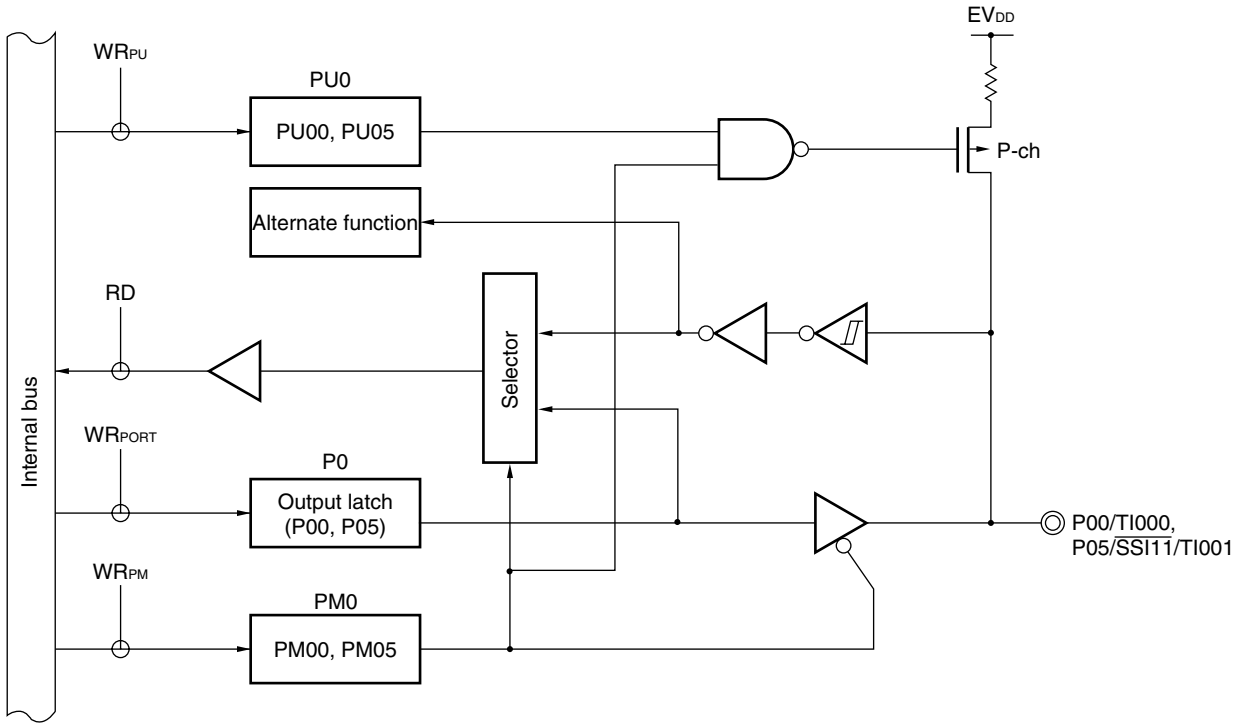
This port can also be used for timer I/O, serial interface chip select input.

Reset signal generation sets port 0 to input mode.

Figures 5-2 and 5-3 show block diagrams of port 0.

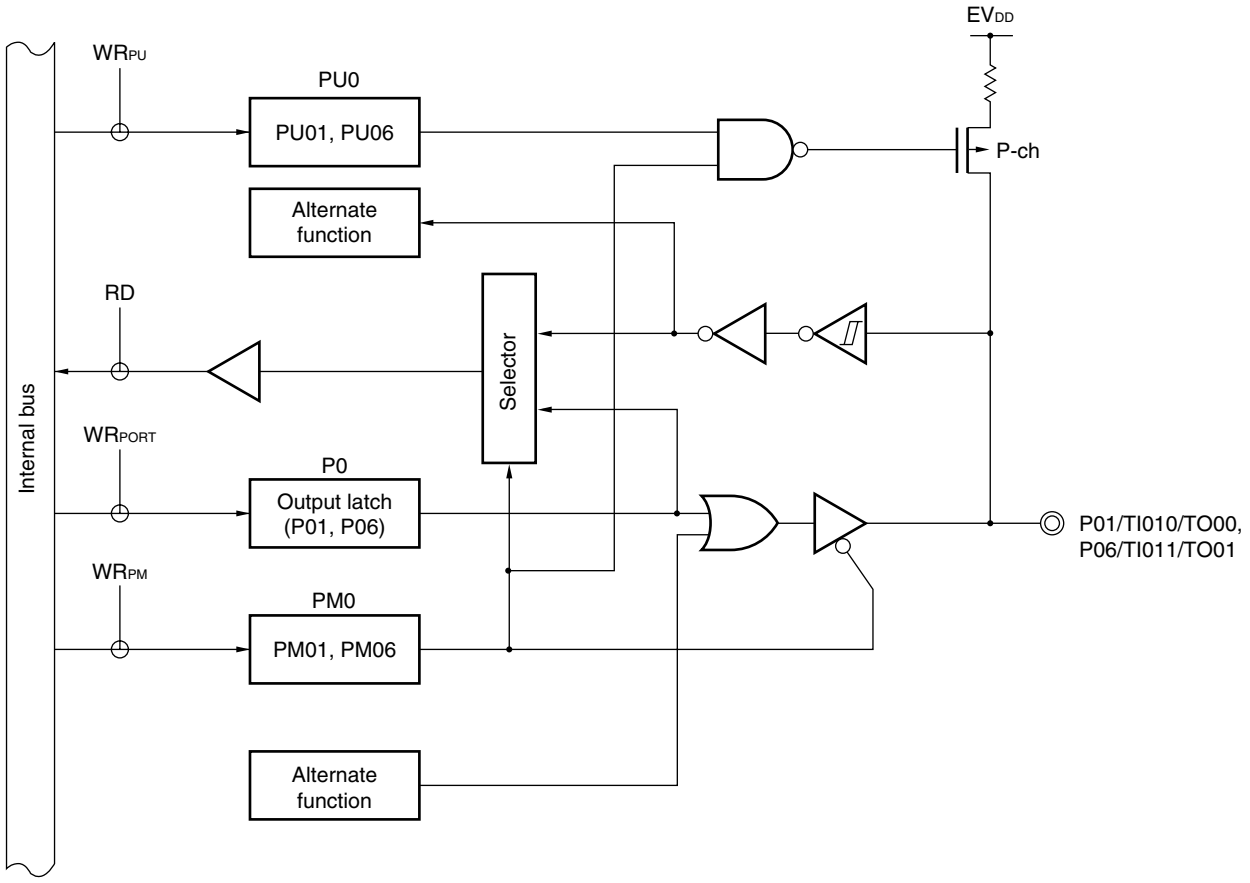
Caution To use P05/$\overline{\text{SSI11}}$/TI001 as general-purpose ports, set serial operation mode register 11 (CSIM11) to the default status (00H).

Figure 5-2. Block Diagram of P00 and P05



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

Figure 5-3. Block Diagram of P01 and P06



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

5.2.2 Port 1

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

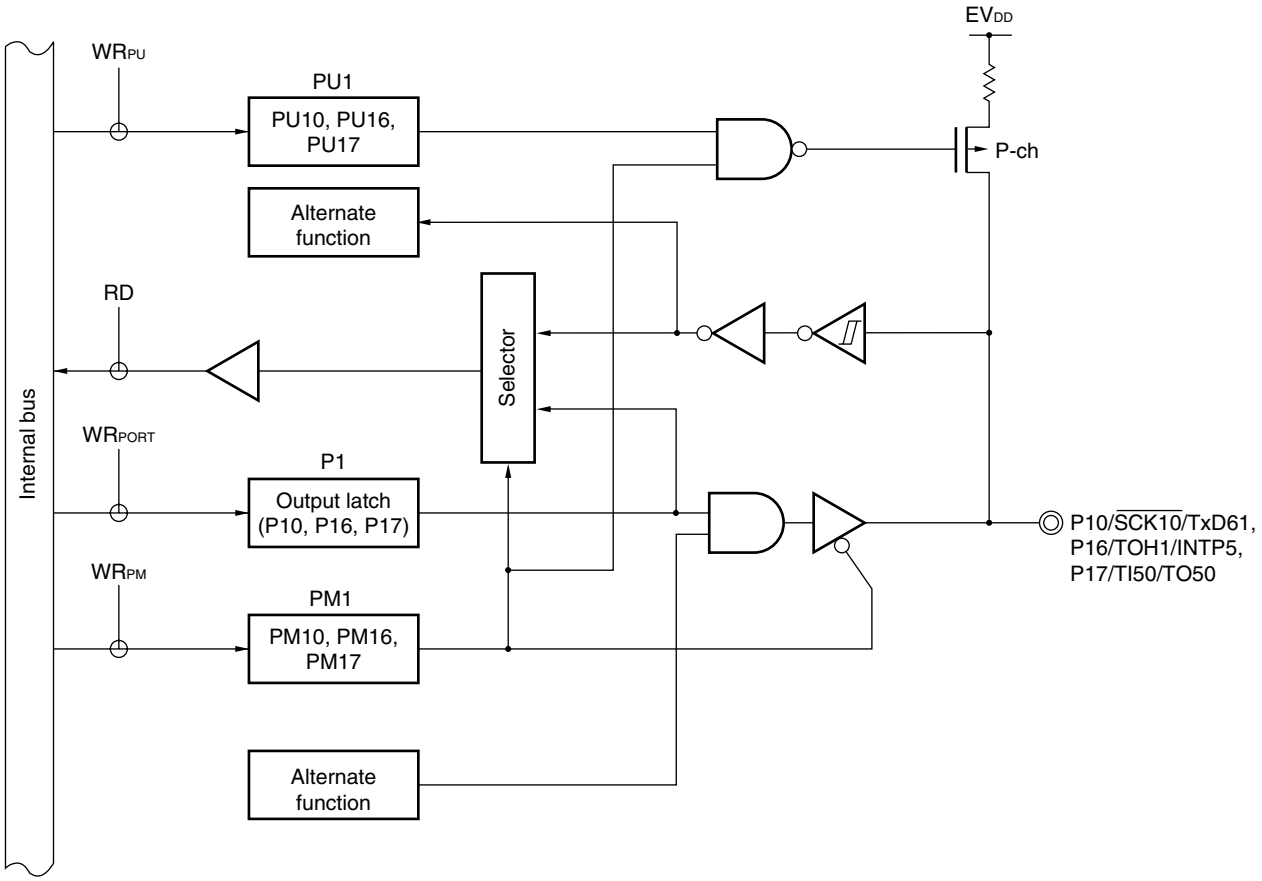
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

Reset signal generation sets port 1 to input mode.

Figures 5-4 to 5-6 show block diagrams of port 1.

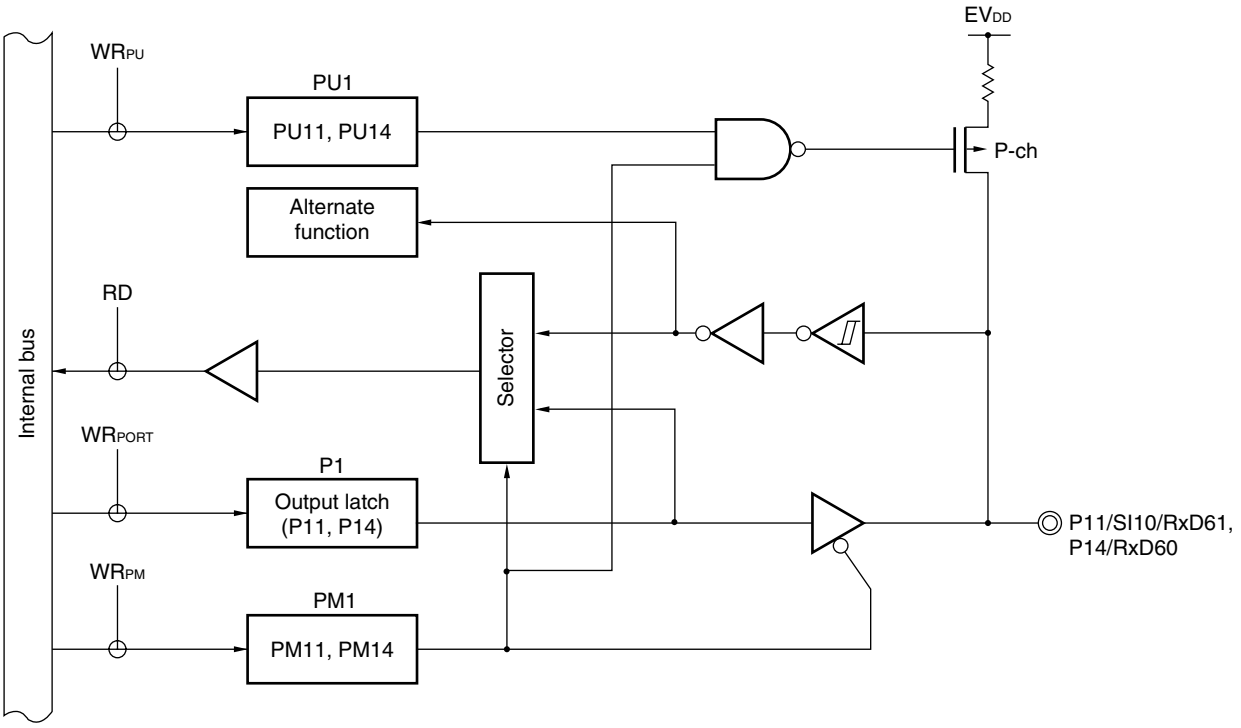
Caution To use P10/$\overline{\text{SCK10}}$/TxD61 and P12/SO10 as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection register 10 (CSIC10) to the default status (00H).

Figure 5-4. Block Diagram of P10, P16 and P17



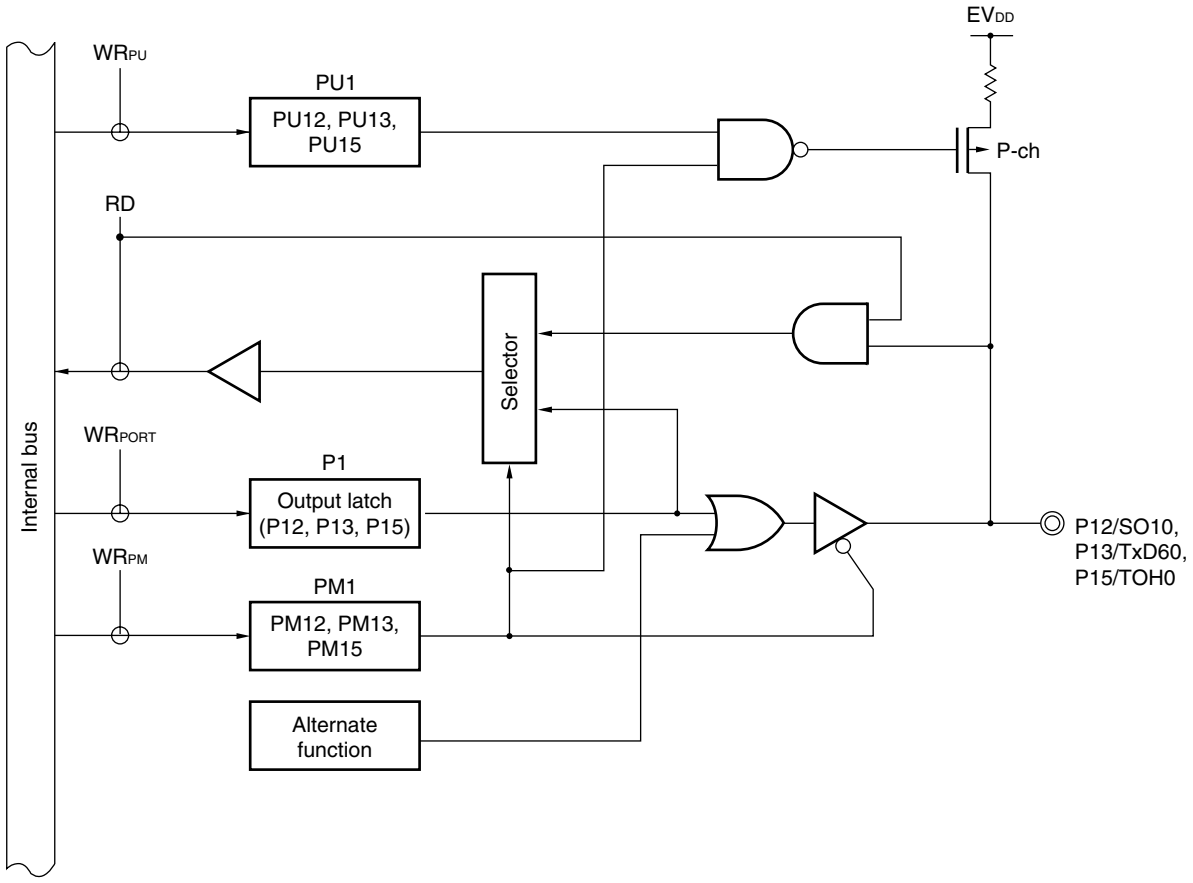
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 5-5. Block Diagram of P11 and P14



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

Figure 5-6. Block Diagram of P12, P13 and P15



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

5.2.3 Port 3

Port 3 is a 4-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When 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 and timer I/O.

Reset signal generation sets port 3 to input mode.

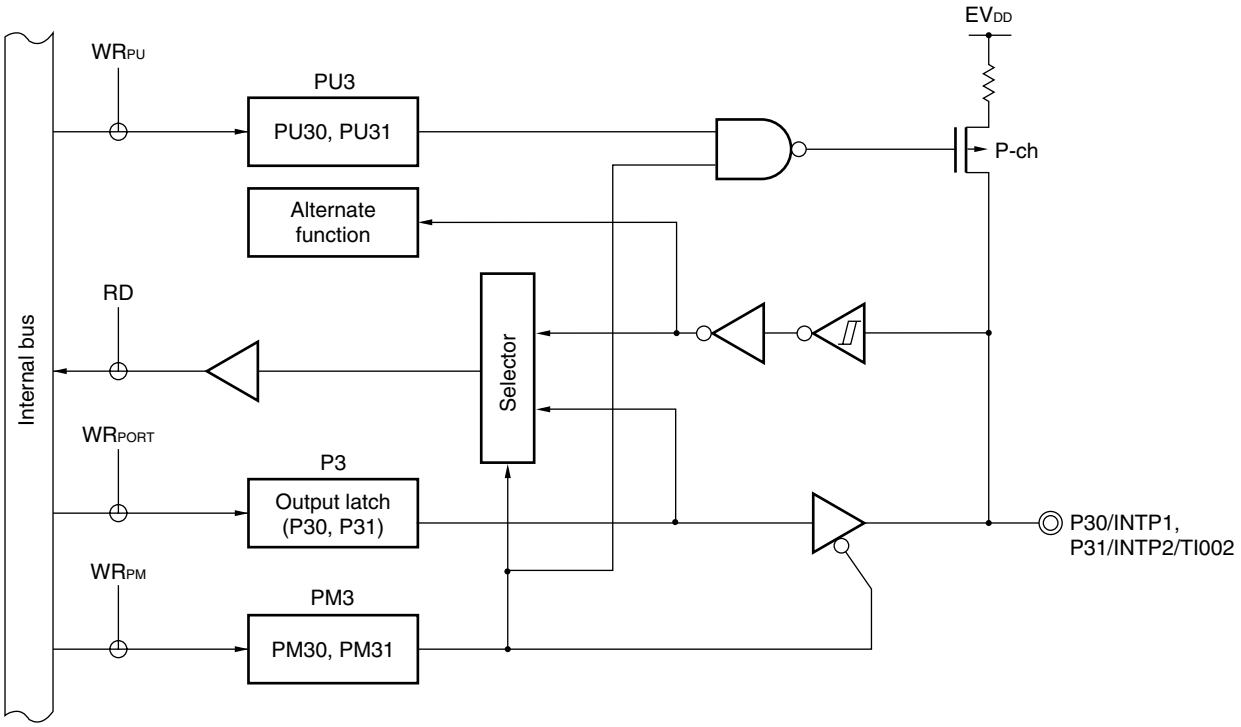
Figures 5-7 and 5-8 show block diagrams of port 3.

<R> **Cautions 1. Be sure to pull the P31 pin down before a reset release, to prevent malfunction.**

2. Connect P31/TI002/INTP2 as follows when writing the flash memory with a flash programmer.
- P31/TI002/INTP2: Connect to EV_{SS} via a resistor (10 kΩ: recommended).
The above connection is not necessary when writing the flash memory by means of self programming.

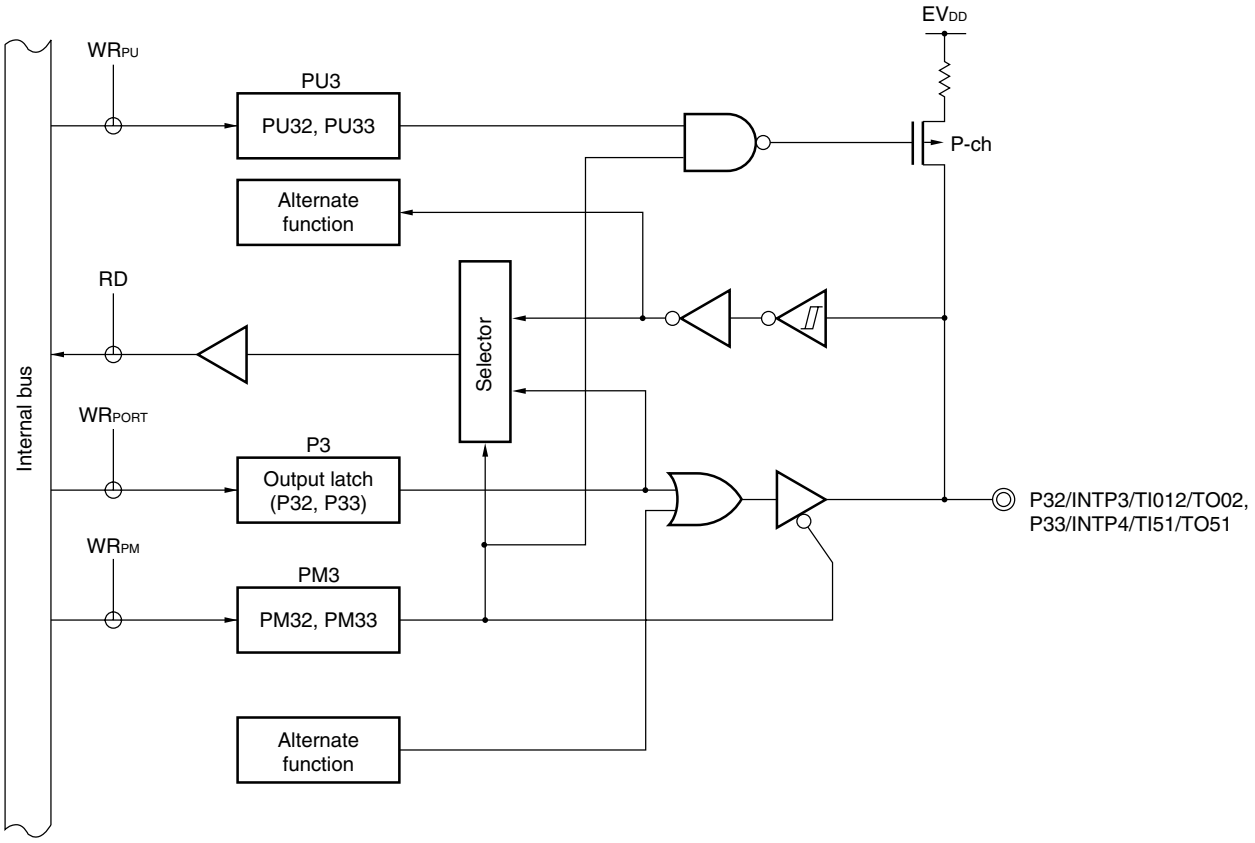
Remark P31/INTP2/TI002 and P32/INTP3/TI012/TO02 can be used for on-chip debug mode setting when the on-chip debug function is used. For details, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION.**

Figure 5-7. Block Diagram of P30 and P31



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

Figure 5-8. Block Diagram of P32 and P33



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

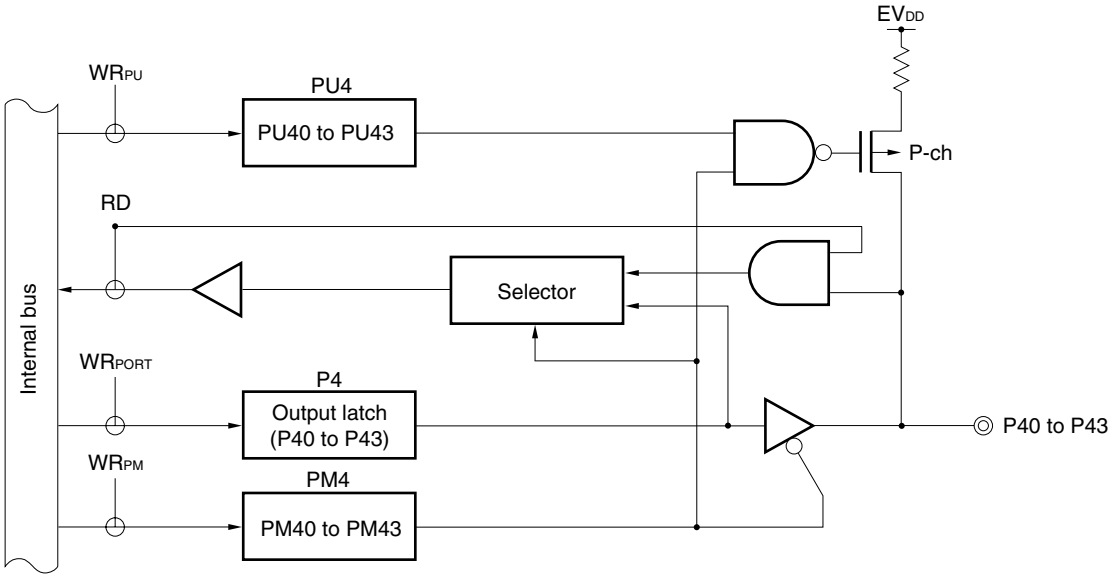
5.2.4 Port 4

Port 4 is a 4-bit I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 4 (PU4).

Reset signal generation sets port 4 to input mode.

Figure 5-9 shows a block diagram of port 4.

Figure 5-9. Block Diagram of P40 to P43



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

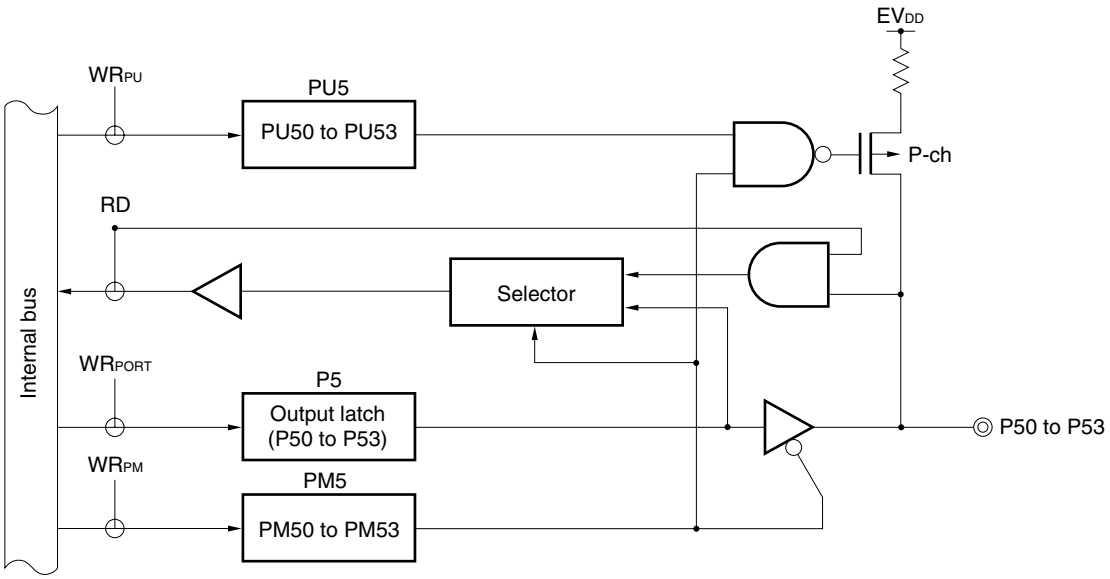
5.2.5 Port 5

Port 5 is 4-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified in 1-bit units using pull-up resistor option register 5 (PU5).

Reset signal generation sets port 5 to input mode.

Figure 5-10 shows a block diagram of port 5.

Figure 5-10. Block Diagram of P50 to P53



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

5.2.6 Port 6

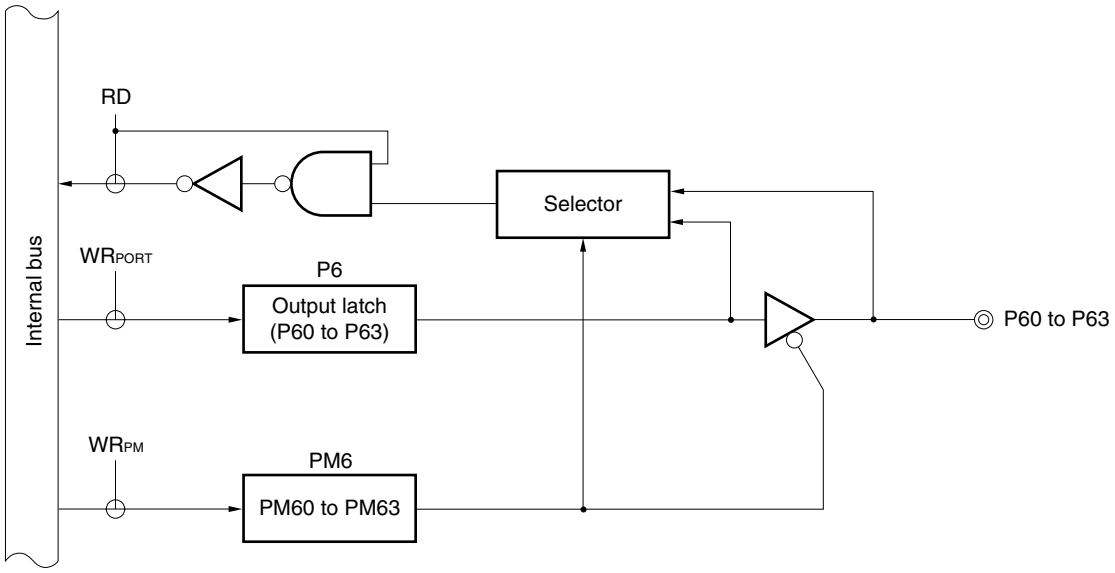
Port 6 is a 4-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

The P60 to P63 pins are N-ch open-drain pins (6 V tolerance).

Reset signal generation sets port 6 to input mode.

Figure 5-11 shows block diagram of port 6.

Figure 5-11. Block Diagram of P60 to P63



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

5.2.7 Port 7

Port 7 is an 7-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P76 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 7 (PU7).

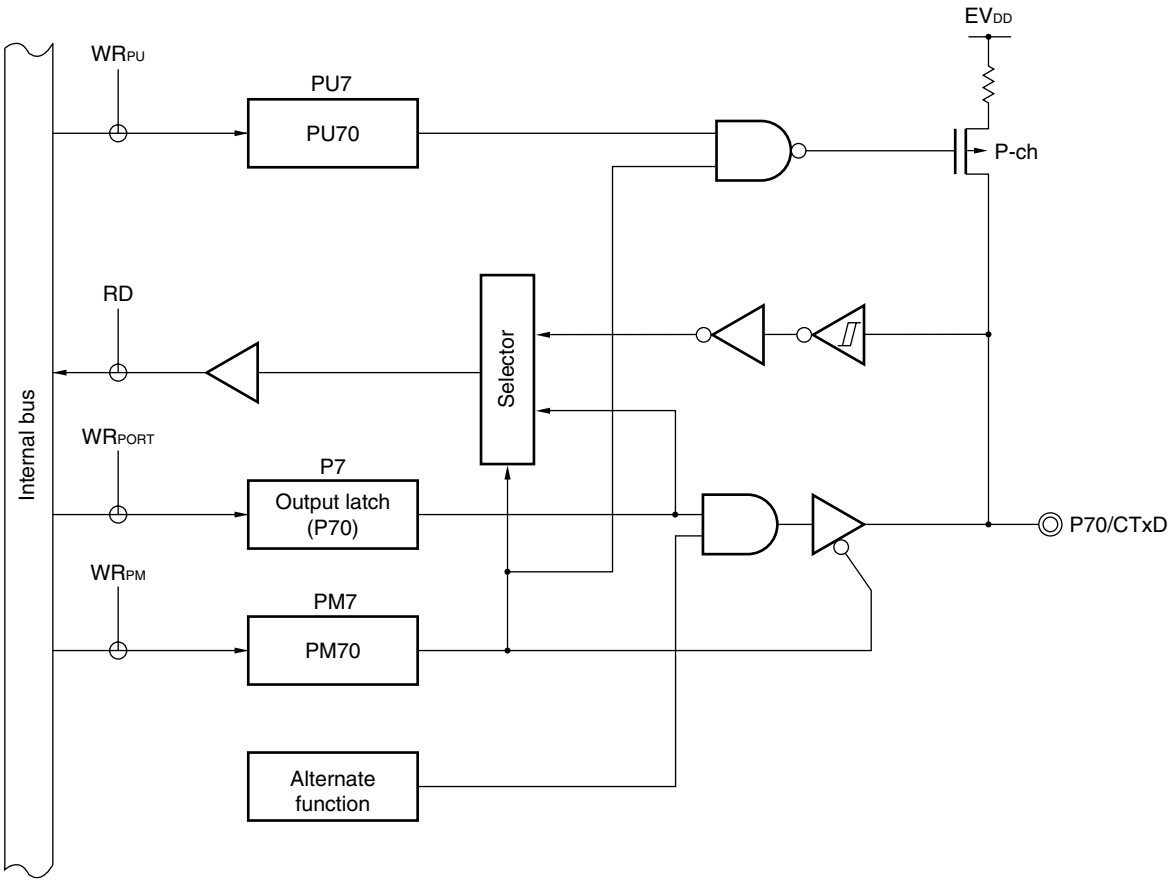
This port can also be used for external interrupt request input, and clock output pins, buzzer output pins, CAN I/F I/O, serial interface data I/O, clock I/O.

Reset signal generation sets port 7 to input mode.

Figures 5-12 to 5-16 show block diagrams of port 7.

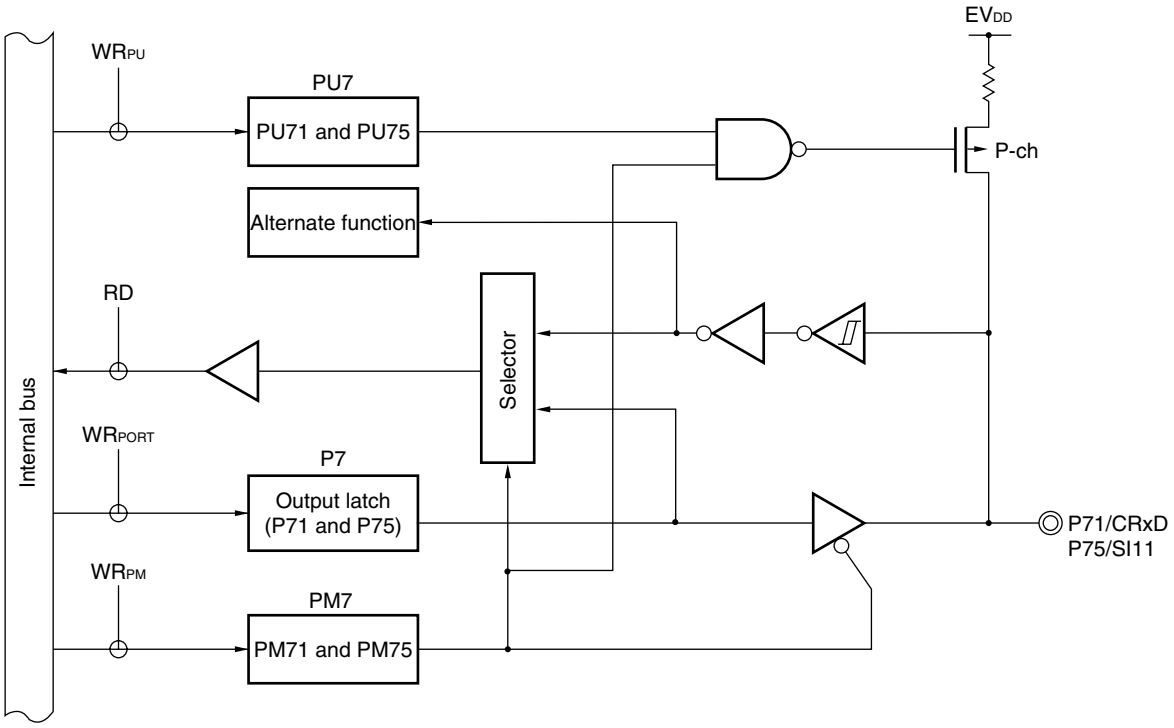
Caution To use P74/SO11 and P76/SCK11 as general-purpose ports, set serial operation mode register 10 (CSIM 10) and serial clock selection resister 10 (CSIC10) to the default status (00H).

<R> **Figure 5-12. Block Diagram of P70**



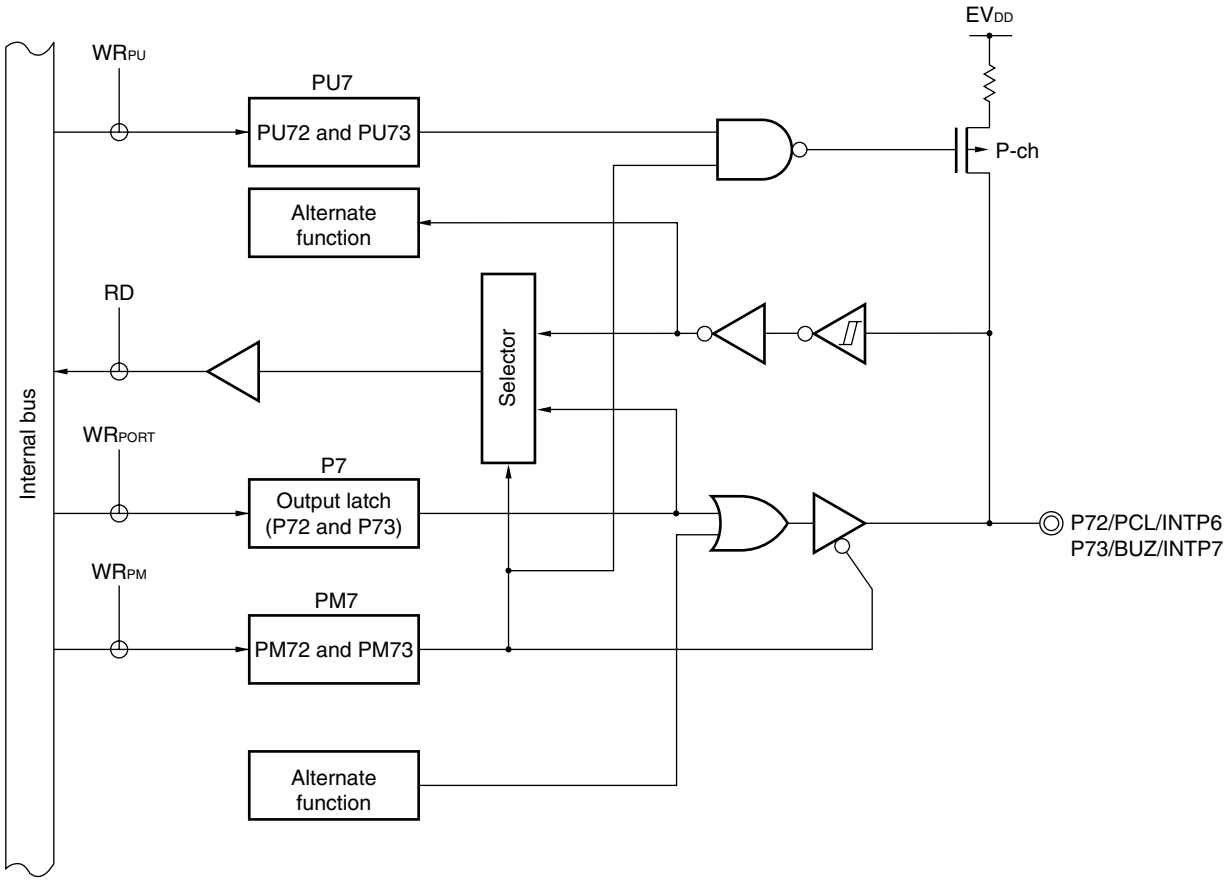
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

Figure 5-13. Block Diagram of P71 and P75



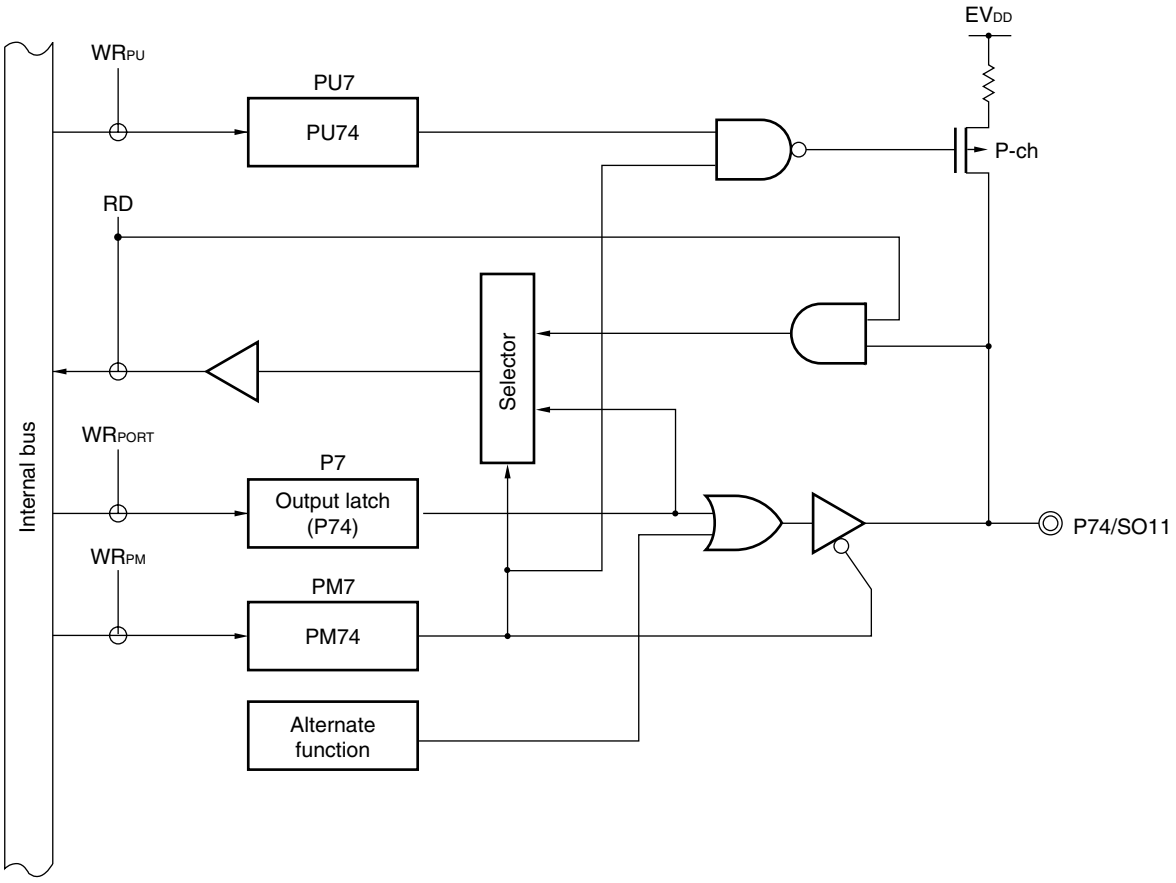
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

<R> **Figure 5-14. Block Diagram of P72 and P73**



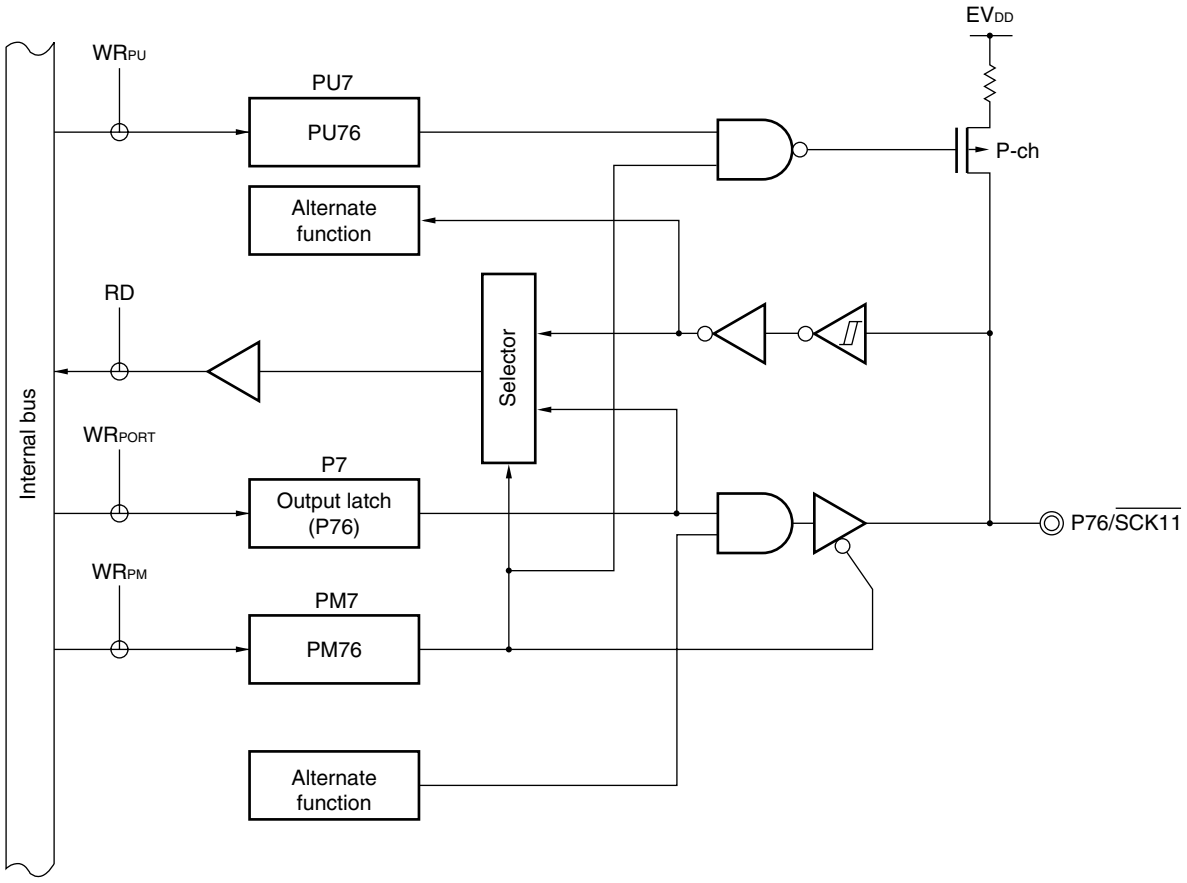
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

<R> **Figure 5-15. Block Diagram of P74**



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

<R> **Figure 5-16. Block Diagram of P76**



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

5.2.8 Port 8

Port 8 is an 8-bit I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8).

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

To use P80/ANI0 to P87/ANI7 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM8. Use these pins starting from the lower bit.

To use P80/ANI0 to P87/ANI7 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM8 (for details, see **13.3 (5) A/D port configuration register (ADPC)**).

Table 5-3. Setting Functions of P80/ANI0 to P87/ANI7 Pins

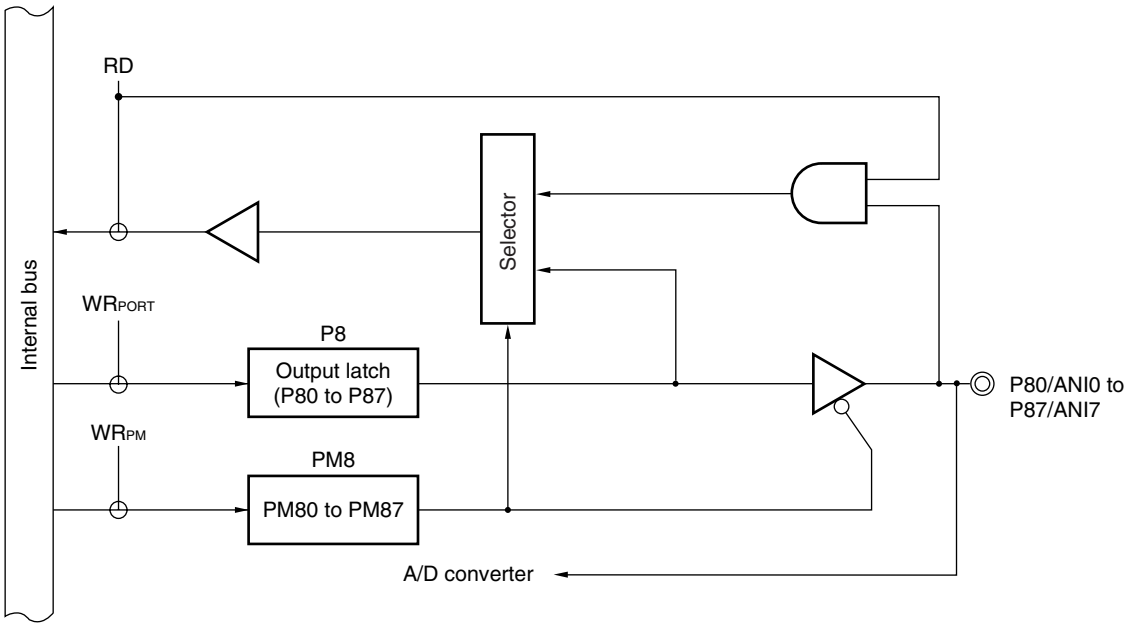
ADPC	PM8	ADS	P80/ANI0 to P87/ANI7 Pin
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P80/ANI0 to P87/ANI7 are set in the analog input mode when the reset signal is generated.

Figure 5-17 shows a block diagram of port 8.

Caution Make the AV_{REF} pin the same potential as the V_{DD} pin when port 8 is used as a digital port.

Figure 5-17. Block Diagram of P80 to P87



P8: Port register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

5.2.9 Port 9

Port 9 is an 4-bit I/O port with an output latch. Port 9 can be set to the input mode or output mode in 1-bit units using port mode register 9 (PM9).

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

To use P90/ANI8 to P93/ANI11 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM9. Use these pins starting from the lower bit.

To use P90/ANI8 to P93/ANI11 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM9 (for details, see **13.3 (5) A/D port configuration register (ADPC)**).

Table 5-4. Setting Functions of P90/ANI8 to P93/ANI11 Pins

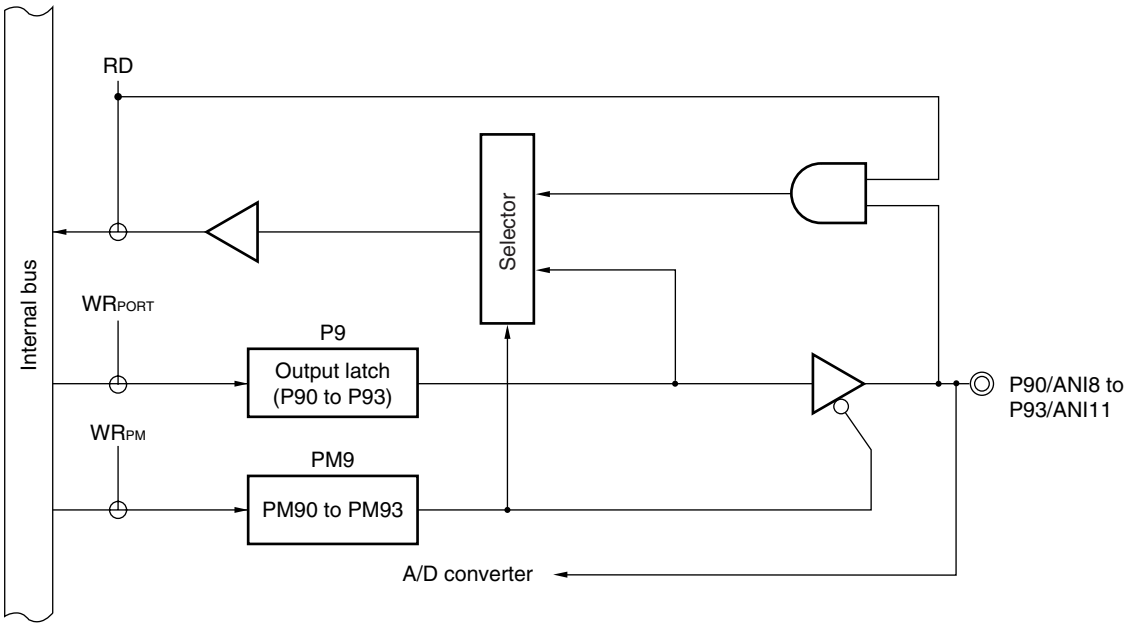
ADPC	PM9	ADS	P90/ANI8 to P93/ANI11 Pin
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P90/ANI8 to P93/ANI11 are set in the analog input mode when the reset signal is generated.

Figure 5-18 shows a block diagram of port 9.

Caution Make the AV_{REF} pin the same potential as the V_{DD} pin when port 9 is used as a digital port.

Figure 5-18. Block Diagram of P90 to P93



P9: Port register 9
PM9: Port mode register 9
RD: Read signal
WR××: Write signal

5.2.10 Port 12

Port 12 is a 5-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port only for P120, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

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

Reset signal generation sets port 12 to input mode.

Figures 5-19 and 5-20 show block diagrams of port 12.

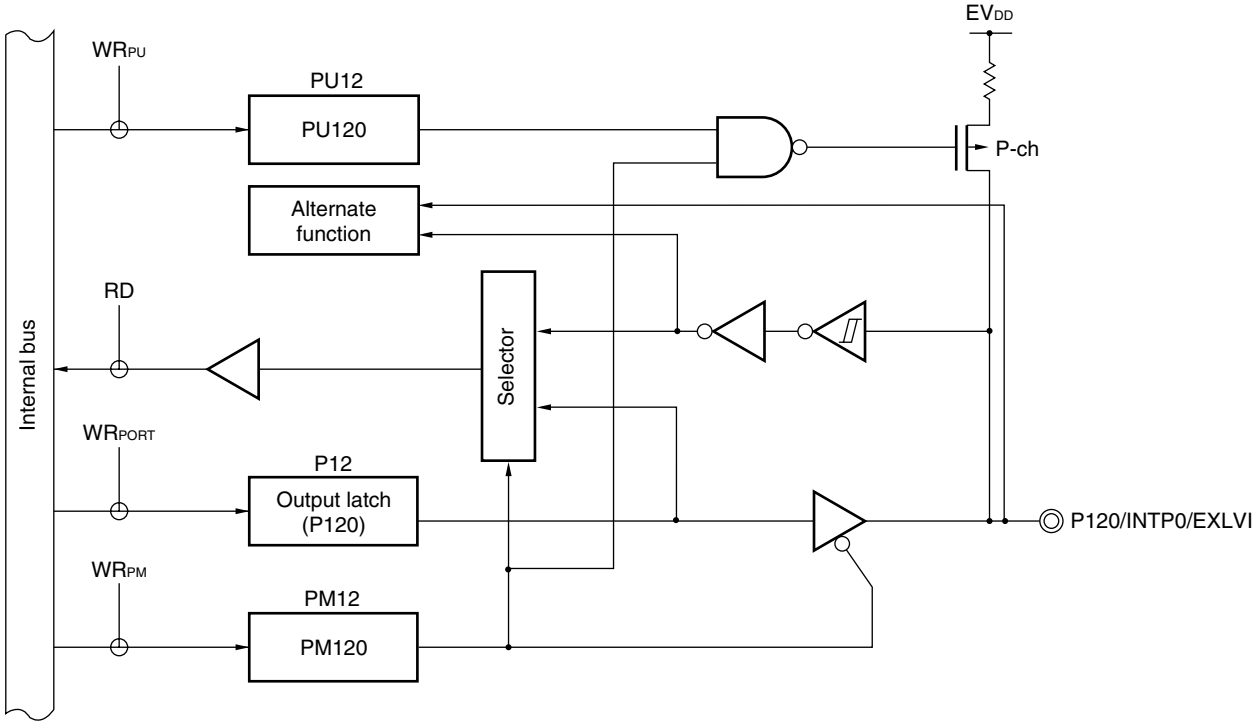
Cautions 1. When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2) or subsystem clock (XT1, XT2), or to input an external clock for the main system clock (EXCLK) or subsystem clock (EXCLKS), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for detail, see 6.3 (5) Clock operation mode select register (OSCCTL)). The reset value of OSCCTL is 00H (all of the P121 to P124 pins are I/O port pins). At this time, setting of the PM121 to PM124 and P121 to P124 pins is not necessary.

2. Connect P121/X1 as follows when writing the flash memory with a flash programmer.

- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).

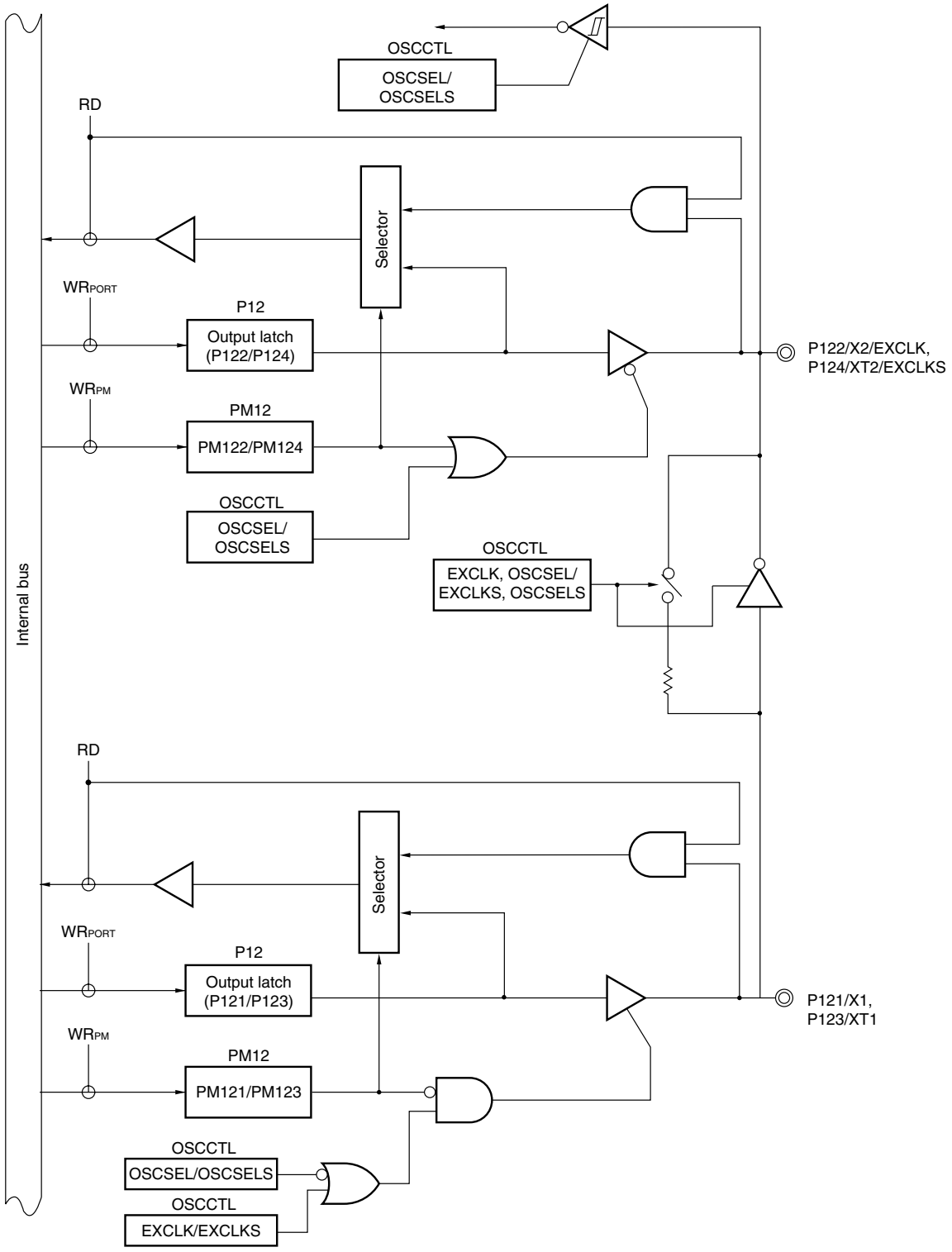
The above connection is not necessary when writing the flash memory by means of self programming.

Figure 5-19. Block Diagram of P120



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

Figure 5-20. Block Diagram of P121 to P124



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

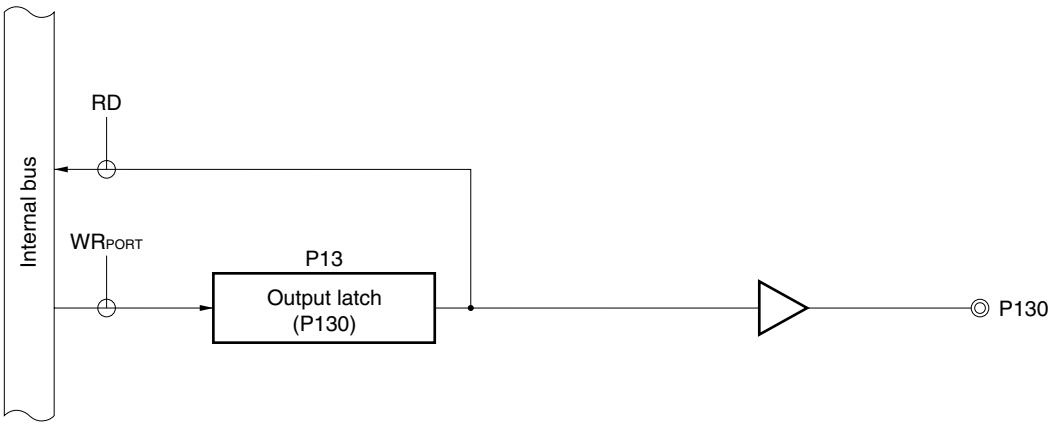
5.2.11 Port 13

Port 130 is a 1-bit output-only port.

Port 131 and 132 are 2-bit I/O port. P131 and P132 can be set to the input mode or output mode in 1-bit units using port mode register 13 (PM13). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 13 (PU13).

Figures 5-21 to 5-23 show block diagrams of port 13.

Figure 5-21. Block Diagram of P130



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

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

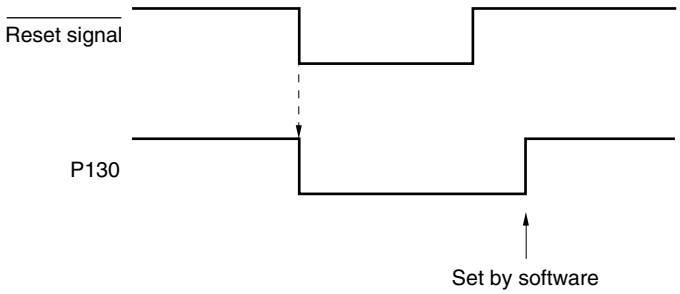
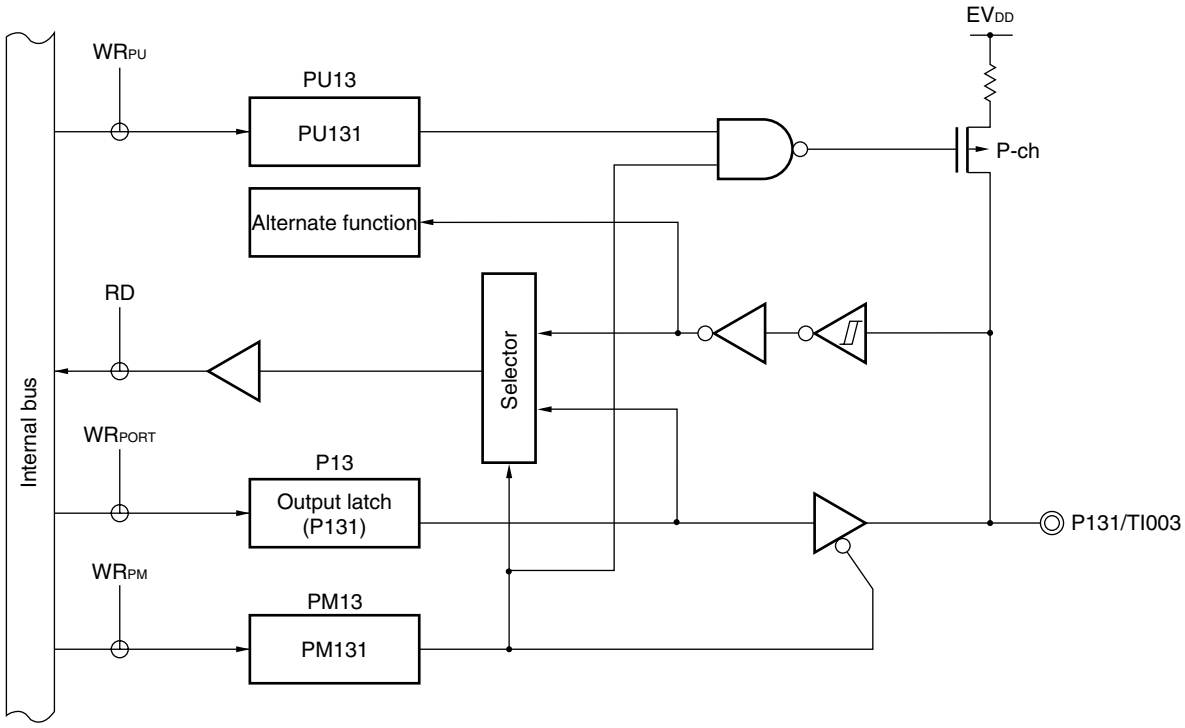
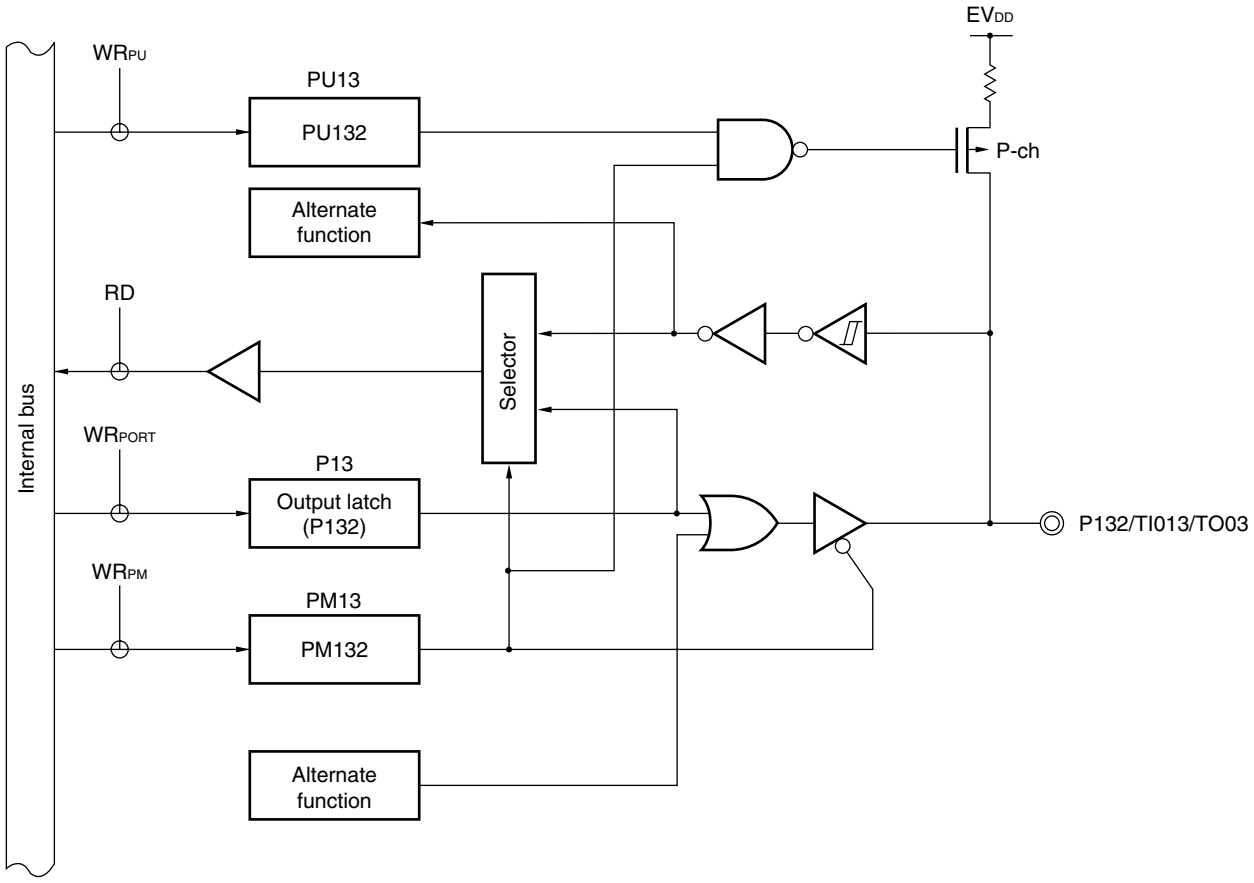


Figure 5-22. Block Diagram of P131



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WR××: Write signal

Figure 5-23. Block Diagram of P132



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WR××: Write signal

5.3 Registers Controlling Port Function

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

- Port mode registers (PM0, PM1, PM3 to PM9, PM12, PM13)
- Port registers (P0, P1, P3 to P9, P12, P13)
- Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU13)
- A/D port configuration register (ADPC)

(1) Port mode registers (PM0, PM1, PM3 to PM9, PM12, PM13)

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 except for PM13. PM13 is set to FEH.

When port pins are used as alternate-function pins, set the port mode register and output latch as shown in Table 5-5.

Figure 5-24. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	1	1	1	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM5	1	1	1	1	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	1	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM9	1	1	1	1	PM93	PM92	PM91	PM90	FF29H	FFH	R/W
PM12	1	1	1	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W
PM13	1	1	1	1	1	PM132	PM131	0	FF2DH	FEH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3 to 9, 12, 13; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Table 5-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI000	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P05	$\overline{\text{SSI11}}$	Input	1	×
	TI001	Input	1	×
P06	TI011	Input	1	×
	TO01	Output	0	0
P10	$\overline{\text{SCK10}}$	Input	1	×
		Output	0	1
	TxD61	Output	0	1
P11	SI10	Input	1	×
	RxD61	Input	1	×
P12	SO10	Output	0	0
P13	TxD60	Output	0	1
P14	RxD60	Input	1	×
P15	TOH0	Output	0	0
P16	TOH1	Output	0	0
	INTP5	Input	1	×
P17	TI50	Input	1	×
	TO50	Output	0	0
P30	INTP1	Input	1	×
P31	INTP2	Input	1	×
	TI002	Input	1	×
P32	INTP3	Input	1	×
	TI012	Input	1	×
	TO02	Output	0	0
P33	INTP4	Input	1	×
	TI51	Input	1	×
	TO51	Output	0	0
P70	CTxD	Output	0	1
P71	CRxD	Input	1	×
P72	PCL	Output	0	0
	INTP6	Input	1	×
P73	BUZ	Output	0	0
	INTP7	Input	1	×
P74	SO11	Output	0	0
P75	SI11	Input	1	×

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

Table 5-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P76	$\overline{\text{SCK11}}$	Input	1	×
		Output	0	1
P80-P87	ANI0-ANI7	Input	1	×
P90-P93	ANI8-ANI11	Input	1	×
P120	INTP0	Input	1	×
	EXLVI	Input	1	×
P121	X1	Input	1	×
P122	X2	Input	1	×
	EXCLK	Input	1	×
P123	XT1	Input	1	×
P124	XT2	Input	1	×
	EXCLKS	Input	1	×
P131	TI003	Input	1	×
P132	TI013	Input	1	×
	TO03	Output	0	0

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

Caution When using P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 in the input mode, not only PM8 and PM9 (input/output) but also the A/D port configuration register (ADPC) (analog input/digital input) must be set (for details, see 13.3 (4) Analog input channel specification register (ADS) to (7) Port mode register 9 (PM9)). The reset value of ADPC is 00H (P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 are all analog input pins).

(2) Port registers (P0, P1, P3 to P9, P12, P13)

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 value of the output latch 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 5-25. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	P06	P05	0	0	0	P01	P00	FF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
P4	0	0	0	0	P43	P42	P41	P40	FF04H	00H (output latch)	R/W
P5	0	0	0	0	P53	P52	P51	P50	FF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FF06H	00H (output latch)	R/W
P7	0	P76	P75	P74	P73	P72	P71	P70	FF07H	00H (output latch)	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FF08H	00H (output latch)	R/W
P9	0	0	0	0	P93	P92	P91	P90	FF09H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FF0CH	00H (output latch)	R/W
P13	0	0	0	0	0	P132	P131	P130	FF0DH	00H (output latch)	R/W

Pmn	m = 0, 1, 3 to 9, 12, 13; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Remark An undefined value (pin input level) is read for the value after reset when P0 is read in the input mode. When P8 and P9 are read in the output mode, 00H (output latch value) is output.

(3) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU13)

These registers specify whether the on-chip pull-up resistors of P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P131 and P132 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 PU0, PU1, PU3 to PU5, PU7, PU12, and PU13. 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 PU0, PU1, PU3 to PU5, PU7, PU12, and PU13.

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

Reset signal generation clears these registers to 00H.

Figure 5-26. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU06	PU05	0	0	0	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU40	FF34H	00H	R/W
PU5	0	0	0	0	PU53	PU52	PU51	PU50	FF35H	00H	R/W
PU7	0	PU76	PU75	PU74	PU73	PU72	PU71	PU70	FF37H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
PU13	0	0	0	0	0	PU132	PU131	0	FF3DH	00H	R/W

PUmn	PUmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 5, 7, 12, 13, n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

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

This register switches the P80/ANI0 to P87/ANI7 and P90/ANI8 to P93/ANI11 pins to digital I/O of port or analog input of A/D converter.

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

Reset signal generation clears this register to 00H.

Table 5-6. Format of A/D Port Configuration Register (ADPC)

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/ digital input (D) switching											
				P93/ ANI11	P92/ ANI10	P91/ ANI9	P90/ ANI8	P87/ ANI7	P86/ ANI6	P85/ ANI5	P84/ ANI4	P83/ ANI3	P82/ ANI2	P81/ ANI1	P80/ ANI0
0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
0	1	1	1	A	A	A	A	A	D	D	D	D	D	D	D
1	0	0	0	A	A	A	A	D	D	D	D	D	D	D	D
1	0	0	1	A	A	A	D	D	D	D	D	D	D	D	D
1	0	1	0	A	A	D	D	D	D	D	D	D	D	D	D
1	0	1	1	A	D	D	D	D	D	D	D	D	D	D	D
1	1	0	0	D	D	D	D	D	D	D	D	D	D	D	D
Other than above				Setting prohibited											

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 8 (PM8) and port mode register 9 (PM9).

2. If data is written to ADPC, a wait cycle is generated. Do not write data to ADPC when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

5.4 Port Function Operations

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

5.4.1 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 by reset.

(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.

5.4.2 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.

5.4.3 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 by reset.

(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.

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

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

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

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

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

A 1-bit manipulation instruction is executed in the following order in the 78K0/FE2.

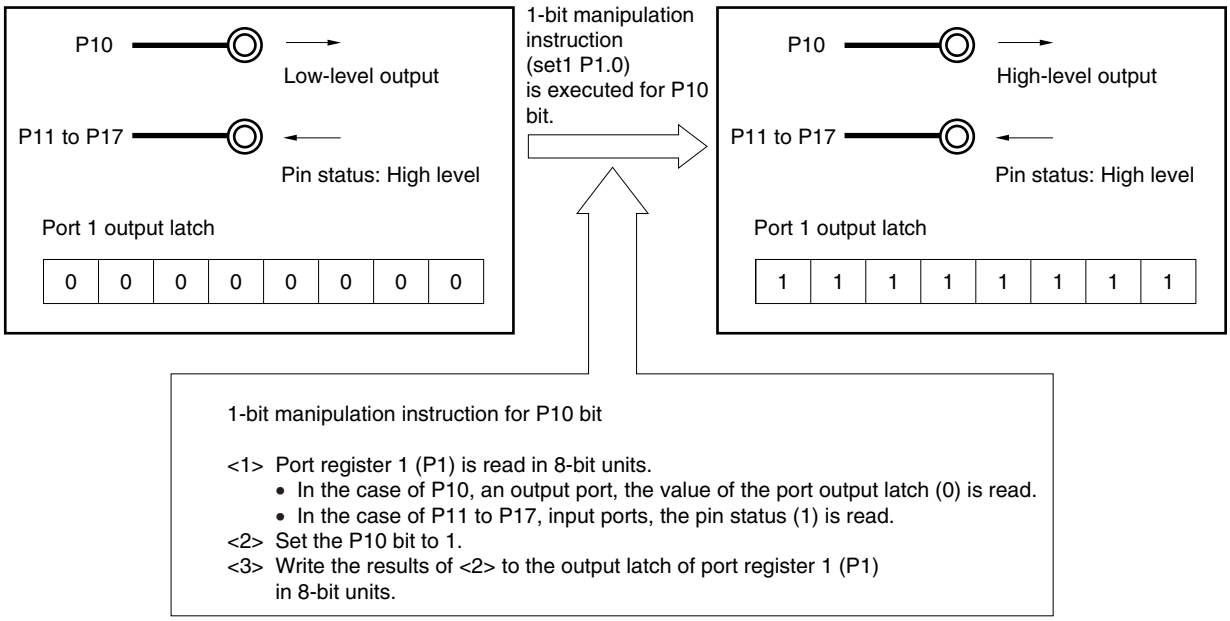
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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

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

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

<R> **Figure 5-27. Bit Manipulation Instruction (P10)**



CHAPTER 6 CLOCK GENERATOR

6.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 4 to 20 MHz. Oscillation can be stopped by executing the STOP instruction or using the main OSC control register (MOC).

<2> Internal high-speed oscillator

This circuit oscillates a clock of f_{RH} = 8 MHz (TYP.). After a $\overline{\text{RESET}}$ 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 oscillator mode register (RCM).

An external main system clock (f_{EXCLK} = 4 to 20 MHz) can also be supplied from the EXCLK pin. As the main system clock, a high-speed system clock (X1 clock or external main system clock) or internal high-speed oscillation clock can be selected by using the main clock mode register (MCM).

(2) Subsystem clock

- **Subsystem clock oscillator**

This circuit oscillates at a frequency of f_{XT} = 32.768 kHz. Oscillation can be stopped by using the processor clock control register (PCC) and clock operation mode select register (OSCCTL). An external subsystem clock (f_{EXCLKS} = 32.768 kHz) can also be supplied from the EXCLKS pin.

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

- **Internal low-speed oscillator**

This circuit oscillates a clock of f_{RL} = 240 kHz (TYP.). After a $\overline{\text{RESET}}$ release, the internal low-speed oscillation clock always starts operating. Oscillation can be stopped by using the internal oscillator mode register (RCM).

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
- TMH1 (f_{RL}, $f_{RL}/2^7$, $f_{RL}/2^9$)

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

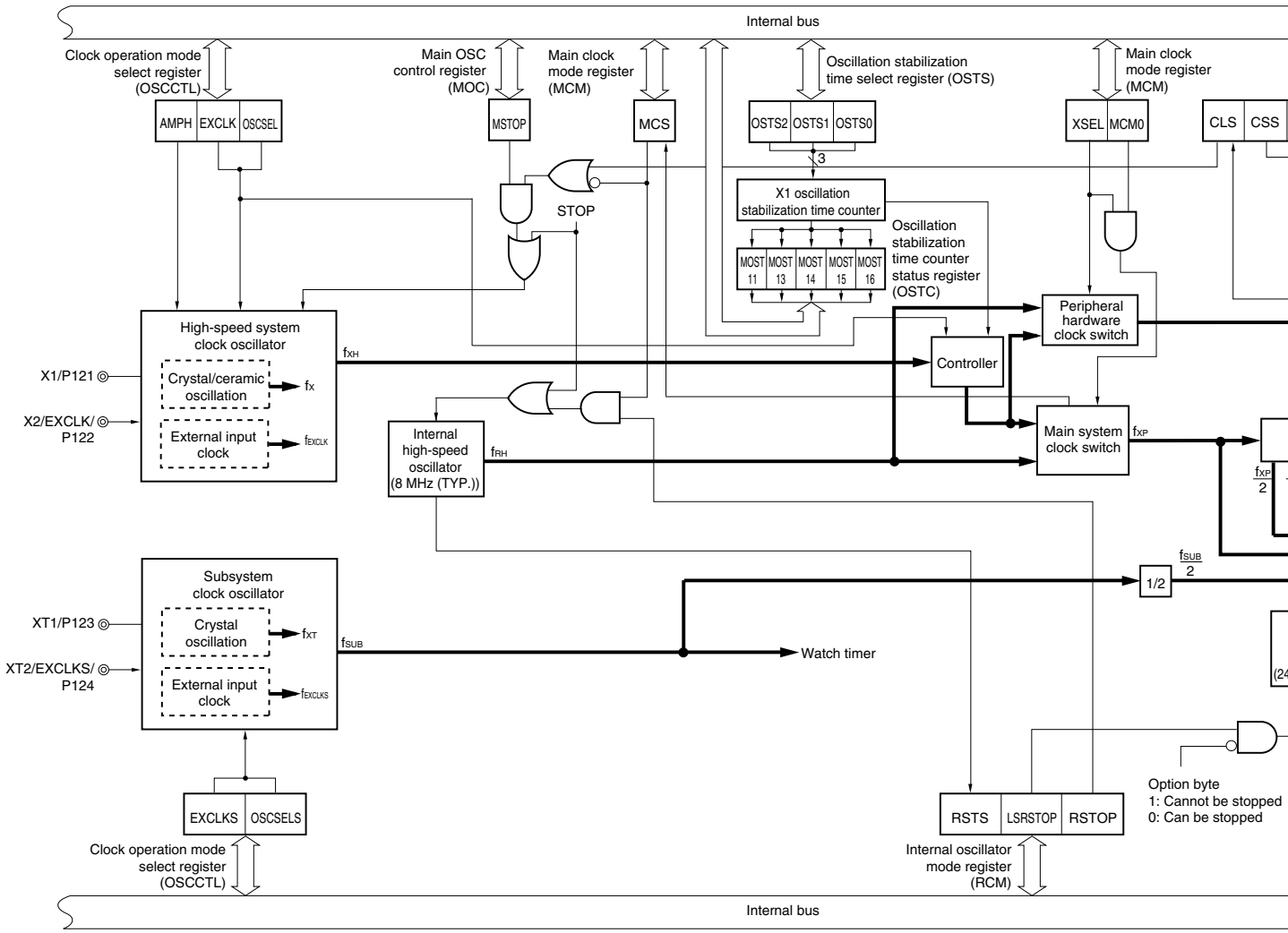
6.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 6-1. Configuration of Clock Generator

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

Figure 6-1. Block Diagram of Clock Generator



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

6.3 Registers Controlling Clock Generator

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

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

(1) Processor clock control register (PCC)

This register is used to select the CPU clock and the division ratio.

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

Reset signal generation sets PCC to 01H.

Figure 6-2. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 01H R/W[Note 1]

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

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

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

Notes 1. Bit 5 is read-only.

2. Be sure to switch CSS from 1 to 0 when bits 1 (MCS) and 0 (MCM0) of the main clock mode register (MCM) are 1.

Caution Be sure to clear bits 3 and 6 to 0.

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

2. f_{SUB}: Subsystem clock frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/FE2. Therefore, the relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 6-2.

Table 6-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$			
	High-Speed System Clock[Note]		Internal high-speed Oscillator Clock[Note]	Subsystem Clock
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{XP}	0.2 μs	0.1 μs	0.25 μs (TYP.)	–
$f_{XP}/2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	–
$f_{XP}/2^2$	0.8 μs	0.4 μs	1.0 μs (TYP.)	–
$f_{XP}/2^3$	1.6 μs	0.8 μs	2.0 μs (TYP.)	–
$f_{XP}/2^4$	3.2 μs	1.6 μs	4.0 μs (TYP.)	–
$f_{SUB}/2$	–		–	122.1 μs

Note The main clock mode register (MCM) is used to set the CPU clock (high-speed system clock/internal high-speed oscillation clock) (see **Figure 6-4**).

(2) Internal oscillator mode register (RCM)

This register sets the operation mode of internal oscillator.

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

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

Figure 6-3. Format of Internal Oscillator Mode Register (RCM)

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

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

RSTS	Status of internal high-speed oscillator oscillation
0	Waiting for stabilization of internal high-speed oscillator oscillation in high-accuracy mode (internal high-speed oscillator operation in low-accuracy mode)
1	Internal high-speed oscillator operation in high-accuracy 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 oscillation has been stabilized.

2. Bit 7 is read-only.

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

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

(3) 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 6-4. Format of Main Clock Mode Register (MCM)

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

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

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

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

Note Bit 1 is read-only.

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

2. 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.

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

- **Watchdog timer**
- **When "$f_{RL}/2^7$" is selected as the count clock for 8-bit timer H1**
- **Peripheral hardware selects the external clock as the clock source (Except when the external count clock of TM0n (n = 0, 1) is selected (TI00n pin valid edge))**

4. It takes one clock to change the CPU clock.

(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 6-5. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 80H R/W

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

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

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

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

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

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

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

(5) Clock operation mode select register (OSCCTL)

This register selects the operation modes of the high-speed system and subsystem clocks.

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

Reset signal generation clears this register to 00H.

Figure 6-6. Format of Clock Operation Mode Select Register (OSCCTL)

Address: FFEFH After reset: 00H R/W

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

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

EXCLKS	OSCSELS	Subsystem clock operation mode	P123/XT1 pin	P124/XT2/EXCLKS pin
0	0	I/O port mode	I/O port	
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	I/O port mode	I/O port	
1	1	External clock input mode	I/O port	External clock input

AMPH	Operating frequency control
0	4 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 peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. When the high-speed system clock (X1 oscillation) is selected as the CPU clock, supply of the CPU clock is stopped for 4.06 to 16.12 μs after AMPH is set to 1. When the high-speed system clock (external clock input) is selected as the CPU clock, supply of the CPU clock is stopped for the duration of 160 external clocks after AMPH is set to 1.

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

4. AMPH can be changed only once after a reset release.

Cautions 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. To change the value of EXCLKS and OSCSELS, confirm that bit 5 (CLS) of the processor clock control register (PCC) is 0 (the CPU is operating with the high-speed system clock).

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

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

This is the status register of the X1 clock oscillation stabilization time counter. If the internal high-speed oscillation clock or subsystem clock is 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 6-7. 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		
						fx = 10 MHz	fx = 20 MHz
1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	0	0	2^{14}/fx min.	1.64 ms min.	819.2 μs min.
1	1	1	1	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	2^{16}/fx min.	6.55 ms min.	3.27 ms min.

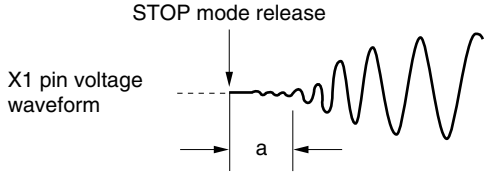
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the internal high-speed oscillation clock or subsystem 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**

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

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



Remark fx: X1 clock oscillation frequency

(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. The wait time set by OSTS is valid only after the STOP mode is released with the X1 clock selected as the CPU clock. After the STOP mode is released with the internal high-speed oscillation clock or subsystem clock selected as the CPU clock, the oscillation stabilization time must be confirmed by OSTC.

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

Reset signal generation sets OSTS to 05H.

Figure 6-8. 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.

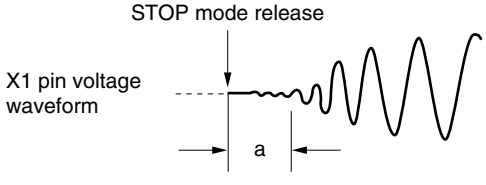
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. If the STOP mode is entered and then released while the internal high-speed oscillation clock or subsystem 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**

The oscillation stabilization time counter counts up to the 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

6.4 System Clock Oscillator

6.4.1 X1 oscillator

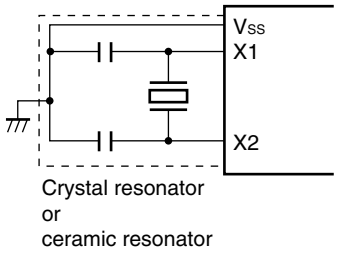
The X1 oscillator oscillates with a crystal resonator or ceramic resonator (4 to 20 MHz) connected to the X1 and X2 pins.

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

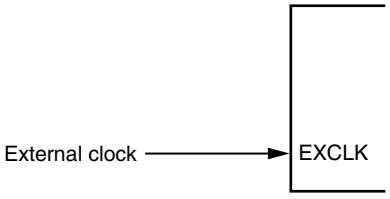
Figure 6-9 shows an example of the external circuit of the X1 oscillator.

Figure 6-9. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

6.4.2 XT1 oscillator

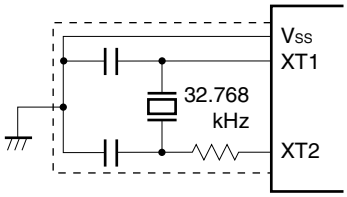
The XT1 oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

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

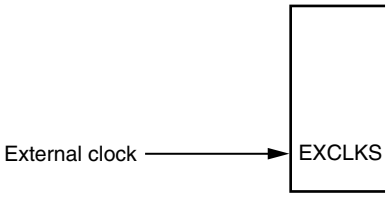
Figure 6-10 shows an example of the external circuit of the XT1 oscillator.

Figure 6-10. Example of External Circuit of XT1 Oscillator

(a) Crystal oscillation



(b) External clock



Caution is listed on the next page.

Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 6-9 and 6-10 to avoid an adverse effect from wiring capacitance.

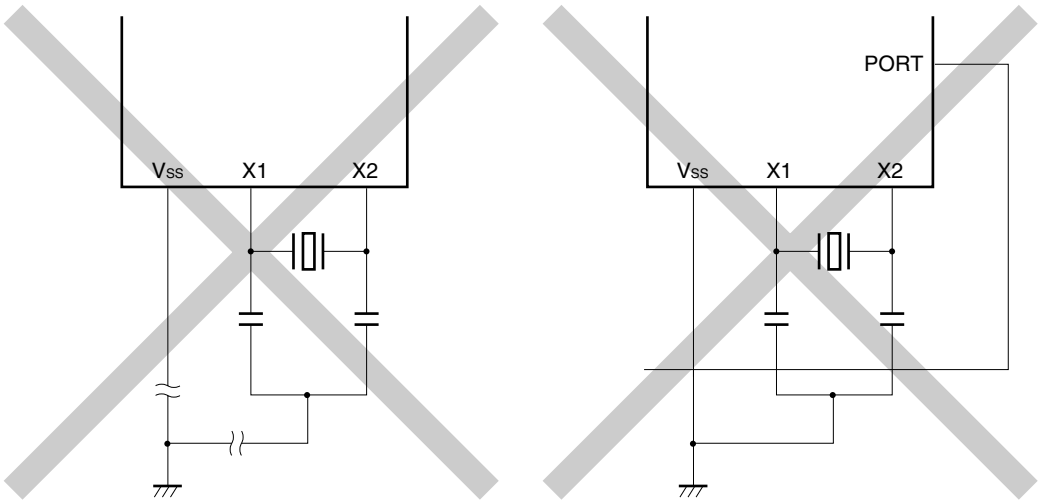
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

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

Figure 6-11 shows examples of incorrect resonator connection.

Figure 6-11. Examples of Incorrect Resonator Connection (1/2)

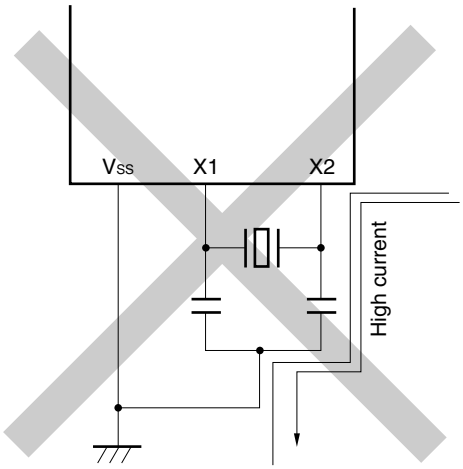
(a) Too long wiring **(b) Crossed signal line**



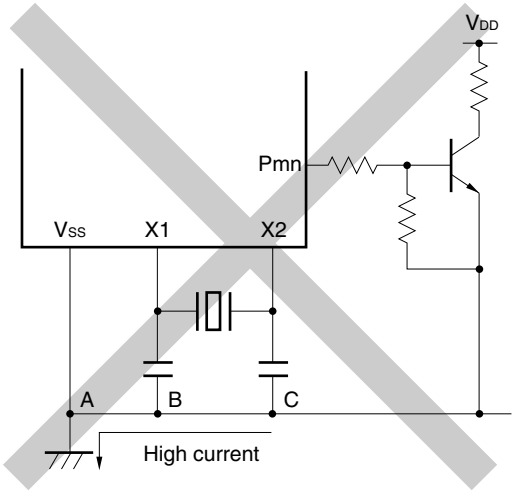
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 6-11. Examples of Incorrect Resonator Connection (2/2)

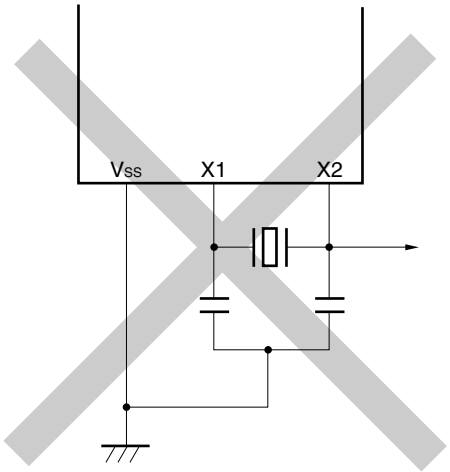
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

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

6.4.3 When subsystem clock is not used

If it is not necessary to use the subsystem clock for low power consumption operations, or if not using the subsystem clock as an I/O port, set the XT1 and XT2 pins to I/O mode (OSCSELS = 0) and connect them as follows.

Input (PM123/PM124 = 1): Independently connect to V_{DD} or V_{SS} via a resistor.
Output (PM123/PM124 = 0): Leave open.

Remark OSCSELS: Bit 4 of clock operation mode select register (OSCCTL)
PM123, PM124: Bits 3 and 4 of port mode register 12 (PM12)

6.4.4 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0/FE2. Oscillation can be controlled by the internal oscillator mode register (RCM).

After a $\overline{\text{RESET}}$ release, the internal high-speed oscillation clock starts oscillation (8 MHz (TYP.)).

6.4.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0/FE2.

The internal low-speed oscillator 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 oscillator mode register (RCM).

After a $\overline{\text{RESET}}$ release, the internal low-speed oscillation clock starts oscillation and the watchdog timer is operated (240 kHz (TYP.)).

6.4.6 Prescaler

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

6.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode.

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

The CPU starts operation when the on-chip internal high-speed oscillator starts outputting after a reset release in the 78K0/FE2, thus enabling the following.

(1) Enhancement of security function

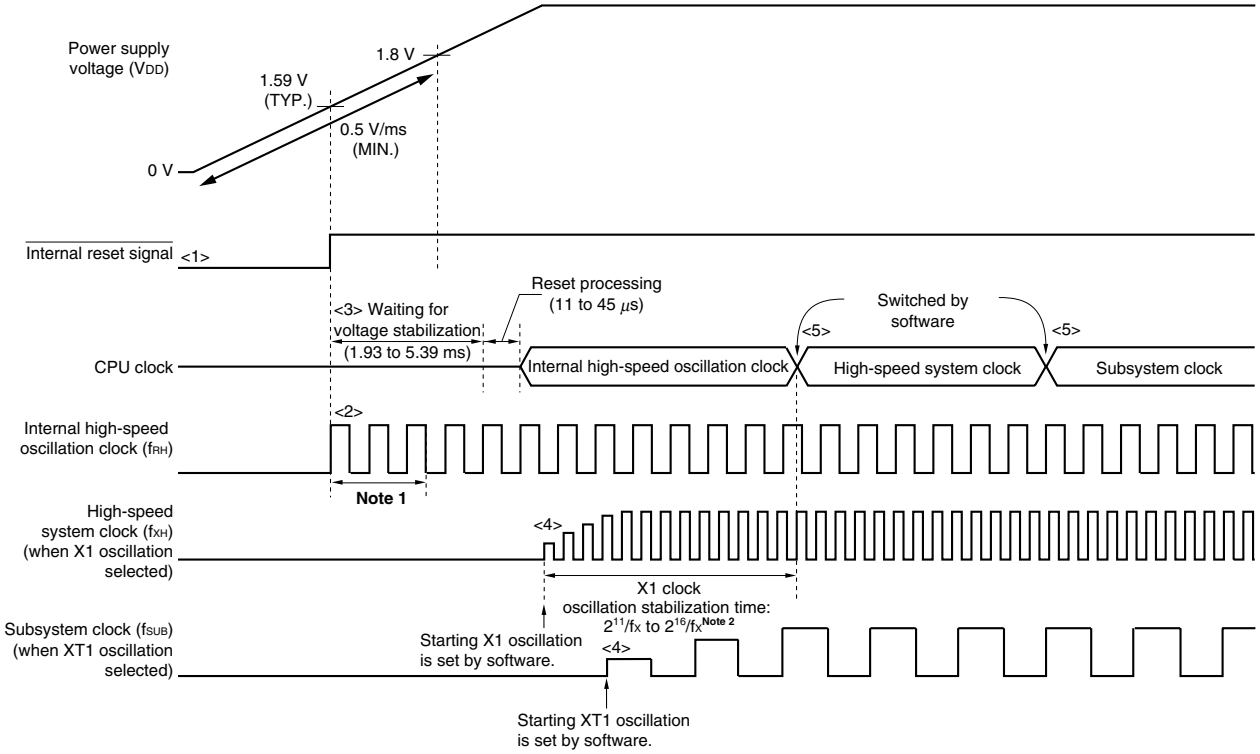
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the on-chip internal high-speed oscillation clock, so the device can 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.

A timing diagram of the CPU default start using the internal high-speed oscillation clock is shown in Figure 6-12 and 6-13.

Figure 6-12 Operation of the clock generating circuit when power supply voltage injection (When 1.59 V POC mode setup (option byte: LVISTART = 0))



<1> The internal reset signal by the power-on clear (POC) circuit is generated after a power supply injection.

<2> If power supply voltage exceeds 1.59 V (TYP.), reset will be released and the oscillation start of the high-speed oscillator will be carried out automatically.

<3> If power supply voltage is rose by inclination of 0.5 V/ms (MAX.), after the voltage stable waiting time of a power supply/regulator passed after reset release and reset processing will be performed, CPU carries out a start of operation with high-speed oscillation clock .

<4> One clock or XT1 clock should set up an oscillation start by software (see **(1)** in **6.6.1 Controlling high-speed system clock** and **(1)** in **6.6.3 Example of controlling subsystem clock**).

<5> When you change CPU to X1 clock or XT1 clock, set up a change by software after the oscillation stability waiting of a clock (see **(3)** in **6.6.1 Controlling high-speed system clock** and **(3)** in **6.6.3 Example of controlling subsystem clock**).

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

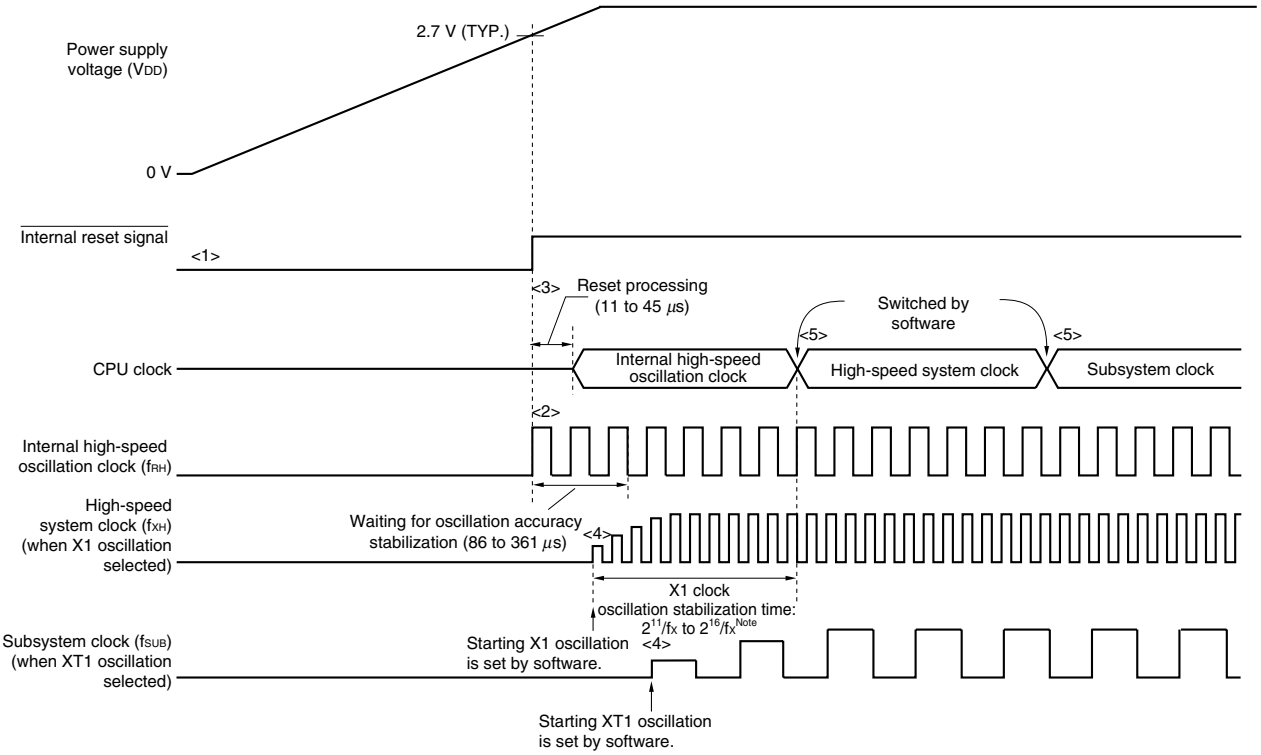
2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed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

Cautions 1. When the standup of voltage until it reaches 1.8 V from the time of a power supply injection is looser than 0.5 V/ms (MAX.), input a low level into $\overline{\text{RESET}}$ pin, or set up 2.7 V/1.59 V POC mode (LVISTART = 1) from an option byte until it reaches 1.8 V from the time of a power supply injection (refer to Figure 6-13). When a low level is inputted into $\overline{\text{RESET}}$ pin until it reaches 1.8 V, after the reset release by $\overline{\text{RESET}}$ pin operates to the same timing as <2> of Figure 6-12 or subsequent ones.

2. When using the external clock input from EXCLK pin and EXCLKS pin, oscillation stable waiting time is unnecessary.

remark The clock which is not used as a CPU clock can be suspended by setup of software during microcomputer operation. Moreover, high-speed oscillation clock and a high-speed system clock can suspend a clock by execution of a STOP command (see **(4)** in **6.6.1 Controlling high-speed system clock**, **(3)** in **6.6.2 Example of controlling internal high-speed oscillation clock**, and **(4)** in **6.6.3 Example of controlling subsystem clock**).

Figure 6-13 Operation of the clock generating circuit when power supply voltage injection (When 2.7 V/1.59V POC mode setup (option byte: LVISTART = 1))



<1> The internal reset signal by the power-on clear (POC) circuit is generated after a power supply injection.

<2> If power supply voltage exceeds 1.59 V (TYP.), reset will be canceled and the oscillation start of the high-speed oscillator will be carried out automatically.

<3> After reset release, after reset processing is performed, CPU carries out a start of operation with high-speed oscillation clock.

<4> X1 clock or XT1 clock should set up an oscillation start by software (see **(1)** in **6.6.1 Controlling high-speed system clock** and **(1)** in **6.6.3 Example of controlling subsystem clock**).

<5> When you change CPU to X1 clock or XT1 clock, set up a change by software after the oscillation stability waiting of a clock (see **(3)** in **6.6.1 Controlling high-speed system clock** and **(3)** in **6.6.3 Example of controlling subsystem clock**).

Note Check the oscillation stable time of X1 clock with an oscillation stable time counter status register (OSTC) when STOP mode release in case the time of reset release (figure 6-13) and a CPU clock are high-speed oscillation clocks . Moreover, when a CPU clock is a high-speed system clock (X1 oscillation), set up the oscillation stable time at the time of STOP mode release by the oscillation stable time selection register (OSTS).

Cautions 1. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) within 1.93 ms, the power supply oscillation stabilization time of 0 to 5.39 ms 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 and EXCLKS pins is used.

remark The clock which is not used as a CPU clock can be suspended by setup of software during microcomputer operation. Moreover, high-speed oscillation clock and a high-speed system clock can suspend a clock by execution of a STOP command (see **(4)** in **6.6.1 Controlling high-speed system clock, (3)** in **6.6.2 Example of controlling internal high-speed oscillation clock, and (4)** in **6.6.3 Example of controlling subsystem clock**).

6.6 Controlling Clock

6.6.1 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 I/O port pins.

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

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

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

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

<1> Setting frequency (OSCCTL register)

Using AMPH, set the gain of the on-chip oscillator according to the frequency to be used.

AMPH[Note]	Operating Frequency Control
0	4 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

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

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

<2> Setting P121/X1 and P122/X2/EXCLK pins and selecting X1 clock or external clock (OSCCTL register)

When EXCLK is cleared to 0 and OSCSEL is set to 1, the mode is switched from port mode to X1 oscillation mode.

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

<3> Controlling oscillation of X1 clock (MOC register)

If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<4> Waiting for the stabilization of the oscillation of X1 clock
Check the OSTC register and wait for the necessary time.
During the wait time, other software processing can be executed with the internal high-speed oscillation clock.

Cautions 1. Do not change the value of EXCLK and OSCSEL while the X1 clock is operating.
2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) or CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)).

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

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

AMPH[Note]	Operating Frequency Control
0	4 MHz ≤ f_{XH} ≤ 10 MHz
1	10 MHz < f_{XH} ≤ 20 MHz

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

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

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

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

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

Cautions 1. Do not change the value of EXCLK and OSCSEL while the external main system clock is operating.
2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 27 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) or 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]
(See **6.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

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

<2> Setting the high-speed system clock as the main system clock (MCM register)

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

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

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

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

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

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

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

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

- Executing the STOP instruction 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, see **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 (PCC and MCM registers)

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

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

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

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

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

Caution Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

6.6.2 Example of controlling internal high-speed oscillation clock

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

(1) When restarting oscillation of the internal high-speed oscillation clock

(2) When using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as peripheral hardware clock

(3) When stopping the internal high-speed oscillation clock

(1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock[Note 1]

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

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

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

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

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

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

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

<1> • Restarting oscillation of the internal high-speed oscillation clock[Note]

(See **6.6.2 (1) Example of setting procedure when restarting internal high-speed oscillation clock**).

• Oscillating the high-speed system clock[Note]

(This setting is required when using the high-speed system clock as the peripheral hardware clock. See **6.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

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

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

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

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

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

(3) Example of setting procedure when stopping the internal high-speed oscillation clock
The internal high-speed oscillation clock can be stopped in the following two ways.

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

(a) To execute a STOP instruction

<1> Setting of peripheral hardware
Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 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 internal high-speed oscillation clock is stopped.

(b) To stop internal high-speed oscillation clock by setting RSTOP to 1

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

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

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

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

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

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

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting RSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.

6.6.3 Example of controlling subsystem clock

The following two types of subsystem clocks are available.

- XT1 clock: Crystal/ceramic resonator is connected across the XT1 and XT2 pins.
- External subsystem clock: External clock is input to the EXCLKS pin.

When the subsystem clock is not used, the XT1/P123 and XT2/EXCLKS/P124 pins can be used as I/O port pins.

Caution The XT1/P123 and XT2/EXCLKS/P124 pins are in the I/O port mode after a reset release.

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

(1) When oscillating XT1 clock
(2) When using external subsystem clock
(3) When using subsystem clock as CPU clock
(4) When stopping subsystem clock

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

<1> Setting XT1 and XT2 pins and selecting operation mode (PCC and OSCCTL registers)

When XTSTART, EXCLKS, and OSCSELS are set as any of the following, the mode is switched from port mode to XT1 oscillation mode.

XTSTART	EXCLKS	OSCSELS	Operation Mode of Subsystem Clock Pin	P123/XT1 Pin	P124/XT2/EXCLKS Pin
0	0	1	XT1 oscillation mode	Crystal/ceramic resonator connection	
1	×	×			

Remark ×: don't care

<2> Waiting for the stabilization of the subsystem clock oscillation

Wait for the oscillation stabilization time of the subsystem clock by software, using a timer function.

Caution Do not change the value of XTSTART, EXCLKS, and OSCSELS while the subsystem clock is operating.

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

<1> Setting XT1 and XT2 pins, selecting XT1 clock/external clock and controlling oscillation (PCC and OSCCTL registers)

When XTSTART is cleared to 0 and EXCLKS and OSCSELS are set to 1, the mode is switched from port mode to external clock input mode. In this case, input the external clock to the EXCLKS/XT2/P124 pins.

XTSTART	EXCLKS	OSCSELS	Operation Mode of Subsystem Clock Pin	P123/XT1 Pin	P124/XT2/ EXCLKS Pin
0	1	1	External clock input mode	I/O port	External clock input

Caution Do not change the value of XTSTART, EXCLKS, and OSCSELS while the subsystem clock is operating.

(3) Example of setting procedure when using the subsystem clock as the CPU clock

<1> Setting subsystem clock oscillation[Note]

(See **6.6.3 (1) Example of setting procedure when oscillating the XT1 clock** and **(2) Example of setting procedure when using the external subsystem clock**.)

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

<2> Switching the CPU clock (PCC register)

When CSS is set to 1, the subsystem clock is supplied to the CPU.

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

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

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

Confirm with CLS and MCS that the CPU is operating on a clock other than the subsystem clock.

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

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

<2> Stopping the subsystem clock (OSCCTL register)

When OSCSELS is cleared to 0, XT1 oscillation is stopped (the input of the external clock is disabled).

Cautions 1. Be sure to confirm that CLS = 0 when clearing OSCSELS to 0. In addition, stop the watch timer if it is operating on the subsystem clock.

2. The subsystem clock oscillation cannot be stopped using the STOP instruction.

6.6.4 Controlling internal low-speed oscillation clock

The internal low-speed oscillation clock is a clock for the watchdog timer. It cannot be used as the CPU clock. With this clock, only the following peripheral hardware can operate.

- Watchdog timer
- 8-bit timer H1 (if f_{RL} is selected as the count clock)

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

- Internal low-speed oscillation clock oscillation cannot be stopped
- Internal low-speed oscillation clock oscillation can be stopped by software

After a reset release, the internal low-speed oscillation clock automatically oscillates.

(1) To stop the internal low-speed oscillation clock (example of setting method)

<1> Setting LSRSTOP to 1 (RCM register)

If LSRSTOP is set to 1, the internal low-speed oscillator oscillation is stopped.

(2) To oscillate the internal low-speed oscillation clock (example of setting method)

<1> Clearing LSRSTOP to 0 (RCM register)

If LSRSTOP is cleared to 0, the internal low-speed oscillation clock is oscillated.

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

6.6.5 Clocks supplied to CPU and peripheral hardware

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

Table 6-3. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting

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

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

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

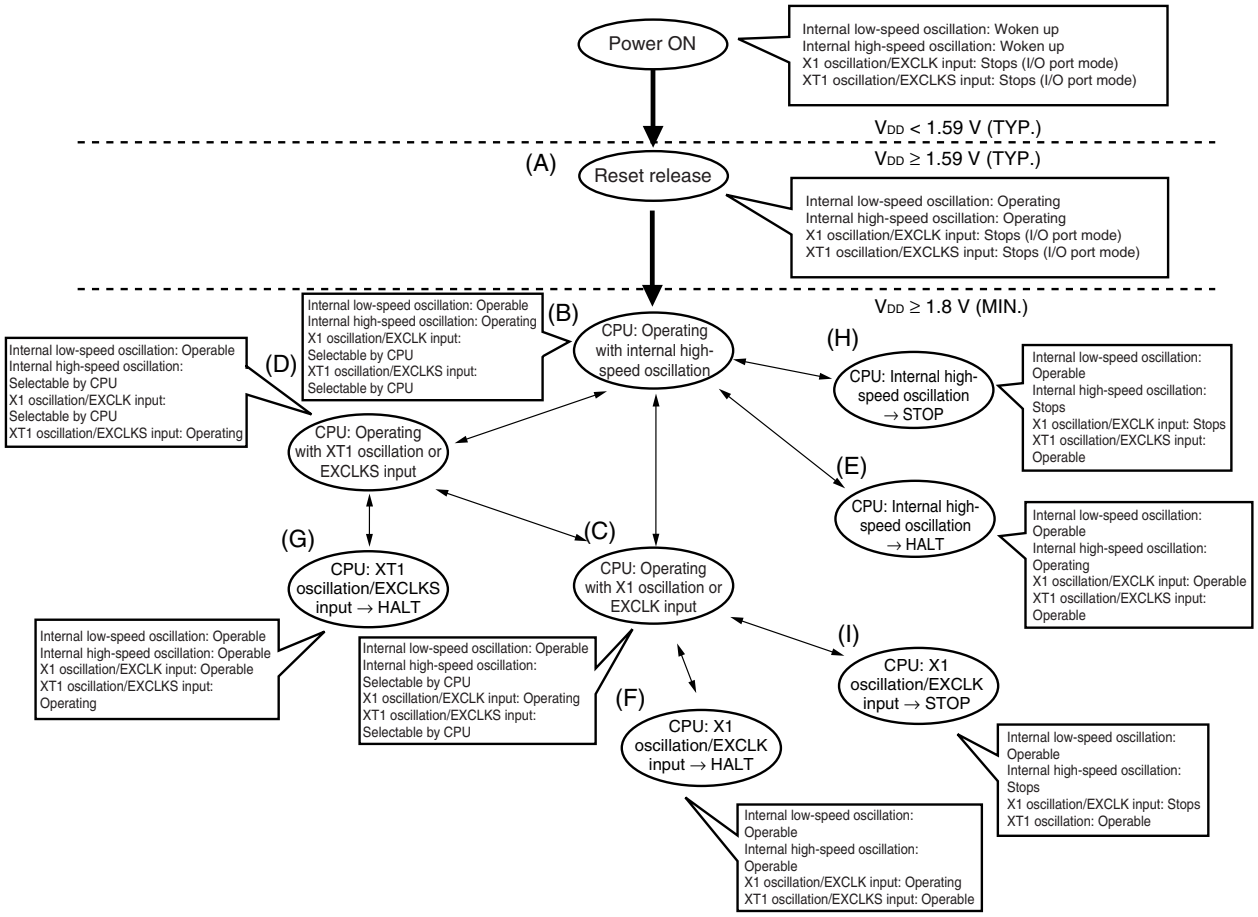
3. MCM0: Bit 0 of MCM

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

6.6.6 CPU clock status transition diagram

Figure 6-14 shows the CPU clock status transition diagram of this product.

Figure 6-14. CPU Clock Status Transition Diagram



Remark In the 2.7 V/1.59 V POC mode (option byte: LVISTART = 1), the CPU clock status changes to (A) in the above figure when the supply voltage exceeds 2.7 V (TYP.), and to (B) after reset processing (11 to 45 μs).

Table 6-4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 6-4. CPU Clock Transition and SFR Register Setting Examples (1/4)

(1) CPU operating with high-speed system clock (C) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(A) → (B) → (C) (X1 clock: less than 10 MHz)	0	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: less than 10 MHz)	0	1	1	0	Must not be checked	1	1
(A) → (B) → (C) (X1 clock: 10 MHz or more)	1	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: 10 MHz or more)	1	1	1	0	Must not be checked	1	1

(2) 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).

(3) CPU operating with subsystem clock (D) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	EXCLKS	OSCSELS	Waiting for Oscillation Stabilization	CSS
(A) → (B) → (D) (XT1 clock)	0	1	Necessary	1
(A) → (B) → (D) (external subsystem clock)	1	1	Unnecessary	1

Remarks 1. (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-14.

2. EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH:
Bits 7 to 4 and 0 of the clock operation mode select register (OSCCTL)

MSTOP: Bit 7 of the main OSC control register (MOC)

XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)

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

Table 6-4. CPU Clock Transition and SFR Register Setting Examples (2/4)

(4) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(B) → (C) (X1 clock: less than 10 MHz)	0	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: less than 10 MHz)	0	1	1	0	Must not be checked	1	1
(B) → (C) (X1 clock: 10 MHz or more)	1	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: 10 MHz or more)	1	1	1	0	Must not be checked	1	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Unnecessary if this register is already set

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	EXCLKS	OSCSELS	Waiting for Oscillation Stabilization	CSS
(B) → (D) (XT1 clock)	0	1	Necessary	1
(B) → (D) (external subsystem clock)	1	1	Unnecessary	1

Unnecessary if the CPU is operating with the subsystem clock

Remarks 1. (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-14.

2. EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH: Bits 7 to 4 and 0 of the clock operation mode select register (OSCCTL)

MSTOP: Bit 7 of the main OSC control register (MOC)

XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)

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

Table 6-4. CPU Clock Transition and SFR Register Setting Examples (3/4)

(6) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	RSTOP	RSTS	MCM0
(C) → (B)	0	Confirm this flag is 1.	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

(7) CPU clock changing from high-speed system clock (C) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	EXCLKS	OSCSELS	Waiting for Oscillation Stabilization	CSS
(C) → (D) (XT1 clock)	0	1	Necessary	1
(C) → (D) (external subsystem clock)	1	1	Unnecessary	1

Unnecessary if the CPU is operating with the subsystem clock

(8) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0	CSS
(D) → (C) (X1 clock: less than 10 MHz)	0	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: less than 10 MHz)	0	1	1	0	Must not be checked	1	1	0
(D) → (C) (X1 clock: 10 MHz or more)	1	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: 10 MHz or more)	1	1	1	0	Must not be checked	1	1	0

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

↑ Unnecessary if this register is already set

Remarks 1. (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-14.

2. EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH: Bits 7 to 4 and 0 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)
RSTS, RSTOP: Bits 7 and 0 of the internal oscillator mode register (RCM)

Table 6-4. CPU Clock Transition and SFR Register Setting Examples (4/4)

(9) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	RSTOP	RSTS	MCM0	CSS
(D) → (B)	0	Confirm this flag is 1.	0	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

↑ Unnecessary if XSEL is 0

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

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

(11) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

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

Remarks 1. (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-14.

2. MCM0: Bit 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)
RSTS, RSTOP: Bits 7 and 0 of the internal oscillator mode register (RCM)

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

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

Table 6-5. 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). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin • MSTOP = 0, OSCSEL = 1, EXCLK = 1	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 1.
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • RSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
Internal high-speed oscillation clock	XT1 clock	Stabilization of XT1 oscillation • XTSTART = 0, EXCLKS = 0, OSCSELS = 1, or XTSTART = 1 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (RSTOP = 1).
X1 clock			X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
Internal high-speed oscillation clock	External subsystem clock	Enabling input of external clock from EXCLKS pin • XTSTART = 0, EXCLKS = 1, OSCSELS = 1	Operating current can be reduced by stopping internal high-speed oscillator (RSTOP = 1).
X1 clock			X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
XT1 clock, external subsystem clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • RSTOP = 0, MCS = 0	XT1 oscillation can be stopped or external subsystem clock input can be disabled (OSCSELS = 0).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time • MCS = 1	• XT1 oscillation can be stopped or external subsystem clock input can be disabled (OSCSELS = 0). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 1 • MCS = 1	• XT1 oscillation can be stopped or external subsystem clock input can be disabled (OSCSELS = 0). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 1.

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

By setting bits 0 to 2 (PCC0 to PCC2) and bit 4 (CSS) of the processor clock control register (PCC), the CPU clock can be switched (between the main system clock and the subsystem clock) and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to PCC; operation continues on the pre-switchover clock for several clocks (see **Table 6-6**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 5 (CLS) of the PCC register.

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

Set Value Before Switchover				Set Value After Switchover																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
0	0	0	0					16 clocks				16 clocks				16 clocks				16 clocks				$2f_{XP}/f_{SUB}$ clocks			
	0	0	1	8 clocks								8 clocks				8 clocks				8 clocks				f_{XP}/f_{SUB} clocks			
	0	1	0	4 clocks				4 clocks								4 clocks				4 clocks				$f_{XP}/2f_{SUB}$ clocks			
	0	1	1	2 clocks				2 clocks				2 clocks								2 clocks				$f_{XP}/4f_{SUB}$ clocks			
	1	0	0	1 clock				1 clock				1 clock				1 clock								$f_{XP}/8f_{SUB}$ clocks			
1	×	×	×	2 clocks				2 clocks				2 clocks				2 clocks				2 clocks							

Caution Selection of the main system clock cycle division factor (PCC0 to PCC2) and switchover from the main system clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously.

Simultaneous setting is possible, however, for selection of the main system clock cycle division factor (PCC0 to PCC2) and switchover from the subsystem clock to the main system clock (changing CSS from 1 to 0).

Remarks 1. The number of clocks listed in Table 6-6 is the number of CPU clocks before switchover.

2. When switching the CPU clock from the main system clock to the subsystem clock, calculate the number of clocks by rounding up to the next clock and discarding the decimal portion, as shown below.

Example When switching CPU clock from $f_{XP}/2$ to $f_{SUB}/2$ (@ oscillation with f_{XP} = 10 MHz, f_{SUB} = 32.768 kHz)

$f_{XP}/f_{SUB} = 10000/32.768 \cong 305.1 \rightarrow 306$ clocks

By setting bit 0 (MCM0) of the main clock mode register (MCM), the main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to MCM0; operation continues on the pre-switchover clock for several clocks (see **Table 6-7**).

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.

Table 6-7. Maximum Time Required for Main System Clock Switchover

Set Value Before Switchover	Set Value After Switchover	
MCM0	MCM0	
	0	1
0		$1 + 2f_{RH}/f_{XH}$ clock
1	$1 + 2f_{XH}/f_{RH}$ clock	

Caution When switching the internal high-speed oscillation clock to the high-speed system clock, bit 2 (XSEL) of MCM must be set to 1 in advance. The value of XSEL can be changed only once after a reset release.

Remarks 1. The number of clocks listed in Table 6-7 is the number of main system clocks before switchover.

2. Calculate the number of clocks in Table 6-7 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_{RH} = 8 MHz, f_{XH} = 10 MHz)

$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2$ clocks

6.6.9 Conditions before clock oscillation is stopped

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

Table 6-8. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
Internal high-speed oscillation clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the internal high-speed oscillation clock)	RSTOP = 1
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
External main system clock		
XT1 clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock)	OSCSELS = 0
External subsystem clock		

CHAPTER 7 16-BIT TIMER/EVENT COUNTERS 00 TO 03

The 78K0/FE2 incorporates 16-bit timer/event counters 00 to 03.

7.1 Functions of 16-Bit Timer/Event Counters 00 to 03

16-bit timer/event counters 00 to 03 have the following functions.

- Interval timer
- PPG output
- Pulse width measurement
- External event counter
- Square-wave output
- One-shot pulse output

(1) Interval timer

16-bit timer/event counters 00 to 03 generate an interrupt request at the preset time interval.

(2) PPG output

16-bit timer/event counters 00 to 03 can output a rectangular wave whose frequency and output pulse width can be set freely.

(3) Pulse width measurement

16-bit timer/event counters 00 to 03 can measure the pulse width of an externally input signal.

(4) External event counter

16-bit timer/event counters 00 to 03 can measure the number of pulses of an externally input signal.

(5) Square-wave output

16-bit timer/event counters 00 to 03 can output a square wave with any selected frequency.

(6) One-shot pulse output

16-bit timer event counters 00 to 03 can output a one-shot pulse whose output pulse width can be set freely.

7.2 Configuration of 16-Bit Timer/Event Counters 00 to 03

16-bit timer/event counters 00 to 03 include the following hardware.

Table 7-1. Configuration of 16-Bit Timer/Event Counters 00 to 03

Item	Configuration
Timer counter	16 bits (TM0n)
Register	16-bit timer capture/compare register: 16 bits (CR00n, CR01n)
Timer input	TI00n, TI01n
Timer output	TO0n, output controller
Control registers	16-bit timer mode control register 0n (TMC0n) 16-bit timer capture/compare control register 0n (CRC0n) 16-bit timer output control register 0n (TOC0n) Prescaler mode register 0n (PRM0n) Port mode register 0, 3, 13 (PM0, PM3, PM13) Port register 0, 3, 13 (P0, P3, P13)

Remark n = 0 to 3

Figures 7-1 to 7-4 show the block diagrams.

Figure 7-1. Block Diagram of 16-Bit Timer/Event Counter 00

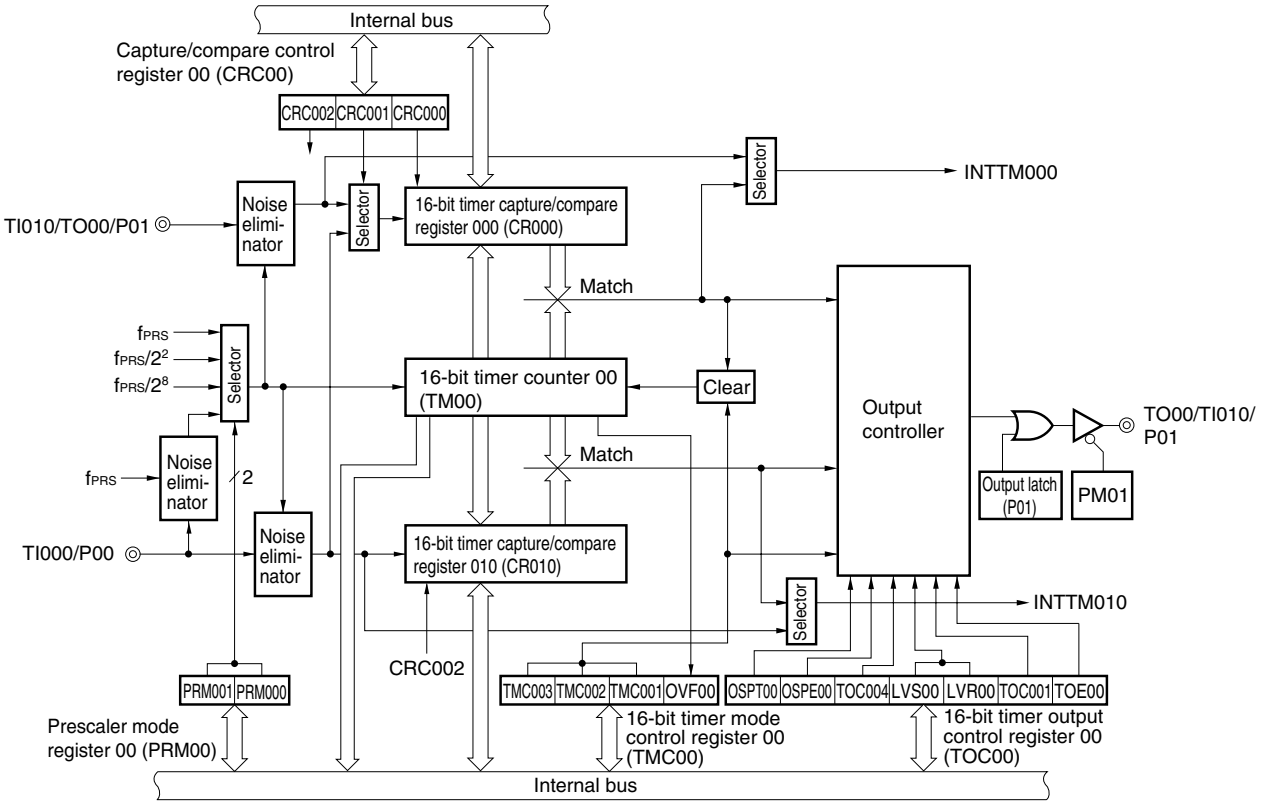


Figure 7-2. Block Diagram of 16-Bit Timer/Event Counter 01

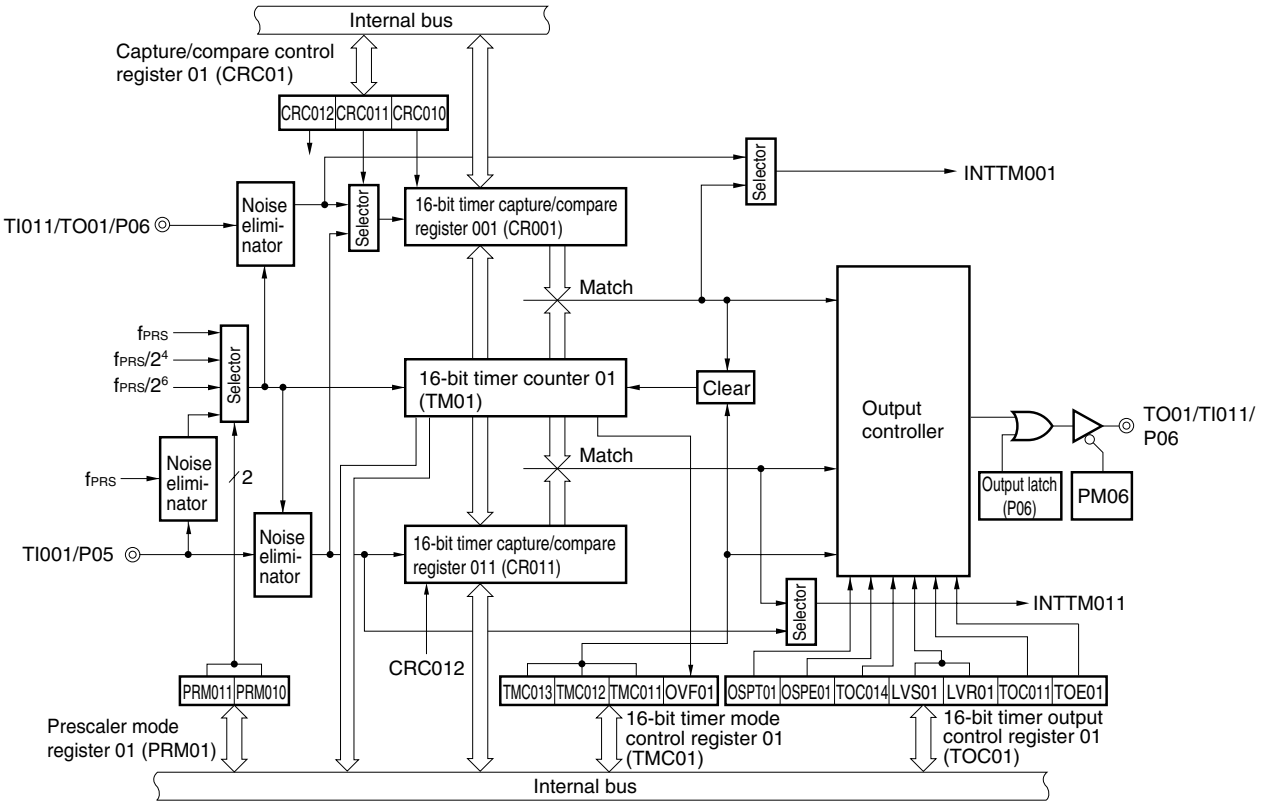


Figure 7-3. Block Diagram of 16-Bit Timer/Event Counter 02

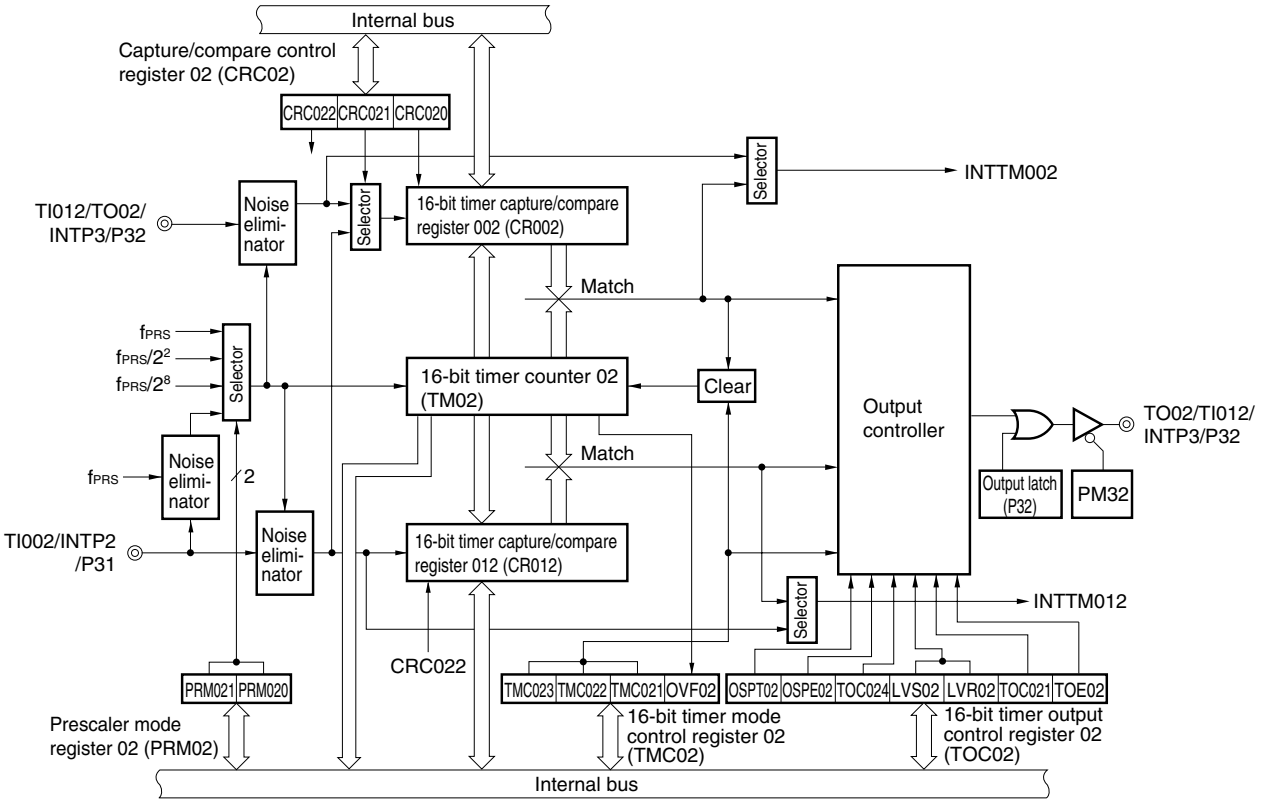
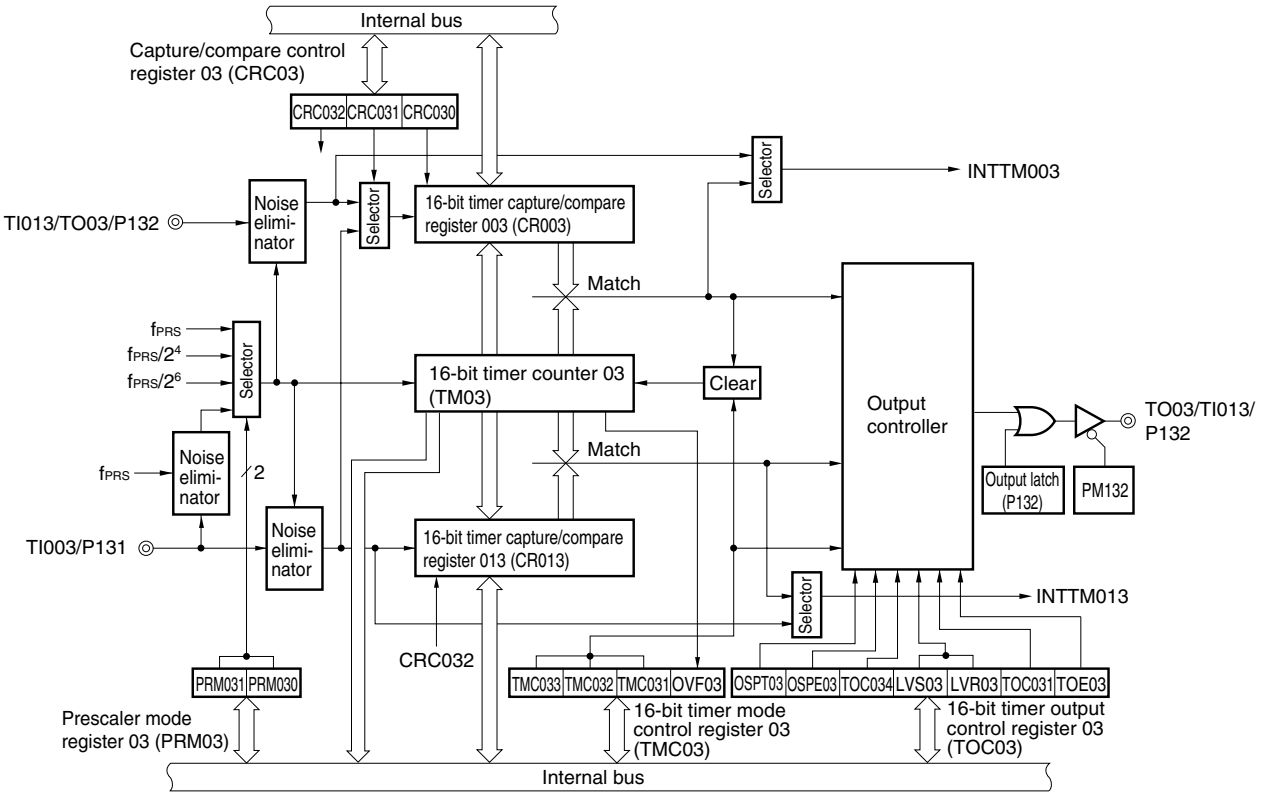


Figure 7-4. Block Diagram of 16-Bit Timer/Event Counter 03



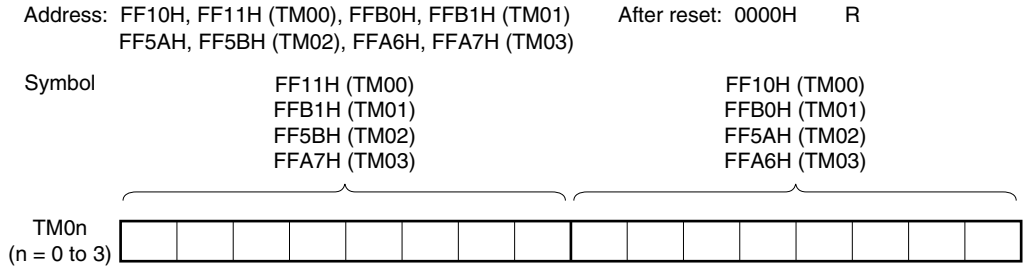
(1) 16-bit timer counter 0n (TM0n)

TM0n is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the count clock.

If the count value is read during operation, then input of the count clock is temporarily stopped, and the count value at that point is read.

Figure 7-5. Format of 16-Bit Timer Counter 0n (TM0n)



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

<1> At Reset signal generation

<2> If TMC0n3 and TMC0n2 are cleared

<3> If the valid edge of the TI00n pin is input in the mode in which clear & start occurs when inputting the valid edge of the TI00n pin

<4> If TM0n and CR00n match in the mode in which clear & start occurs on a match of TM0n and CR00n

<5> OSPT0n is set to 1 in one-shot pulse output mode or the valid edge is input to the TI00n pin

Caution Even if TM0n is read, the value is not captured by CR01n.

(2) 16-bit timer capture/compare register 00n (CR00n)

CR00n is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC0n0) of capture/compare control register 0n (CRC0n).

CR00n can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-6. Format of 16-Bit Timer Capture/Compare Register 00n (CR00n)

Address: FF12H, FF13H (CR000), FFB2H, FFB3H (CR001) FF6CH, FF6DH (CR002), FFA8H, FFA9H (CR003) After reset: 0000H R/W

Symbol	FF13H (CR000) FFB3H (CR001) FF6DH (CR002) FFA9H (CR003)								FF12H (CR000) FFB2H (CR001) FF6CH (CR002) FFA8H (CR003)							
CR00n (n = 0 to 3)																

- **When CR00n is used as a compare register**

  The value set in CR00n is constantly compared with 16-bit timer counter 0n (TM0n) count value, and an interrupt request (INTTM00n) is generated if they match. The set value is held until CR00n is rewritten.

  Caution CR00n does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

- **When CR00n is used as a capture register**

  It is possible to select the valid edge of the TI00n pin or the TI01n pin as the capture trigger. The TI00n or TI01n pin valid edge is set using prescaler mode register 0n (PRM0n) (see **Table 7-2**).

Table 7-2. CR00n Capture Trigger and Valid Edges of TI00n and TI01n Pins

(1) TI00n pin valid edge selected as capture trigger (CRC0n1 = 1, CRC0n0 = 1)

CR00n Capture Trigger	TI00n Pin Valid Edge		
		ES0n1	ES0n0
Falling edge	Rising edge	0	1
Rising edge	Falling edge	0	0
No capture operation	Both rising and falling edges	1	1

(2) TI01n pin valid edge selected as capture trigger (CRC0n1 = 0, CRC0n0 = 1)

CR00n Capture Trigger	TI01n Pin Valid Edge		
		ES1n1	ES1n0
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Cautions 1. Set a value other than 0000H in CR00n in the mode in which clear & start occurs on a match of TM0n and CR00n.

2. If CR00n is cleared to 0000H in the free-running mode and in the clear mode using the valid edge of the TI00n pin, an interrupt request (INTTM00n) is generated when the value of CR00n changes from 0000H to 0001H following TM0n overflow (FFFFH). In addition, INTTM00n is generated after a match between TM0n and CR00n, after detecting the valid edge of the TI01n pin, and the timer is cleared by a one-shot trigger.

3. When P01 or P06 is used as the valid edge input of the TI01n pin, it cannot be used as the timer output (TO0n). Moreover, when P01 or P06 is used as TO0n, it cannot be used as the valid edge input of the TI01n pin.

4. When CR00n is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

5. Do not rewrite CR00n during TM0n operation.

Remarks 1. Setting ES0n1, ES0n0 = 1, 0 and ES1n1, ES1n0 = 1, 0 is prohibited.

2. ES0n1, ES0n0: Bits 5 and 4 of prescaler mode register 0n (PRM0n)
ES1n1, ES1n0: Bits 7 and 6 of prescaler mode register 0n (PRM0n)
CRC0n1, CRC0n0: Bits 1 and 0 of capture/compare control register 0n (CRC0n)

3. n = 0 to 3

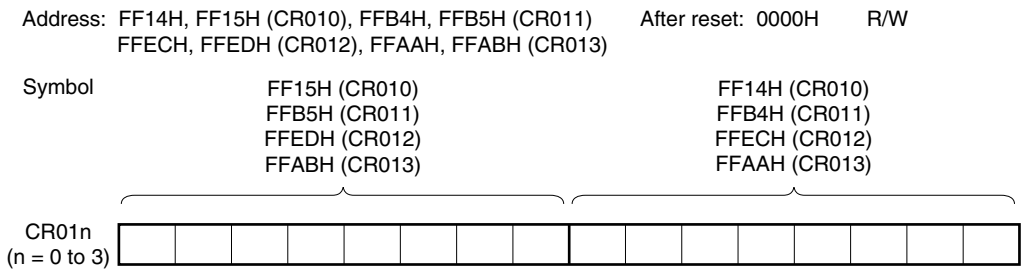
(3) 16-bit timer capture/compare register 01n (CR01n)

CR01n is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC0n2) of capture/compare control register 0n (CRC0n).

CR01n can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-7. Format of 16-Bit Timer Capture/Compare Register 01n (CR01n)



- **When CR01n is used as a compare register**

  The value set in the CR01n is constantly compared with 16-bit timer counter 0n (TM0n) count value, and an interrupt request (INTTM01n) is generated if they match. The set value is held until CR01n is rewritten.

- **When CR01n is used as a capture register**

  It is possible to select the valid edge of the TI00n pin as the capture trigger. The TI00n pin valid edge is set by prescaler mode register 0n (PRM0n) (see **Table 7-3**).

Table 7-3. CR01n Capture Trigger and Valid Edge of TI00n Pin (CRC0n2 = 1)

CR01n Capture Trigger	TI00n Pin Valid Edge		
		ES0n1	ES0n0
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Cautions 1. If the CR01n register is cleared to 0000H, an interrupt request (INTTM01n) is generated when the value of CR01n changes from 0000H to 0001H following TM0n overflow (FFFFH). In addition, INTTM01n is generated after a match between TM0n and CR01n, after detecting the valid edge of the TI00n pin, and the timer is cleared by a one-shot trigger.

2. When CR01n is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

3. CR01n can be rewritten during TM0n operation. For details, see Caution 2 in Figure 7-33 PPG Output Operation Timing.

Remarks 1. Setting ES0n1, ES0n0 = 1, 0 is prohibited.

2. ES0n1, ES0n0: Bits 5 and 4 of prescaler mode register 0n (PRM0n)
CRC0n2: Bit 2 of capture/compare control register 0n (CRC0n)

3. n = 0 to 3

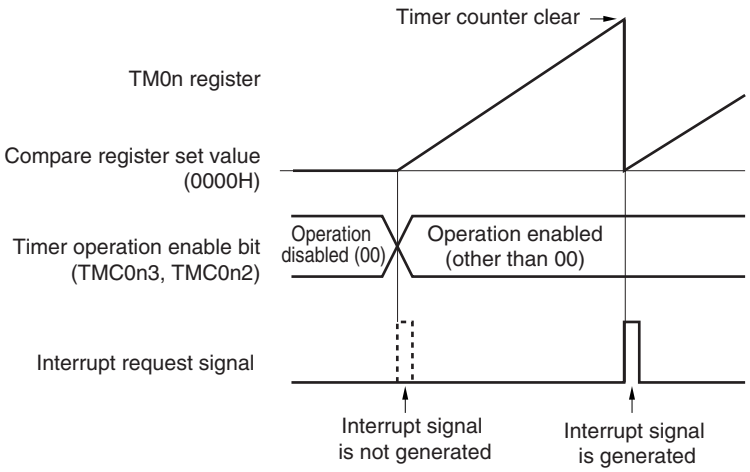
(4) Setting range when CR00n or CR01n is used as a compare register

When CR00n or CR01n is used as a compare register, set it as shown below.

Operation	CR00n Register Setting Range	CR01n Register Setting Range
Operation as interval timer	0000H < N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH Normally, this setting is not used. Mask the match interrupt signal (INTTM01n).
Operation as square-wave output		
Operation as external event counter		
Operation in the clear & start mode entered by TI00n pin valid edge input	0000H[Note] ≤ N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH
Operation as free-running timer		
Operation as PPG output	M < N ≤ FFFFH	0000H[Note] ≤ M < N
Operation as one-shot pulse output	0000H[Note] ≤ N ≤ FFFFH (N ≠ M)	0000H[Note] ≤ M ≤ FFFFH (M ≠ N)

Note When 0000H is set, a match interrupt immediately after the timer operation does not occur and timer output is not changed, and the first match timing is as follows. A match interrupt occurs at the timing when the timer counter (TM0n register) is changed from 0000H to 0001H.

- When the timer counter is cleared due to overflow
- When the timer counter is cleared due to TI00n pin valid edge (when clear & start mode is entered by TI00n pin valid edge input)
- When the timer counter is cleared due to compare match (when clear & start mode is entered by match between TM0n and CR00n (CR00n = other than 0000H, CR01n = 0000H))

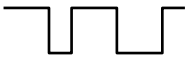
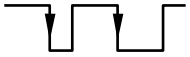
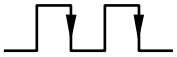
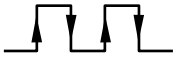
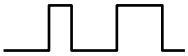
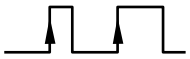
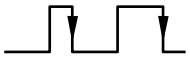
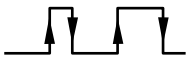


Remarks 1. N: CR00n register set value, M: CR01n register set value

2. For details of TMC0n3 and TMC0n2, see **7.3 (1) 16-bit timer mode control register 0n (TMC0n)**.

3. n = 0 to 3

Table 7-4. Capture Operation of CR00n and CR01n

External Input Signal / Capture Operation	TI00n Pin Input		TI01n Pin Input	
Capture operation of CR00n	CRC0n1 = 1 TI00n pin input (reverse phase) 	Set values of ES0n1 and ES0n0 Position of edge to be captured	CRC0n1 bit = 0 TI01n pin input 	Set values of ES1n1 and ES1n0 Position of edge to be captured
		01: Rising 		01: Rising 
		00: Falling 		00: Falling 
		11: Both edges (cannot be captured)		11: Both edges 
	Interrupt signal	INTTM00n signal is not generated even if value is captured.	Interrupt signal	INTTM00n signal is generated each time value is captured.
Capture operation of CR01n	TI00n pin input[Note] 	Set values of ES0n1 and ES0n0 Position of edge to be captured		
		01: Rising 		
		00: Falling 		
		11: Both edges 		
	Interrupt signal	INTTM01n signal is generated each time value is captured.		

Note The capture operation of CR01n is not affected by the setting of the CRC0n1 bit.

Caution To capture the count value of the TM0n register to the CR00n register by using the phase reverse to that input to the TI00n pin, the interrupt request signal (INTTM00n) is not generated after the value has been captured. If the valid edge is detected on the TI01n pin during this operation, the capture operation is not performed but the INTTM00n signal is generated as an external interrupt signal. To not use the external interrupt, mask the INTTM00n signal.

Remarks 1. CRC0n1: See **7.3 (2) Capture/compare control register 0n (CRC0n)**.
ES1n1, ES1n0, ES0n1, ES0n0: See **7.3 (4) Prescaler mode register 0n (PRM0n)**.

2. n = 0 to 3

7.3 Registers Controlling 16-Bit Timer/Event Counters 00 to 03

The following six registers are used to control 16-bit timer/event counters 00 to 03.

- 16-bit timer mode control register 0n (TMC0n)
- Capture/compare control register 0n (CRC0n)
- 16-bit timer output control register 0n (TOC0n)
- Prescaler mode register 0n (PRM0n)
- Port mode register 0, 3, 13 (PM0, PM3, PM13)
- Port register 0, 3, 13 (P0, P3, P13)

(1) 16-bit timer mode control register 0n (TMC0n)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 0n (TM0n) clear mode, and output timing, and detects an overflow.

Rewriting TMC0n is prohibited during operation (when TMC0n3 and TMC0n2 = other than 00). However, it can be changed when TMC0n3 and TMC0n2 are cleared to 00 (stopping operation) and when OVF0n is cleared to 0.

TMC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TMC0n to 00H.

Caution 16-bit timer counter 0n (TM0n) starts operation at the moment TMC0n2 and TMC0n3 are set to values other than 0, 0 (operation stop mode), respectively. Set TMC0n2 and TMC0n3 to 0, 0 to stop the operation.

Remark n = 0 to 3

Figure 7-8. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FFBAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	Operation enable of 16-bit timer/event counter 00
0	0	Disables 16-bit timer/event counter 00 operation. Stops supplying operating clock. Clears 16-bit timer counter 00 (TM00).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI000 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM00 and CR000

TMC001	Condition to reverse timer output (TO00)
0	• Match between TM00 and CR000 or match between TM00 and CR010
1	• Match between TM00 and CR000 or match between TM00 and CR010 • Trigger input of TI000 pin valid edge

OVF00	TM00 overflow flag
Clear (0)	Clears OVF00 to 0 or TMC003 and TMC002 = 00
Set (1)	Overflow occurs.
OVF00 is set to 1 when the value of TM00 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI000 pin valid edge input, and clear & start mode entered upon a match between TM00 and CR000). It can also be set to 1 by writing 1 to OVF00.	

Note The TI000 pin valid edge is set by bits 5 and 4 (ES001, ES000) of prescaler mode register 00 (PRM00).

Remark TO00: 16-bit timer/event counter 00 output pin
TI000: 16-bit timer/event counter 00 input pin
TM00: 16-bit timer counter 00
CR000: 16-bit timer capture/compare register 000
CR010: 16-bit timer capture/compare register 010

Figure 7-9. Format of 16-Bit Timer Mode Control Register 01 (TMC01)

Address: FFB6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC01	0	0	0	0	TMC013	TMC012	TMC011	OVF01

TMC013	TMC012	Operation enable of 16-bit timer/event counter 01
0	0	Disables 16-bit timer/event counter 01 operation. Stops supplying operating clock. Clears 16-bit timer counter 01 (TM01).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI001 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM01 and CR001

TMC011	Condition to reverse timer output (TO01)
0	• Match between TM01 and CR001 or match between TM01 and CR011
1	• Match between TM01 and CR001 or match between TM01 and CR011 • Trigger input of TI001 pin valid edge

OVF01	TM01 overflow flag
Clear (0)	Clears OVF01 to 0 or TMC013 and TMC012 = 00
Set (1)	Overflow occurs.
OVF01 is set to 1 when the value of TM01 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI001 pin valid edge input, and clear & start mode entered upon a match between TM01 and CR001). It can also be set to 1 by writing 1 to OVF01.	

Note The TI001 pin valid edge is set by bits 5 and 4 (ES011, ES010) of prescaler mode register 01 (PRM01).

Remark TO01: 16-bit timer/event counter 01 output pin
TI001: 16-bit timer/event counter 01 input pin
TM01: 16-bit timer counter 01
CR001: 16-bit timer capture/compare register 001
CR011: 16-bit timer capture/compare register 011

Figure 7-10. Format of 16-Bit Timer Mode Control Register 02 (TMC02)

Address: FF54H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC02	0	0	0	0	TMC023	TMC022	TMC021	OVF02

TMC023	TMC022	Operation enable of 16-bit timer/event counter 01
0	0	Disables 16-bit timer/event counter 02 operation. Stops supplying operating clock. Clears 16-bit timer counter 02 (TM02).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI002 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM02 and CR002

TMC021	Condition to reverse timer output (TO02)
0	• Match between TM02 and CR002 or match between TM02 and CR012
1	• Match between TM02 and CR002 or match between TM02 and CR012 • Trigger input of TI002 pin valid edge

OVF02	TM02 overflow flag
Clear (0)	Clears OVF02 to 0 or TMC023 and TMC022 = 00
Set (1)	Overflow occurs.
OVF02 is set to 1 when the value of TM02 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI002 pin valid edge input, and clear & start mode entered upon a match between TM02 and CR002). It can also be set to 1 by writing 1 to OVF02.	

Note The TI002 pin valid edge is set by bits 5 and 4 (ES021, ES020) of prescaler mode register 02 (PRM02).

Remark TO02: 16-bit timer/event counter 02 output pin
TI002: 16-bit timer/event counter 02 input pin
TM02: 16-bit timer counter 02
CR002: 16-bit timer capture/compare register 002
CR012: 16-bit timer capture/compare register 012

Figure 7-11. Format of 16-Bit Timer Mode Control Register 03 (TMC03)

Address: FFADH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC03	0	0	0	0	TMC033	TMC032	TMC031	OVF03

TMC033	TMC032	Operation enable of 16-bit timer/event counter 03
0	0	Disables 16-bit timer/event counter 03 operation. Stops supplying operating clock. Clears 16-bit timer counter 03 (TM03).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI003 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM03 and CR003

TMC031	Condition to reverse timer output (TO03)
0	• Match between TM03 and CR003 or match between TM03 and CR013
1	• Match between TM03 and CR003 or match between TM03 and CR013 • Trigger input of TI003 pin valid edge

OVF03	TM03 overflow flag
Clear (0)	Clears OVF03 to 0 or TMC033 and TMC032 = 00
Set (1)	Overflow occurs.
OVF03 is set to 1 when the value of TM03 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI003 pin valid edge input, and clear & start mode entered upon a match between TM03 and CR003). It can also be set to 1 by writing 1 to OVF03.	

Note The TI003 pin valid edge is set by bits 5 and 4 (ES031, ES030) of prescaler mode register 03 (PRM03).

Remark TO03: 16-bit timer/event counter 03 output pin
TI003: 16-bit timer/event counter 03 input pin
TM03: 16-bit timer counter 03
CR003: 16-bit timer capture/compare register 003
CR013: 16-bit timer capture/compare register 013

(2) Capture/compare control register 0n (CRC0n)

CRC0n is the register that controls the operation of CR00n and CR01n.

Changing the value of CRC0n is prohibited during operation (when TMC0n3 and TMC0n2 = other than 00).

CRC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CRC0n to 00H.

Figure 7-12. Format of Capture/Compare Control Register 00 (CRC00)

Address: FFBCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010 pin
1	Captures on valid edge of TI000 pin by reverse phase[Note]
The valid edge of the TI010 and TI000 pin is set by PRM00. If ES001 and ES000 are set to 11 (both edges) when CRC001 is 1, the valid edge of the TI000 pin cannot be detected.	

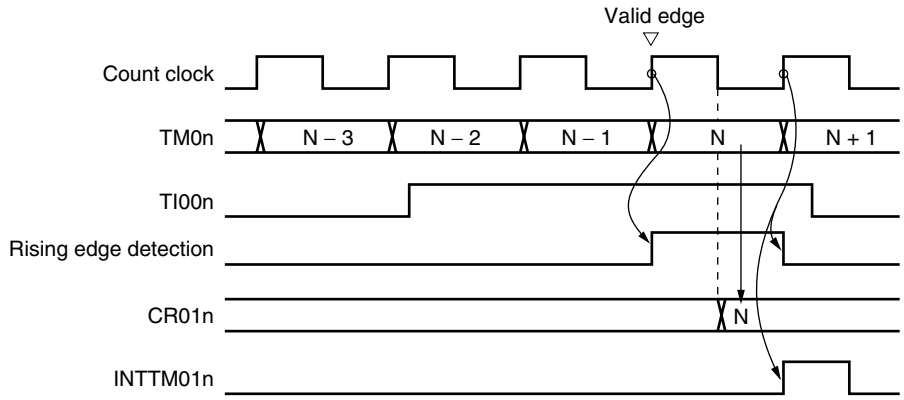
CRC000	CR000 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC003 and TMC002 are set to 11 (clear & start mode entered upon a match between TM00 and CR000), be sure to set CRC000 to 0.	

Note When the valid edge is detected from the TI010 pin, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

Remark n = 0 to 3

Figure 7-13. Example of CR01n Capture Operation (When Rising Edge Is Specified)



Remark n = 0 to 3

Figure 7-14. Format of Capture/Compare Control Register 01 (CRC01)

Address: FFB8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC01	0	0	0	0	0	CRC012	CRC011	CRC010

CRC012	CR011 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC011	CR001 capture trigger selection
0	Captures on valid edge of TI011 pin
1	Captures on valid edge of TI001 pin by reverse phase[Note]
The valid edge of the TI011 and TI001 pin is set by PRM01. If ES011 and ES010 are set to 11 (both edges) when CRC011 is 1, the valid edge of the TI001 pin cannot be detected.	

CRC010	CR001 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC013 and TMC012 are set to 11 (clear & start mode entered upon a match between TM01 and CR001), be sure to set CRC010 to 0.	

Note When the valid edge is detected from the TI011 pin, the capture operation is not performed but the INTTM001 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 01 (PRM01) (see Figure 7-13 Example of CR01n Capture Operation (When Rising Edge Is Specified)).

Figure 7-15. Format of Capture/Compare Control Register 02 (CRC02)

Address: FF5CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC02	0	0	0	0	0	CRC022	CRC021	CRC020

CRC022	CR012 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC021	CR002 capture trigger selection
0	Captures on valid edge of TI012 pin
1	Captures on valid edge of TI002 pin by reverse phase[Note]
The valid edge of the TI012 and TI002 pin is set by PRM02. If ES021 and ES020 are set to 11 (both edges) when CRC021 is 1, the valid edge of the TI002 pin cannot be detected.	

CRC020	CR002 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC023 and TMC022 are set to 11 (clear & start mode entered upon a match between TM02 and CR002), be sure to set CRC020 to 0.	

Note When the valid edge is detected from the TI012 pin, the capture operation is not performed but the INTTM002 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 02 (PRM02) (see Figure 7-13 Example of CR01n Capture Operation (When Rising Edge Is Specified)).

Figure 7-16. Format of Capture/Compare Control Register 03 (CRC03)

Address: FF52H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC03	0	0	0	0	0	CRC032	CRC031	CRC030

CRC032	CR013 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC031	CR003 capture trigger selection
0	Captures on valid edge of TI013 pin
1	Captures on valid edge of TI003 pin by reverse phase[Note]
The valid edge of the TI013 and TI003 pin is set by PRM03. If ES031 and ES030 are set to 11 (both edges) when CRC031 is 1, the valid edge of the TI003 pin cannot be detected.	

CRC030	CR003 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC033 and TMC032 are set to 11 (clear & start mode entered upon a match between TM03 and CR003), be sure to set CRC030 to 0.	

Note When the valid edge is detected from the TI013 pin, the capture operation is not performed but the INTTM003 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 03 (PRM03) (see Figure 7-13 Example of CR01n Capture Operation (When Rising Edge Is Specified)).

(3) 16-bit timer output control register 0n (TOC0n)

TOC0n is an 8-bit register that controls the TO0n pin output.

TOC0n can be rewritten while only OSPT0n is operating (when TMC0n3 and TMC0n2 = other than 00). Rewriting the other bits is prohibited during operation.

However, TOC0n4 can be rewritten during timer operation as a means to rewrite CR01n (see **7.5.1 Rewriting CR01n during TM0n operation**).

TOC0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TOC0n to 00H.

Caution Be sure to set TOC0n using the following procedure.

<1> Set TOC0n4 and TOC0n1 to 1.

<2> Set only TOE0n to 1.

<3> Set either of LVS0n or LVR0n to 1.

Remark n = 0 to 3

Figure 7-17. 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 pin 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 pin output status
0	0	No change
0	1	Initial value of TO00 pin output is low level (TO00 pin output is cleared to 0).
1	0	Initial value of TO00 pin output is high level (TO00 pin output is set to 1).
1	1	Setting prohibited

- LVS00 and LVR00 can be used to set the initial value of the output level of the TO00 pin. 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 output level of the TO00 pin can be set. Even if these bits are cleared to 0, output of the TO00 pin is not affected.
- The values of LVS00 and LVR00 are always 0 when they are read.
- For how to set LVS00 and LVR00, see **7.5.2 Setting LVS0n and LVR0n**.

TOC001	TO00 pin 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 pin output control
0	Disables output (TO00 pin output fixed to low level)
1	Enables output

Figure 7-18. Format of 16-Bit Timer Output Control Register 01 (TOC01)

Address: FFB9H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC01	0	OSPT01	OSPE01	TOC014	LVS01	LVR01	TOC011	TOE01

OSPT01	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always 0 when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM01 is cleared and started.	

OSPE01	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI001 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM01 and CR001.	

TOC014	TO01 pin output control on match between CR011 and TM01
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM011) is generated even when TOC014 = 0.	

LVS01	LVR01	Setting of TO01 pin output status
0	0	No change
0	1	Initial value of TO01 pin output is low level (TO01 pin output is cleared to 0).
1	0	Initial value of TO01 pin output is high level (TO01 pin output is set to 1).
1	1	Setting prohibited

- LVS01 and LVR01 can be used to set the initial value of the output level of the TO01 pin. If the initial value does not have to be set, leave LVS01 and LVR01 as 00.
- Be sure to set LVS01 and LVR01 when TOE01 = 1.
  LVS01, LVR01, and TOE01 being simultaneously set to 1 is prohibited.
- LVS01 and LVR01 are trigger bits. By setting these bits to 1, the initial value of the output level of the TO01 pin can be set. Even if these bits are cleared to 0, output of the TO01 pin is not affected.
- The values of LVS01 and LVR01 are always 0 when they are read.
- For how to set LVS01 and LVR01, see **7.5.2 Setting LVS0n and LVR0n**.

TOC011	TO01 pin output control on match between CR001 and TM01
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM001) is generated even when TOC011 = 0.	

TOE01	TO01 pin output control
0	Disables output (TO01 pin output is fixed to low level)
1	Enables output

Figure 7-19. Format of 16-Bit Timer Output Control Register 02 (TOC02)

Address: FFA5H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC02	0	OSPT02	OSPE02	TOC024	LVS02	LVR02	TOC021	TOE02

OSPT02	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always 0 when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM02 is cleared and started.	

OSPE02	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI002 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM02 and CR002.	

TOC024	TO02 pin output control on match between CR012 and TM02
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM012) is generated even when TOC024 = 0.	

LVS02	LVR02	Setting of TO02 pin output status
0	0	No change
0	1	Initial value of TO02 pin output is low level (TO02 pin output is cleared to 0).
1	0	Initial value of TO02 pin output is high level (TO02 pin output is set to 1).
1	1	Setting prohibited

- LVS02 and LVR02 can be used to set the initial value of the output level of the TO02 pin. If the initial value does not have to be set, leave LVS02 and LVR02 as 00.
- Be sure to set LVS02 and LVR02 when TOE02 = 1.
  LVS02, LVR02, and TOE02 being simultaneously set to 1 is prohibited.
- LVS02 and LVR02 are trigger bits. By setting these bits to 1, the initial value of the output level of the TO02 pin can be set. Even if these bits are cleared to 0, output of the TO02 pin is not affected.
- The values of LVS02 and LVR02 are always 0 when they are read.
- For how to set LVS02 and LVR02, see **7.5.2 Setting LVS0n and LVR0n**.

TOC021	TO02 pin output control on match between CR002 and TM02
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM002) is generated even when TOC012 = 0.	

TOE02	TO02 pin output control
0	Disables output (TO02 pin output is fixed to low level)
1	Enables output

Figure 7-20. Format of 16-Bit Timer Output Control Register 03 (TOC03)

Address: FFF9H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC03	0	OSPT03	OSPE03	TOC034	LVS03	LVR03	TOC031	TOE03

OSPT03	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, TM03 is cleared and started.	

OSPE03	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 TI003 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM03 and CR003.	

TOC034	TO03 pin output control on match between CR013 and TM03
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM013) is generated even when TOC034 = 0.	

LVS03	LVR03	Setting of TO03 pin output status
0	0	No change
0	1	Initial value of TO03 pin output is low level (TO03 pin output is cleared to 0).
1	0	Initial value of TO03 pin output is high level (TO03 pin output is set to 1).
1	1	Setting prohibited

- LVS03 and LVR03 can be used to set the initial value of the output level of the TO03 pin. If the initial value does not have to be set, leave LVS03 and LVR03 as 00.
- Be sure to set LVS03 and LVR03 when TOE03 = 1.
  LVS03, LVR03, and TOE03 being simultaneously set to 1 is prohibited.
- LVS03 and LVR03 are trigger bits. By setting these bits to 1, the initial value of the output level of the TO03 pin can be set. Even if these bits are cleared to 0, output of the TO03 pin is not affected.
- The values of LVS03 and LVR03 are always 0 when they are read.
- For how to set LVS03 and LVR03, see **7.5.2 Setting LVS0n and LVR0n**.

TOC013	TO03 pin output control on match between CR003 and TM03
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM003) is generated even when TOC013 = 0.	

TOE03	TO03 pin output control
0	Disables output (TO03 pin output is fixed to low level)
1	Enables output

(4) Prescaler mode register 0n (PRM0n)

PRM0n is the register that sets the TM0n count clock and TI00n and TI01n pin input valid edges.

Rewriting PRM0n is prohibited during operation (when TMC0n3 and TMC0n2 = other than 00).

PRM0n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears PRM0n to 00H.

Cautions 1. Do not apply the following setting when setting the PRM0n1 and PRM0n0 bits to 11 (to specify the valid edge of the TI00n pin as a count clock).

- **Clear & start mode entered by the TI00n pin valid edge**
- **Setting the TI00n pin as a capture trigger**

2. If the operation of the 16-bit timer/event counter 0n is enabled when the TI00n or TI01n pin is at high level and when the valid edge of the TI00n or TI01n pin is specified to be the rising edge or both edges, the high level of the TI00n or TI01n pin is detected as a rising edge. Note this when the TI00n or TI01n pin is pulled up. However, the rising edge is not detected when the timer operation has been once stopped and then is enabled again.

3. The valid edge of TI010 and timer output (TO00) cannot be used for the P01 pin at the same time, and the valid edge of TI011 and timer output (TO01) cannot be used for the P06 pin at the same time. Select either of the functions.

Remark n = 0 to 3

Figure 7-21. Format of Prescaler Mode Register 00 (PRM00)

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES001	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM001	PRM000	Count clock selection				
			f_{PRS} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}	4 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^2$	1 MHz	1.25 MHz	2.5 MHz	5 MHz
1	0	$f_{PRS}/2^8$	15.62 kHz	19.53 kHz	39.06 kHz	78.12 kHz
1	1	TI000 valid edge[Note]				

Note The external clock requires a pulse two cycles longer than internal clock (f_{PRS}).

Cautions 1. Always set data to PRM00 after stopping the timer operation.

2. If the valid edge of the TI000 pin is to be set for the count clock, do not set the clear & start mode using the valid edge of the TI000 pin and the capture trigger.

3. If the TI000 or TI010 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI000 pin or TI010 pin to enable the operation of 16-bit timer counter 00 (TM00). Care is therefore required when pulling up the TI000 or TI010 pin. However, when re-enabling operation for TI000 pin or TI010 pin are high level after the operation has been stopped, the rising edge is not detected.

4. When TI010 pin is used valid edge, it cannot be used as the timer output (TO00) to P01, and when TO00 is used, it cannot be used to the TI010 pin valid edge.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 7-22. Format of Prescaler Mode Register 01 (PRM01)

Address: FFB7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM01	ES111	ES110	ES011	ES010	0	0	PRM011	PRM010

ES111	ES110	TI011 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES011	ES010	TI001 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM011	PRM010	Count clock selection				
			f_{PRS} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}	4 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^4$	250 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	$f_{PRS}/2^6$	62.5 kHz	78.125 kHz	156.25 kHz	312.5 kHz
1	1	TI001 valid edge[Note]				

Note The external clock requires a pulse two cycles longer than internal clock (f_{PRS}).

Cautions 1. Always set data to PRM01 after stopping the timer operation.

2. If the valid edge of the TI001 pin is to be set for the count clock, do not set the clear & start mode using the valid edge of the TI001 pin and the capture trigger.

3. If the TI001 or TI011 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI001 pin or TI011 pin to enable the operation of 16-bit timer counter 01 (TM01). Care is therefore required when pulling up the TI001 or TI011 pin. However, when re-enabling operation for TI001 pin or TI011 pin are high level after the operation has been stopped, the rising edge is not detected.

4. When TI011 pin is used valid edge, it cannot be used as the timer output (TO01) to P06, and when TO01 is used, it cannot be used to the TI011 pin valid edge.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 7-23. Format of Prescaler Mode Register 02 (PRM02)

Address: FF59H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM02	ES121	ES120	ES021	ES020	0	0	PRM021	PRM020

ES121	ES120	TI012 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES021	ES020	TI002 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM021	PRM020	Count clock selection				
			f_{PRS} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}	4 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^2$	1 MHz	1.25 MHz	2.5 MHz	5 MHz
1	0	$f_{PRS}/2^8$	15.62 kHz	19.53 kHz	39.06 kHz	78.12 kHz
1	1	TI002 valid edge[Note]				

Note The external clock requires a pulse two cycles longer than internal clock (f_{PRS}).

Cautions 1. Always set data to PRM02 after stopping the timer operation.

2. If the valid edge of the TI002 pin is to be set for the count clock, do not set the clear & start mode using the valid edge of the TI002 pin and the capture trigger.

3. If the TI002 or TI012 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI002 pin or TI012 pin to enable the operation of 16-bit timer counter 02 (TM02). Care is therefore required when pulling up the TI002 or TI012 pin. However, when re-enabling operation for TI002 pin or TI012 pin are high level after the operation has been stopped, the rising edge is not detected.

4. When TI012 pin is used valid edge, it cannot be used as the timer output (TO02) to P32, and when TO02 is used, it cannot be used to the TI012 pin valid edge.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 7-24. Format of Prescaler Mode Register 03 (PRM03)

Address: FF51H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM03	ES131	ES130	ES031	ES030	0	0	PRM031	PRM030

ES131	ES130	TI013 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES031	ES030	TI003 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM031	PRM030	Count clock selection				
			f_{PRS} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	f_{PRS}	4 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^4$	250 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	$f_{PRS}/2^6$	62.5 kHz	78.125 kHz	156.25 kHz	312.5 kHz
1	1	TI003 valid edge[Note]				

Note The external clock requires a pulse two cycles longer than internal clock (f_{PRS}).

Cautions 1. Always set data to PRM03 after stopping the timer operation.

2. If the valid edge of the TI003 pin is to be set for the count clock, do not set the clear & start mode using the valid edge of the TI003 pin and the capture trigger.

3. If the TI003 or TI013 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI003 pin or TI013 pin to enable the operation of 16-bit timer counter 03 (TM03). Care is therefore required when pulling up the TI003 or TI013 pin. However, when re-enabling operation for TI003 pin or TI013 pin are high level after the operation has been stopped, the rising edge is not detected.

4. When TI013 pin is used valid edge, it cannot be used as the timer output (TO03) to P132, and when TO03 is used, it cannot be used to the TI013 pin valid edge.

Remark f_{PRS}: Peripheral hardware clock frequency

(5) Port mode register 0 (PM0)

This register sets port 0 input/output in 1-bit units.

When using the P01/TO00/TI010 and P06/TO01/TI011 pins for timer output, set PM01 and PM06 and the output latch of P01 and P06 to 0.

When using the P01/TO00/TI010 and P06/TO01/TI011 pins for timer input, set PM01 and PM06 to 1. At this time, the output latch of P01 and P06 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.

Figure 7-25. 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	PM06	PM05	1	1	1	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0, 1, 5, 6)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

(6) Port mode register 3 (PM3)

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

When using the P32/TO02/TI012/INTP3 pin for timer output, set PM32 and the output latch of P32 to 0.

When using the P31/TI002/INTP2 and P32/TI012/TO02/INTP3 pins for timer input, set PM31 and PM32 to 1. At this time, the output latch of P31 and P32 may be 0 or 1.

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

Reset signal generation sets PM3 to FFH.

Figure 7-26. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

(7) Port mode register 13 (PM13)

This register sets port 13 input/output in 1-bit units.

When using the P132/TO03/TI013 pin for timer output, set PM132 and the output latch of P132 to 0.

When using the P131/TI003 and P132/TI013/TO03 pins for timer input, set PM131 and PM132 to 1. At this time, the output latch of P131 and P132 may be 0 or 1.

PM13 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM13 to FFH.

Figure 7-27. Format of Port Mode Register 13 (PM13)

Address: FF2DH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM13	1	1	1	1	1	PM132	PM131	0

PM13n	P13n pin I/O mode selection (n = 1, 2)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

7.4 Operation of 16-Bit Timer/Event Counters 00 to 03

7.4.1 Interval timer operation

Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 7-28 allows operation as an interval timer.

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC0n register (see **Figure 7-28** for the set value).
<2> Set any value to the CR00n register.
<3> Set the count clock by using the PRM0n register.
<4> Set the TMC0n register to start the operation (see **Figure 7-28** for the set value).

Caution CR00n cannot be rewritten during TM0n operation.

Remark For how to enable the INTTM00n interrupt, see **CHAPTER 17 INTERRUPT FUNCTIONS**.

Interrupt requests are generated repeatedly using the count value preset in 16-bit timer capture/compare register 00n (CR00n) as the interval.

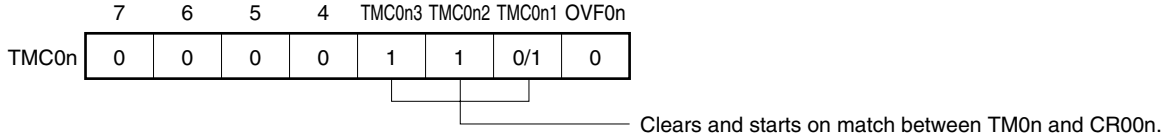
When the count value of 16-bit timer counter 0n (TM0n) matches the value set in CR00n, counting continues with the TM0n value cleared to 0 and the interrupt request signal (INTTM00n) is generated.

The count clock of 16-bit timer/event counter 0n can be selected with bits 0 and 1 (PRM0n0, PRM0n1) of prescaler mode register 0n (PRM0n).

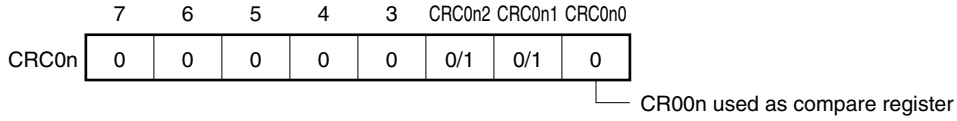
Remark n = 0 to 3

Figure 7-28. Control Register Settings for Interval Timer Operation

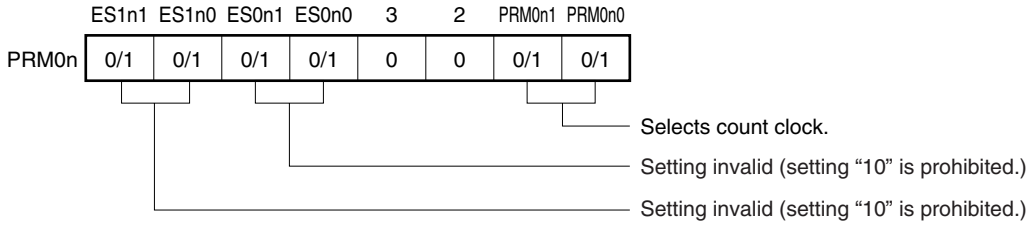
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



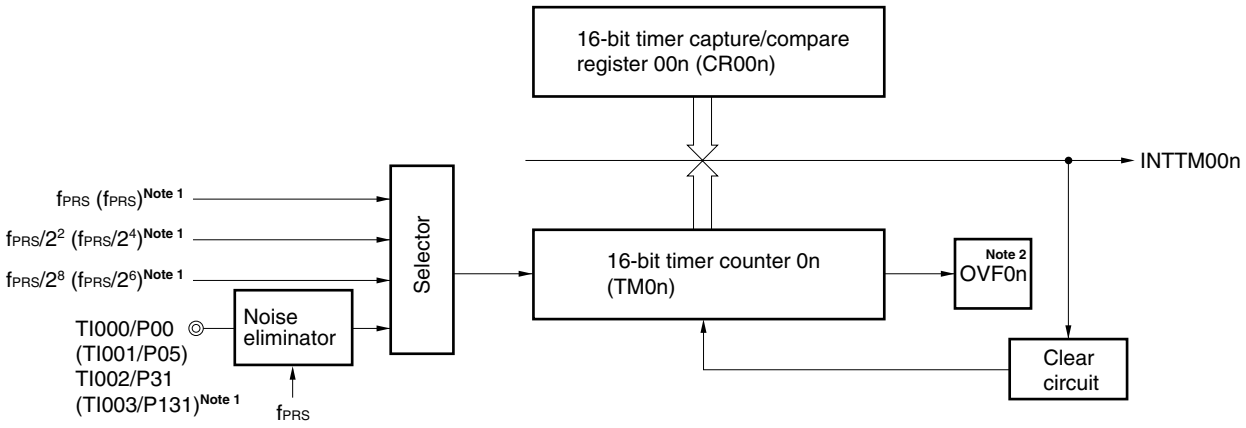
(c) Prescaler mode register 0n (PRM0n)



Remarks 1. 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See the description of the respective control registers for details.

2. n = 0 to 3

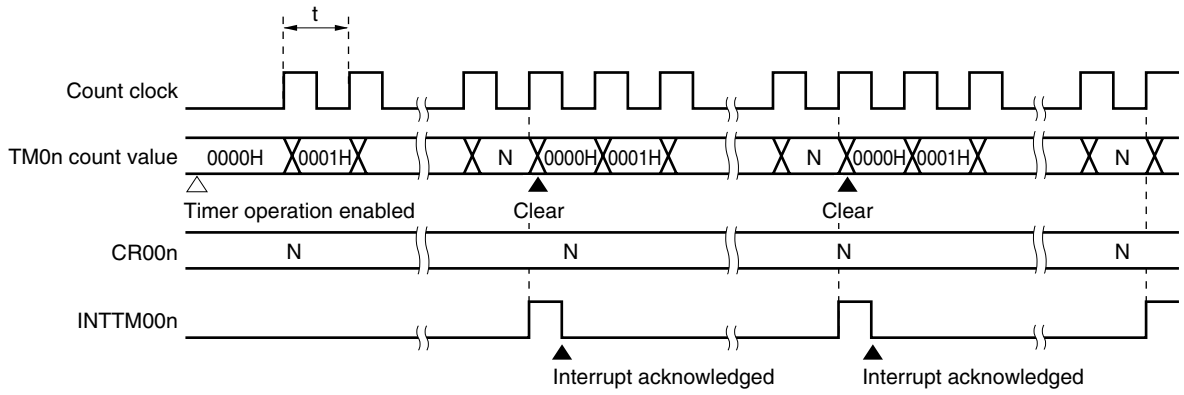
Figure 7-29. Interval Timer Configuration Diagram



Notes 1. Frequencies and pin names without parentheses are for 16-bit timer/event counter 00 and 02, and those in parentheses are for 16-bit timer/event counter 01 and 03.

2. OVF0n is set to 1 only when 16-bit timer capture/compare register 00n is set to FFFFH.

Figure 7-30. Timing of Interval Timer Operation



Remark Interval time = (N + 1) × t

N = 0001H to FFFFH

n = 0 to 3

7.4.2 PPG output operations

Setting 16-bit timer mode control register 0n (TMC0n) and capture/compare control register 0n (CRC0n) as shown in Figure 7-31 allows operation as PPG (Programmable Pulse Generator) output.

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC0n register (see **Figure 7-31** for the set value).
<2> Set any value to the CR00n register as the cycle.
<3> Set any value to the CR01n register as the duty factor.
<4> Set the TOC0n register (see **Figure 7-31** for the set value).
<5> Set the count clock by using the PRM0n register.
<6> Set the TMC0n register to start the operation (see **Figure 7-31** for the set value).

Caution To change the value of the duty factor (the value of the CR01n register) during operation, see Caution 2 in Figure 7-33 PPG Output Operation Timing.

Remarks 1. For the setting of the TO0n pin, see **7.3 (5) Port mode register 0 (PM0) to (7) Port mode register 13 (PM13).**

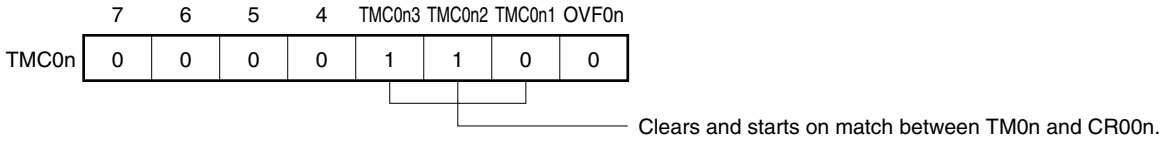
2. For how to enable the INTTM00n interrupt, see **CHAPTER 17 INTERRUPT FUNCTIONS**.

In the PPG output operation, rectangular waves are output from the TO0n pin with the pulse width and the cycle that correspond to the count values preset in 16-bit timer capture/compare register 01n (CR01n) and in 16-bit timer capture/compare register 00n (CR00n), respectively.

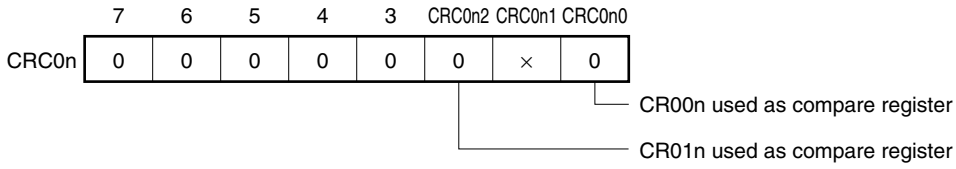
Remark n = 0 to 3

Figure 7-31. Control Register Settings for PPG Output Operation

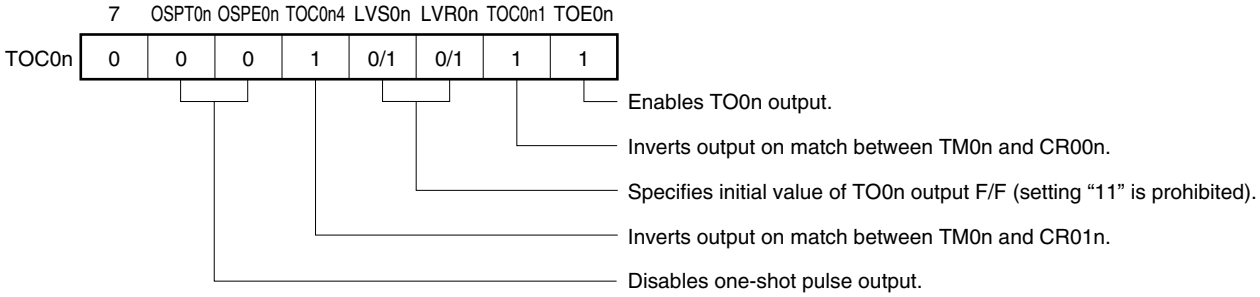
(a) 16-bit timer mode control register 0n (TMC0n)



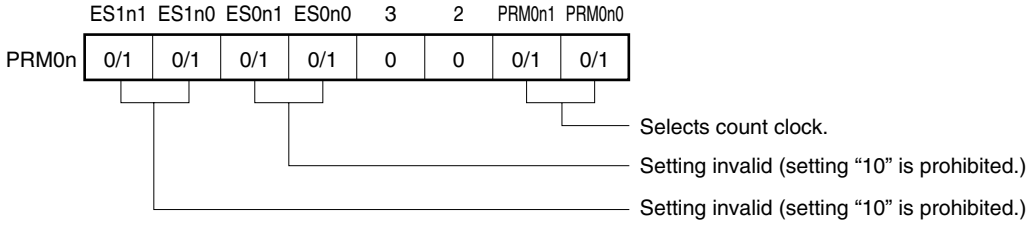
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)



(d) Prescaler mode register 0n (PRM0n)

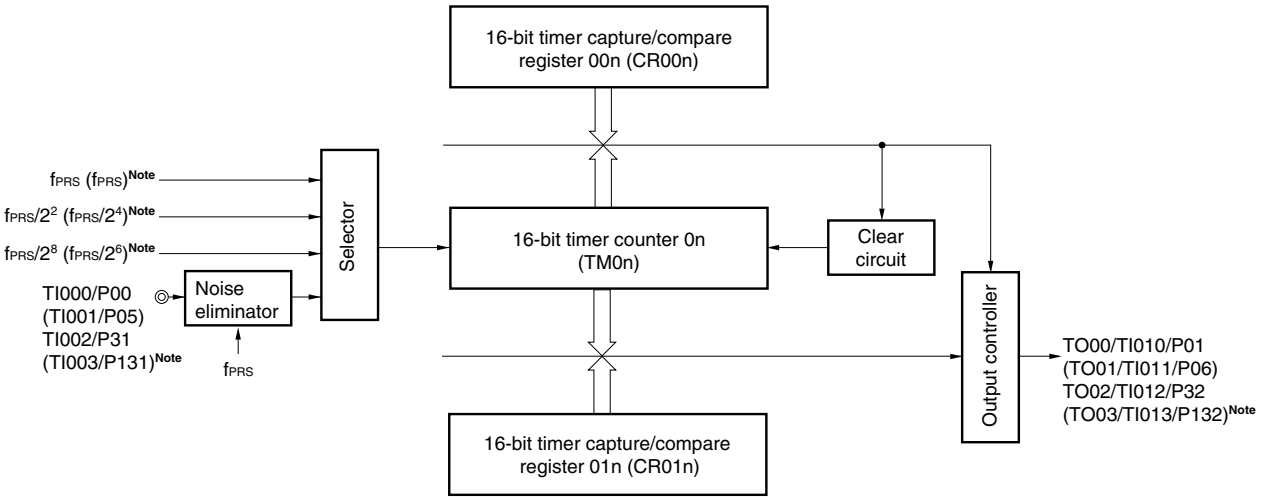


Cautions 1. Values in the following range should be set in CR00n and CR01n:
0000H ≤ CR01n < CR00n ≤ FFFFH

2. The cycle of the pulse generated through PPG output (CR00n setting value + 1) has a duty of (CR01n setting value + 1)/(CR00n setting value + 1).

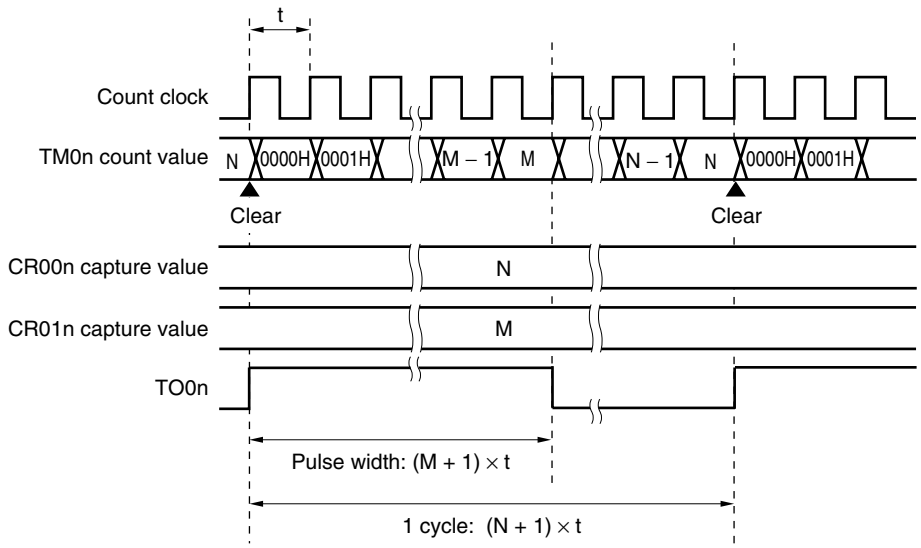
Remark ×: Don’t care
n = 0 to 3

Figure 7-32. Configuration Diagram of PPG Output



Note Frequencies and pin names without parentheses are for 16-bit timer/event counter 00 and 02, and those in parentheses are for 16-bit timer/event counter 01 and 03.

Figure 7-33. PPG Output Operation Timing



Cautions 1. CR00n cannot be rewritten during TM0n operation.

2. In the PPG output operation, change the pulse width (rewrite CR01n) during TM0n operation using the following procedure.

<1> Disable the timer output inversion operation by match of TM0n and CR01n (TOC0n4 = 0)

<2> Disable the INTTM01n interrupt (TMMK01n = 1)

<3> Rewrite CR01n

<4> Wait for 1 cycle of the TM0n count clock

<5> Enable the timer output inversion operation by match of TM0n and CR01n (TOC0n4 = 1)

<6> Clear the interrupt request flag of INTTM01n (TMIF01n = 0)

<7> Enable the INTTM01n interrupt (TMMK01n = 0)

Remarks 1. 0000H ≤ M < N ≤ FFFFH

2. n = 0 to 3

7.4.3 Pulse width measurement operations

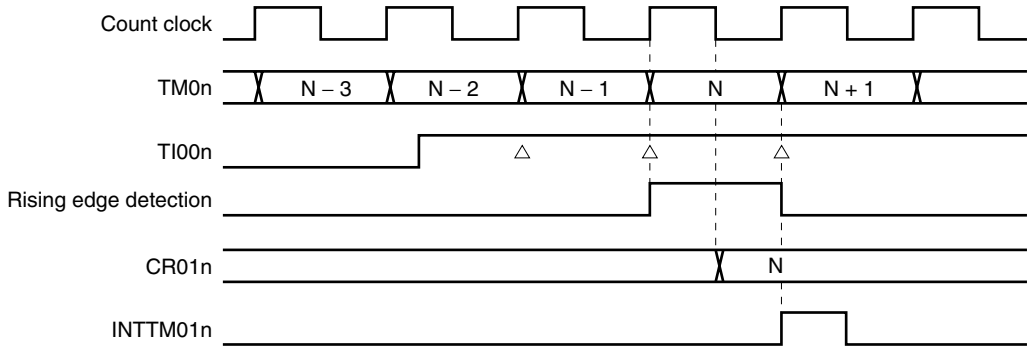
It is possible to measure the pulse width of the signals input to the TI00n pin and TI01n pin using 16-bit timer counter 0n (TM0n).

There are two measurement methods: measuring with TM0n used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI00n pin.

When an interrupt occurs, read the valid value of the capture register, check the overflow flag, and then calculate the necessary pulse width. Clear the overflow flag after checking it.

The capture operation is not performed until the signal pulse width is sampled in the count clock cycle selected by prescaler mode register 0n (PRM0n) and the valid level of the TI00n or TI01n pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-34. CR01n Capture Operation with Rising Edge Specified



Setting

The basic operation setting procedure is as follows.

<1> Set the CRC0n register (see **Figures 7-35**, **7-38**, **7-40**, and **7-42** for the set value).
<2> Set the count clock by using the PRM0n register.
<3> Set the TMC0n register to start the operation (see **Figures 7-35**, **7-38**, **7-40**, and **7-42** for the set value).

Caution To use two capture registers, set the TI00n and TI01n pins.

Remarks 1. For the setting of the TI00n (or TI01n) pin**,** see **7.3 (5) Port mode register 0 (PM0) to (7) Port mode register 13 (PM13).**

2. For how to enable the INTTM00n (or INTTM01n) interrupt, see **CHAPTER 17 INTERRUPT FUNCTIONS.**

3. n = 0 to 3

(1) Pulse width measurement with free-running counter and one capture register

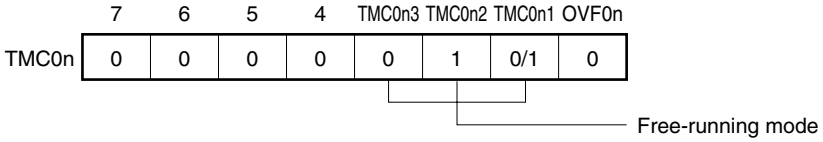
When 16-bit timer counter 0n (TM0n) is operated in free-running mode, and the edge specified by prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an external interrupt request signal (INTTM01n) is set.

Specify both the rising and falling edges of the TI00n pin by using bits 4 and 5 (ES0n0 and ES0n1) of PRM0n.

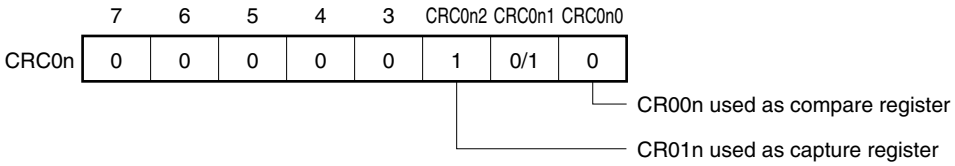
Sampling is performed using the count clock selected by PRM0n, and a capture operation is only performed when a valid level of the TI00n pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-35. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register (When TI00n and CR01n Are Used)

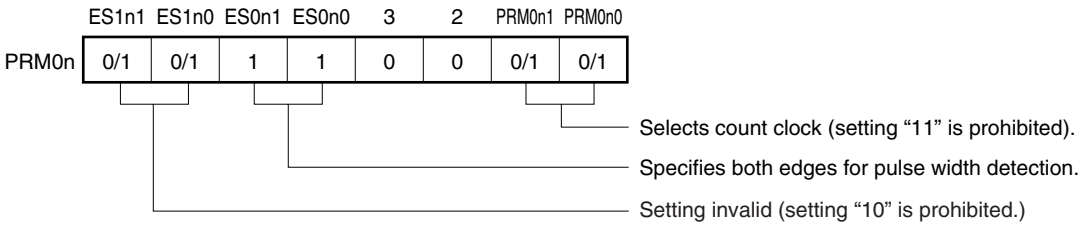
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



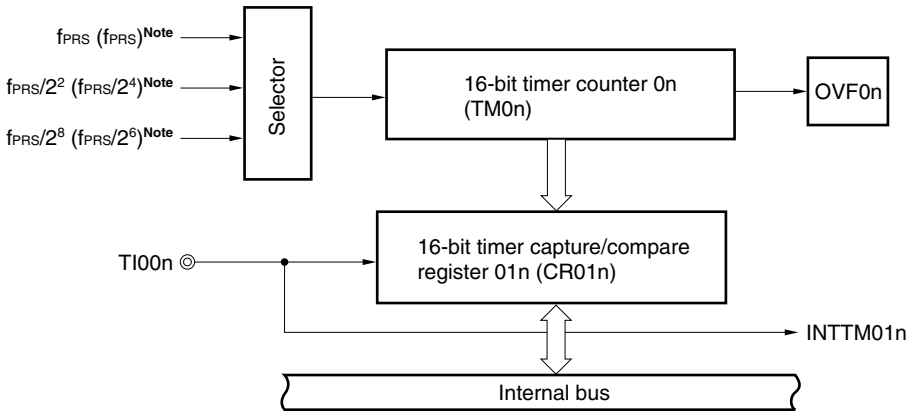
(c) Prescaler mode register 0n (PRM0n)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

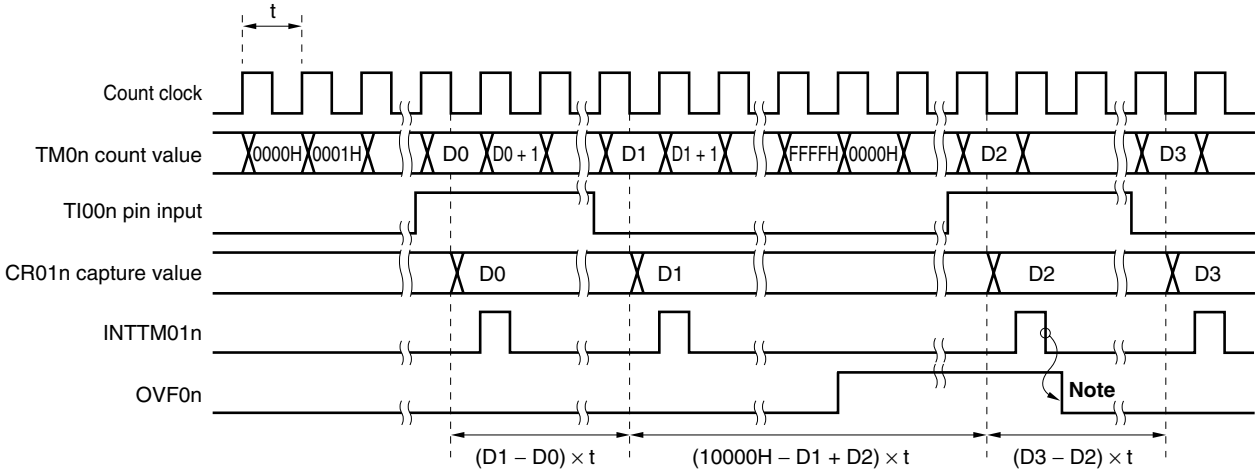
n = 0 to 3

Figure 7-36. Configuration Diagram for Pulse Width Measurement with Free-Running Counter



Note Frequencies without parentheses are for 16-bit timer/event counter 00 and 02, and those in parentheses are for 16-bit timer/event counter 01 and 03.

Figure 7-37. Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)



Note Clear OVF0n by software.

Remark n = 0 to 3

(2) Measurement of two pulse widths with free-running counter

When 16-bit timer counter 0n (TM0n) is operated in free-running mode, it is possible to simultaneously measure the pulse widths of the two signals input to the TI00n pin and the TI01n pin.

When the edge specified by bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an interrupt request signal (INTTM01n) is set.

Also, when the edge specified by bits 6 and 7 (ES1n0 and ES1n1) of PRM0n is input to the TI01n pin, the value of TM0n is taken into 16-bit timer capture/compare register 00n (CR00n) and an interrupt request signal (INTTM00n) is set.

Specify both the rising and falling edges as the edges of the TI00n and TI01n pins, by using bits 4 and 5 (ES0n0 and ES0n1) and bits 6 and 7 (ES1n0 and ES1n1) of PRM0n.

Sampling is performed using the count clock cycle selected by prescaler mode register 0n (PRM0n), and a capture operation is only performed when a valid level of the TI00n or TI01n pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-38. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

(a) 16-bit timer mode control register 0n (TMC0n)

	7	6	5	4	TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 0n (CRC0n)

	7	6	5	4	3	CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	1	0	1

CR00n used as capture register

Captures valid edge of TI01n pin to CR00n.

CR01n used as capture register

(c) Prescaler mode register 0n (PRM0n)

	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	1	1	1	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).

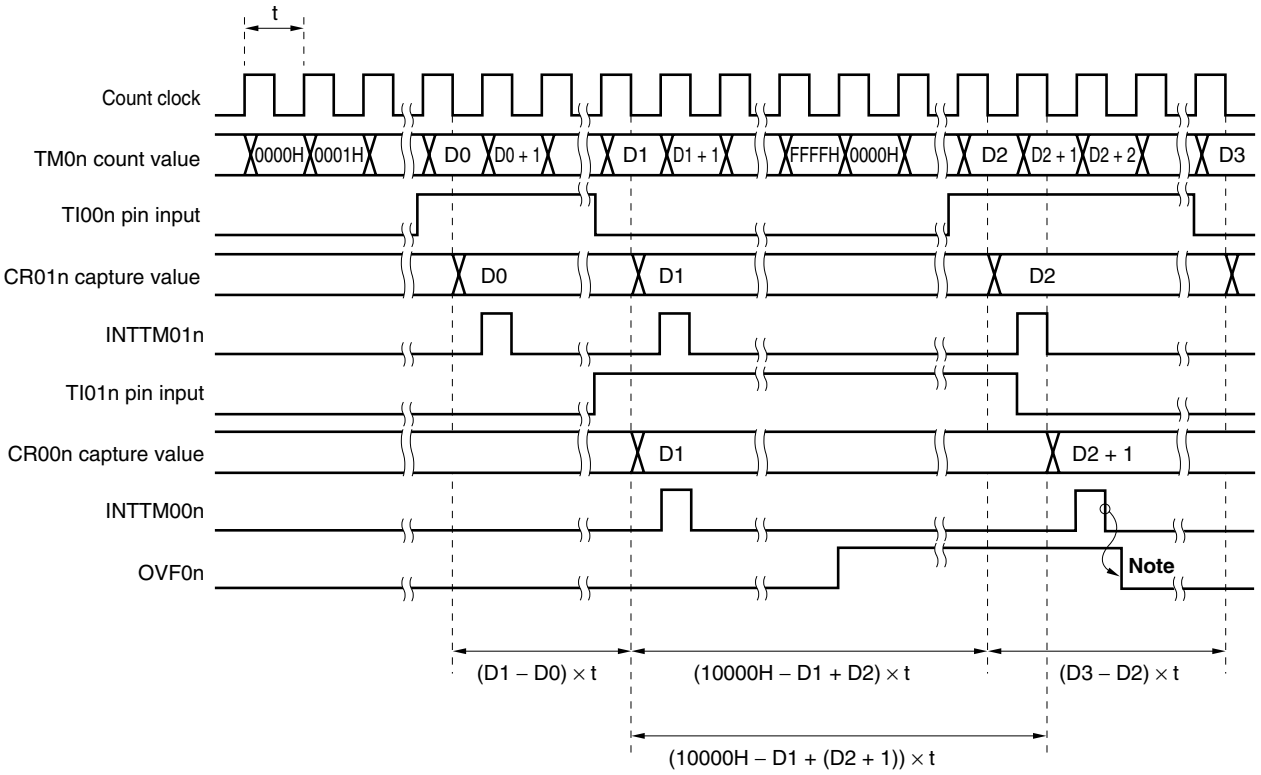
Specifies both edges for pulse width detection.

Specifies both edges for pulse width detection.

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

n = 0 to 3

Figure 7-39. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



Note Clear OVF0n by software.

Remark n = 0 to 3

(3) Pulse width measurement with free-running counter and two capture registers

When 16-bit timer counter 0n (TM0n) is operated in free-running mode, it is possible to measure the pulse width of the signal input to the TI00n pin.

When the rising or falling edge specified by bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n) is input to the TI00n pin, the value of TM0n is taken into 16-bit timer capture/compare register 01n (CR01n) and an interrupt request signal (INTTM01n) is set.

Also, when the inverse edge to that of the capture operation is input into CR01n, the value of TM0n is taken into 16-bit timer capture/compare register 00n (CR00n).

Sampling is performed using the count clock cycle selected by prescaler mode register 0n (PRM0n), and a capture operation is only performed when a valid level of the TI00n pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-40. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)

(a) 16-bit timer mode control register 0n (TMC0n)

	7	6	5	4	TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 0n (CRC0n)

	7	6	5	4	3	CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	1	1	1

CR00n used as capture register

Captures to CR00n at inverse edge to valid edge of TI00n.

CR01n used as capture register

(c) Prescaler mode register 0n (PRM0n)

	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	0	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).

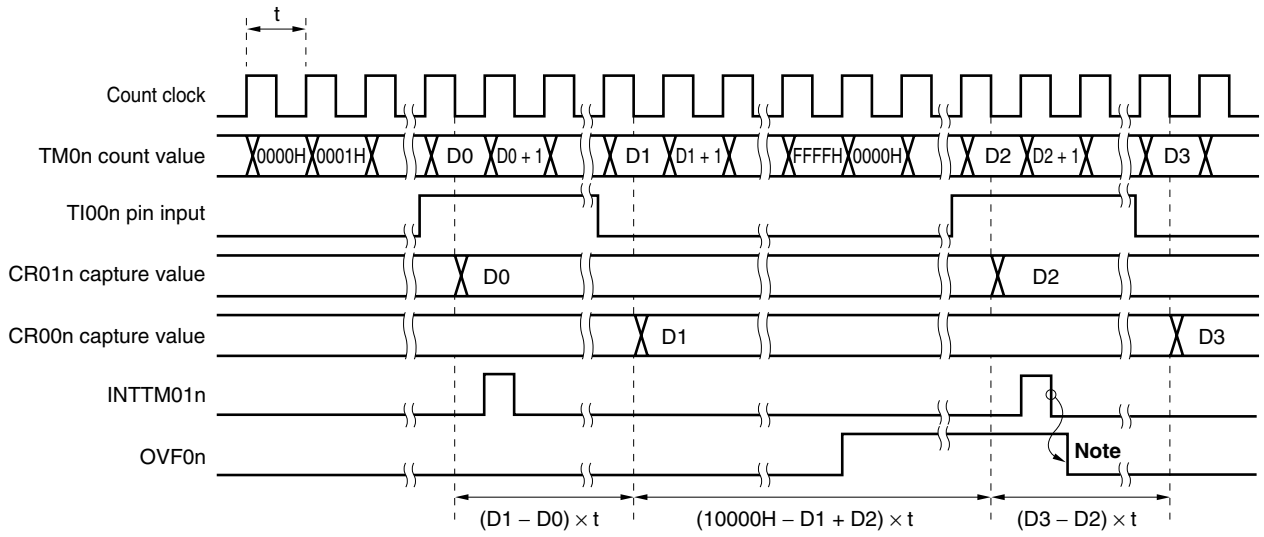
Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

n = 0 to 3

Figure 7-41. Timing of Pulse Width Measurement Operation with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Note Clear OVF0n by software.

(4) Pulse width measurement by means of restart

When input of a valid edge to the TI00n pin is detected, the count value of 16-bit timer counter 0n (TM0n) is taken into 16-bit timer capture/compare register 01n (CR01n), and then the pulse width of the signal input to the TI00n pin is measured by clearing TM0n and restarting the count operation.

Either of two edges—rising or falling—can be selected using bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

Sampling is performed using the count clock cycle selected by prescaler mode register 0n (PRM0n) and a capture operation is only performed when a valid level of the TI00n pin is detected twice, thus eliminating noise with a short pulse width.

Remark n = 0 to 3

Figure 7-42. Control Register Settings for Pulse Width Measurement by Means of Restart (with Rising Edge Specified)

(a) 16-bit timer mode control register 0n (TMC0n)

	7	6	5	4	TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	0	0/1	0

Clears and starts at valid edge of TI00n pin.

(b) Capture/compare control register 0n (CRC0n)

	7	6	5	4	3	CRC0n2	CRC0n1	CRC00n
CRC0n	0	0	0	0	0	1	1	1

CR00n used as capture register

Captures to CR00n at inverse edge to valid edge of TI00n.

CR01n used as capture register

(c) Prescaler mode register 0n (PRM0n)

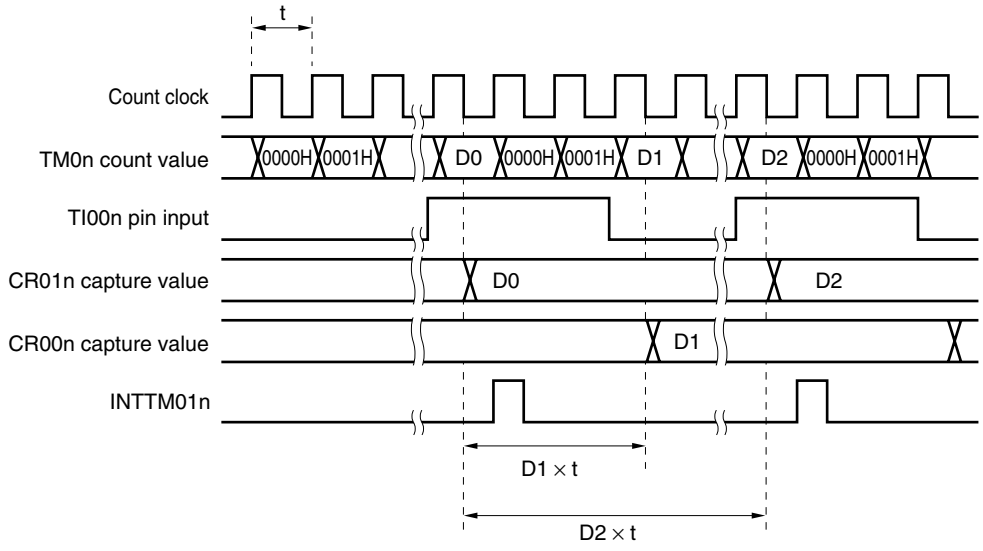
	ES1n1	ES1n0	ES0n1	ES0n0	3	2	PRM0n1	PRM0n0
PRM0n	0/1	0/1	0	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).

Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Figure 7-43. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



Remark n = 0 to 3

7.4.4 External event counter operation

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC0n register (see **Figure 7-44** for the set value).
<2> Set the count clock by using the PRM0n register.
<3> Set any value to the CR00n register (0000H cannot be set).
<4> Set the TMC0n register to start the operation (see **Figure 7-44** for the set value).

Remarks 1. For the setting of the TI00n pin, see **7.3 (5) Port mode register 0 (PM0) to (7) Port mode register 13 (PM13)**.

2. For how to enable the INTTM00n interrupt, see **CHAPTER 17 INTERRUPT FUNCTIONS**.

The external event counter counts the number of external clock pulses input to the TI00n pin using 16-bit timer counter 0n (TM0n).

TM0n is incremented each time the valid edge specified by prescaler mode register 0n (PRM0n) is input.

When the TM0n count value matches the 16-bit timer capture/compare register 00n (CR00n) value, TM0n is cleared to 0 and the interrupt request signal (INTTM00n) is generated.

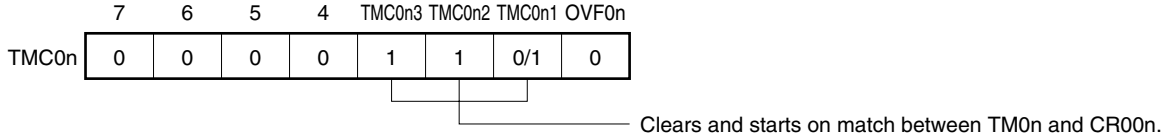
Input a value other than 0000H to CR00n (a count operation with 1-bit pulse cannot be carried out).

Any of three edges—rising, falling, or both edges—can be selected using bits 4 and 5 (ES0n0 and ES0n1) of prescaler mode register 0n (PRM0n).

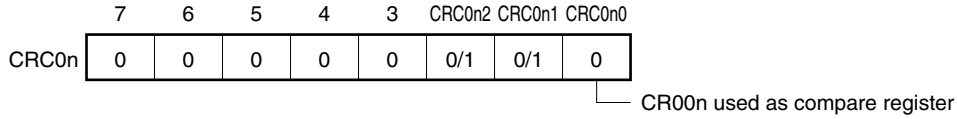
Sampling is performed using the internal clock (f_{PRS}) and an operation is only performed when a valid level of the TI00n pin is detected twice, thus eliminating noise with a short pulse width.

Figure 7-44. Control Register Settings in External Event Counter Mode (with Rising Edge Specified)

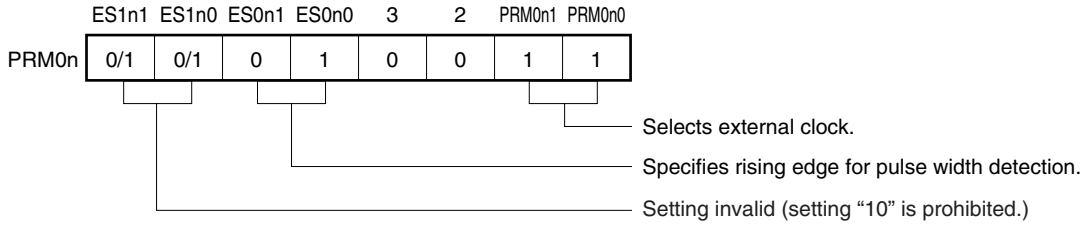
(a) 16-bit timer mode control register 0n (TMC0n)



(b) Capture/compare control register 0n (CRC0n)



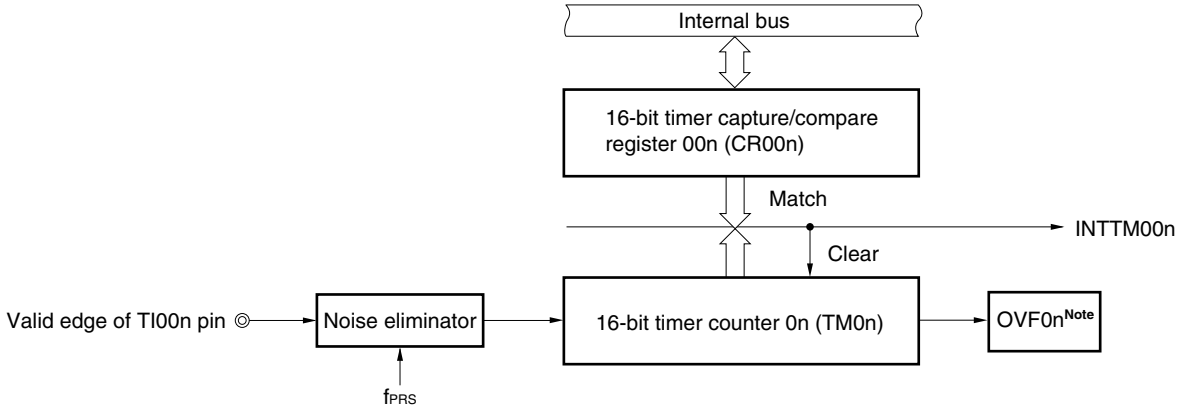
(c) Prescaler mode register 0n (PRM0n)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See the description of the respective control registers for details.

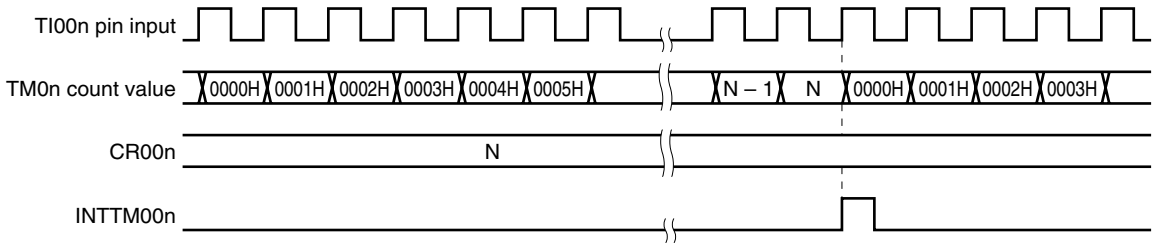
n = 0 to 3

Figure 7-45. Configuration Diagram of External Event Counter



Note OVF0n is set to 1 only when CR00n is set to FFFFH.

Figure 7-46. External Event Counter Operation Timing (with Rising Edge Specified)



Caution When reading the external event counter count value, TM0n should be read.

Remark n = 0 to 3

7.4.5 Square-wave output operation

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM0n register.
<2> Set the CRC0n register (see **Figure 7-47** for the set value).
<3> Set the TOC0n register (see **Figure 7-47** for the set value).
<4> Set any value to the CR00n register (0000H cannot be set).
<5> Set the TMC0n register to start the operation (see **Figure 7-47** for the set value).

Caution CR00n cannot be rewritten during TM0n operation.

Remarks 1. For the setting of the TO0n pin, see **7.3 (5) Port mode register 0 (PM0) to (7) Port mode register 13 (PM13)**.

2. For how to enable the INTTM00n interrupt, see **CHAPTER 17 INTERRUPT FUNCTIONS**.

A square wave with any selected frequency can be output at intervals determined by the count value preset to 16-bit timer capture/compare register 00n (CR00n).

The TO0n pin output status is reversed at intervals determined by the count value preset to CR00n + 1 by setting bit 0 (TOE0n) and bit 1 (TOC0n1) of 16-bit timer output control register 0n (TOC0n) to 1. This enables a square wave with any selected frequency to be output.

Figure 7-47. Control Register Settings in Square-Wave Output Mode (1/2)

(a) 16-bit timer mode control register 0n (TMC0n)

	7	6	5	4	TMC0n3	TMC0n2	TMC0n1	OVF0n
TMC0n	0	0	0	0	1	1	0	0

Clears and starts on match between TM0n and CR00n.

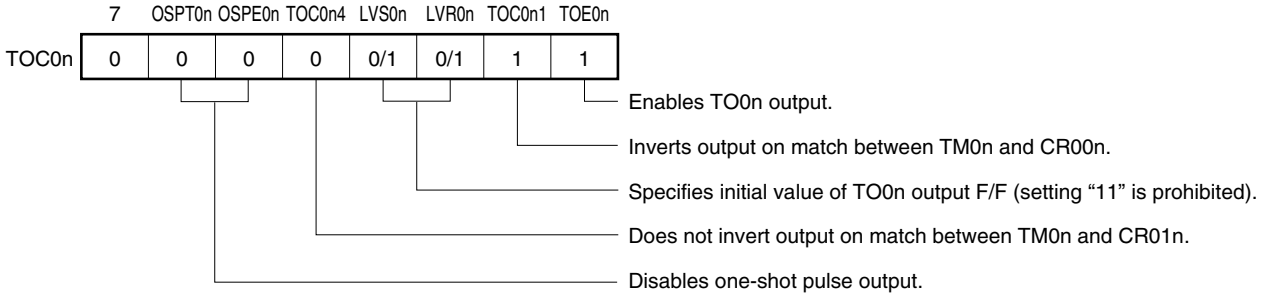
(b) Capture/compare control register 0n (CRC0n)

	7	6	5	4	3	CRC0n2	CRC0n1	CRC0n0
CRC0n	0	0	0	0	0	0/1	0/1	0

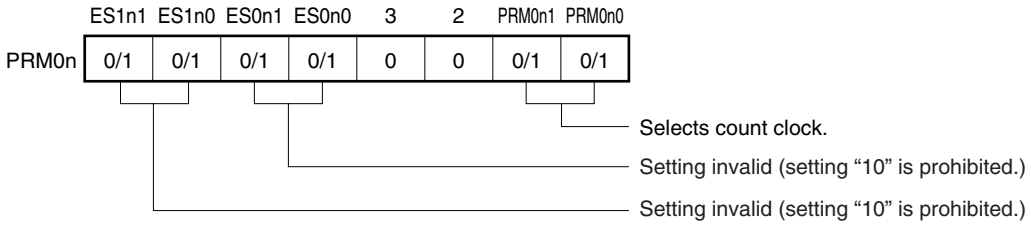
CR00n used as compare register

Figure 7-47. Control Register Settings in Square-Wave Output Mode (2/2)

(c) 16-bit timer output control register 0n (TOC0n)



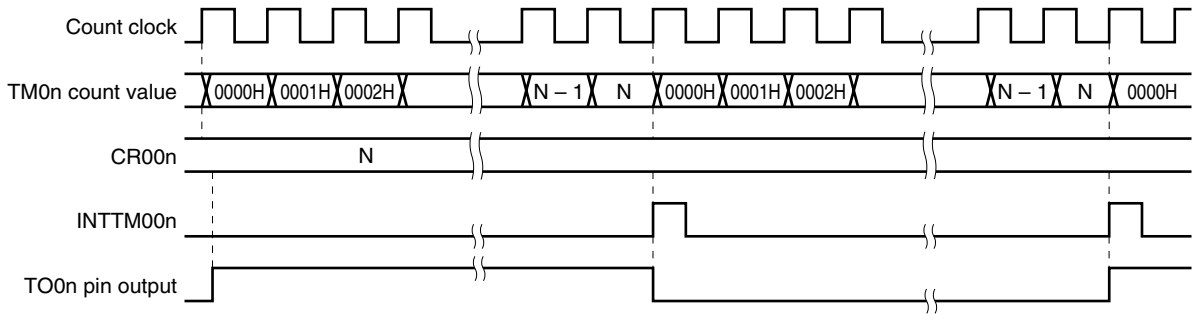
(d) Prescaler mode register 0n (PRM0n)



Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See the description of the respective control registers for details.

n = 0 to 3

Figure 7-48. Square-Wave Output Operation Timing



Remark n = 0 to 3

7.4.6 One-shot pulse output operation

16-bit timer/event counter 0n can output a one-shot pulse in synchronization with a software trigger or an external trigger (TI00n pin input).

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM0n register.
<2> Set the CRC0n register (see **Figures 7-49** and **7-51** for the set value).
<3> Set the TOC0n register (see **Figures 7-49** and **7-51** for the set value).
<4> Set any value to the CR00n and CR01n registers (0000H cannot be set).
<5> Set the TMC0n register to start the operation (see **Figures 7-49** and **7-51** for the set value).

Remarks 1. For the setting of the TO0n pin, see **7.3 (5) Port mode register 0 (PM0) to (7) Port mode register 13 (PM13)**.

2. For how to enable the INTTM00n (if necessary, INTTM01n) interrupt, see **CHAPTER 17 INTERRUPT FUNCTIONS**.

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TO0n pin by setting 16-bit timer mode control register 0n (TMC0n), capture/compare control register 0n (CRC0n), and 16-bit timer output control register 0n (TOC0n) as shown in Figure 7-49, and by setting bit 6 (OSPT0n) of the TOC0n register to 1 by software.

By setting the OSPT0n bit to 1, 16-bit timer/event counter 0n is cleared and started, and its output becomes active at the count value (N) set in advance to 16-bit timer capture/compare register 01n (CR01n). After that, the output becomes inactive at the count value (M) set in advance to 16-bit timer capture/compare register 00n (CR00n)[Note].

Even after the one-shot pulse has been output, the TM0n register continues its operation. To stop the TM0n register, the TMC0n3 and TMC0n2 bits of the TMC0n register must be set to 00.

Note The case where N < M is described here. When N > M, the output becomes active with the CR00n register and inactive with the CR01n register. Do not set N to M.

Cautions 1. Do not set the OSPT0n bit while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is completed.

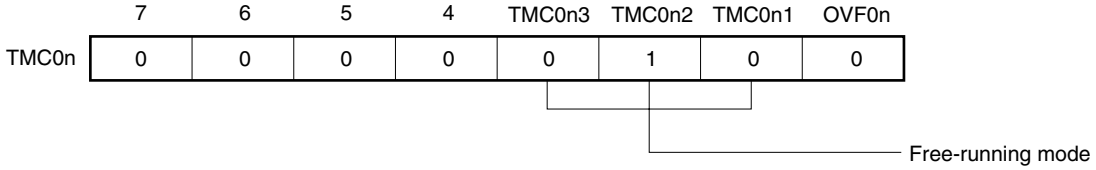
2. When using the one-shot pulse output of 16-bit timer/event counter 0n with a software trigger, do not change the level of the TI00n pin or its alternate-function port pin.

Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI00n pin or its alternate-function port pin, resulting in the output of a pulse at an undesired timing.

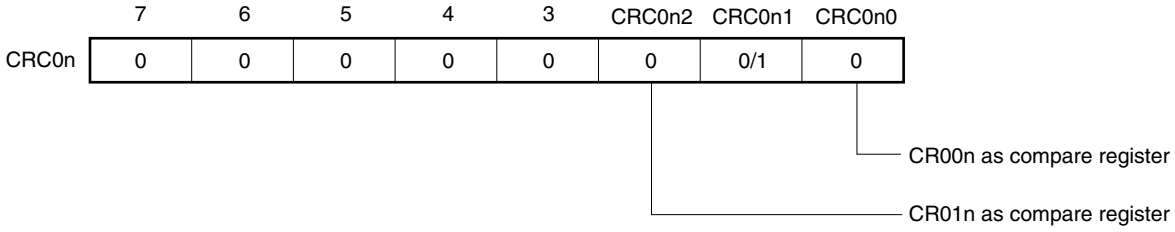
Remark n = 0 to 3

Figure 7-49. Control Register Settings for One-Shot Pulse Output with Software Trigger

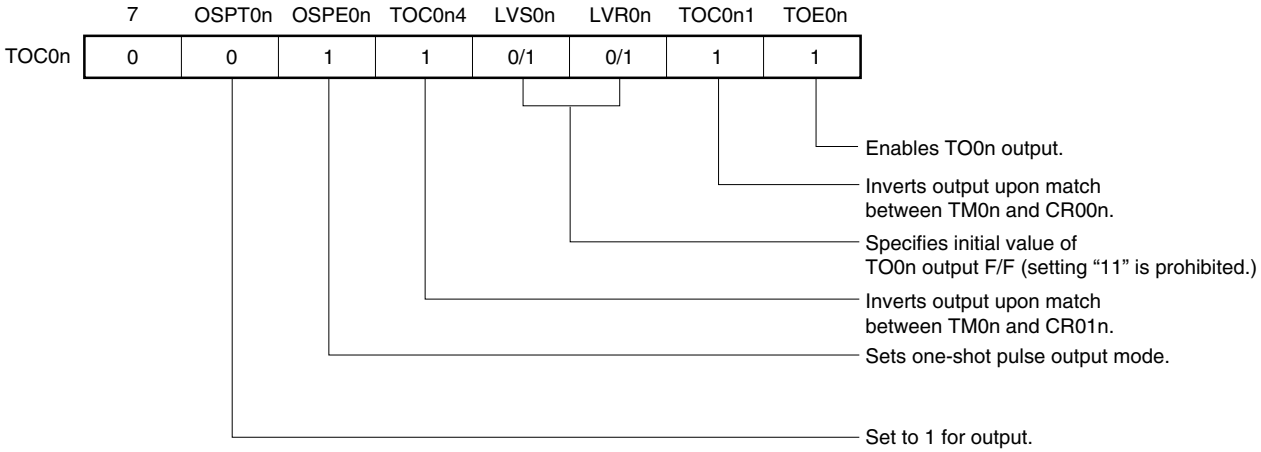
(a) 16-bit timer mode control register 0n (TMC0n)



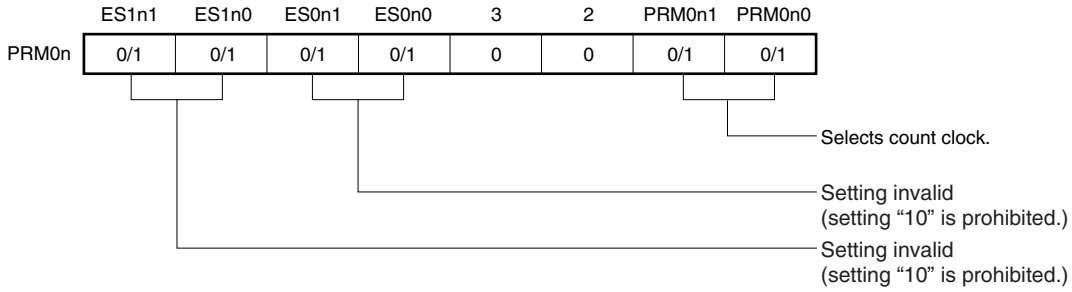
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)



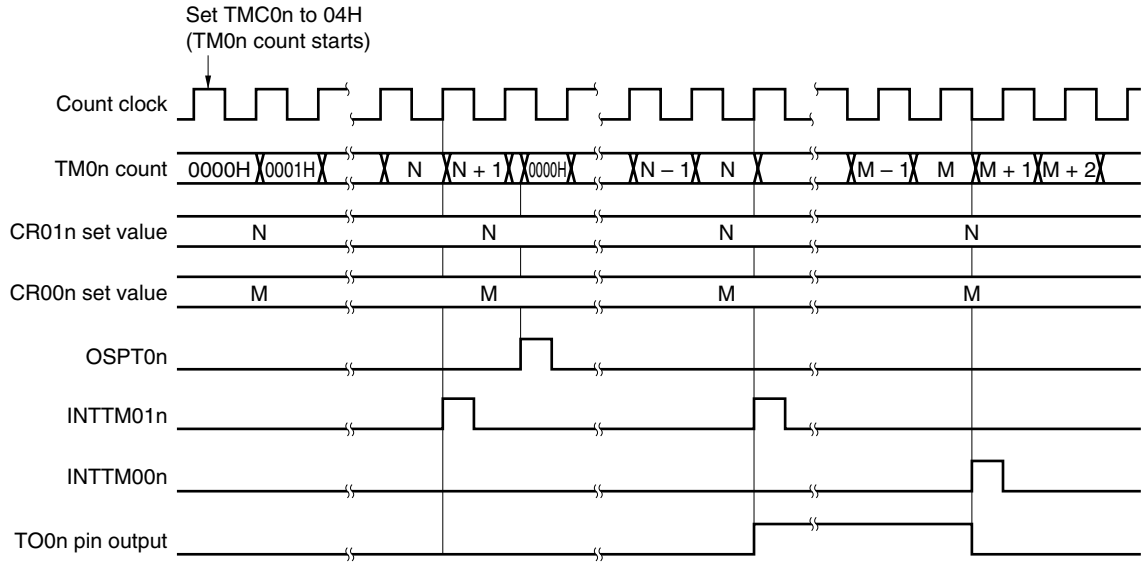
(d) Prescaler mode register 0n (PRM0n)



Caution Do not set 0000H to the CR00n and CR01n registers.

Remark n = 0 to 3

Figure 7-50. Timing of One-Shot Pulse Output Operation with Software Trigger



Caution 16-bit timer counter 0n starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC0n3 and TMC0n2 bits.

Remark N < M

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TO0n pin by setting 16-bit timer mode control register 0n (TMC0n), capture/compare control register 0n (CRC0n), and 16-bit timer output control register 0n (TOC0n) as shown in Figure 7-51, and by using the valid edge of the TI00n pin as an external trigger.

The valid edge of the TI00n pin is specified by bits 4 and 5 (ES0n0, ES0n1) of prescaler mode register 0n (PRM0n). The rising, falling, or both the rising and falling edges can be specified.

When the valid edge of the TI00n pin is detected, the 16-bit timer/event counter is cleared and started, and the output becomes active at the count value set in advance to 16-bit timer capture/compare register 01n (CR01n). After that, the output becomes inactive at the count value set in advance to 16-bit timer capture/compare register 00n (CR00n)[Note].

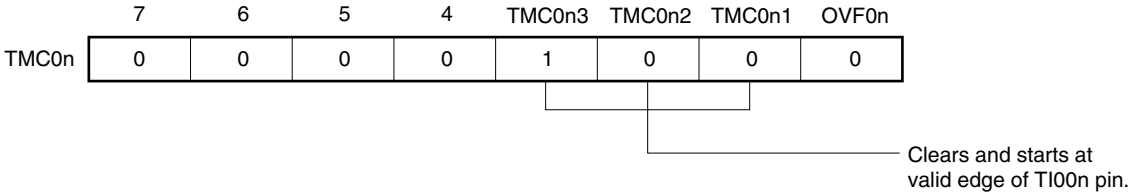
Note The case where N < M is described here. When N > M, the output becomes active with the CR00n register and inactive with the CR01n register. Do not set N to M.

Caution Even if the external trigger is generated again while the one-shot pulse is output, it is ignored.

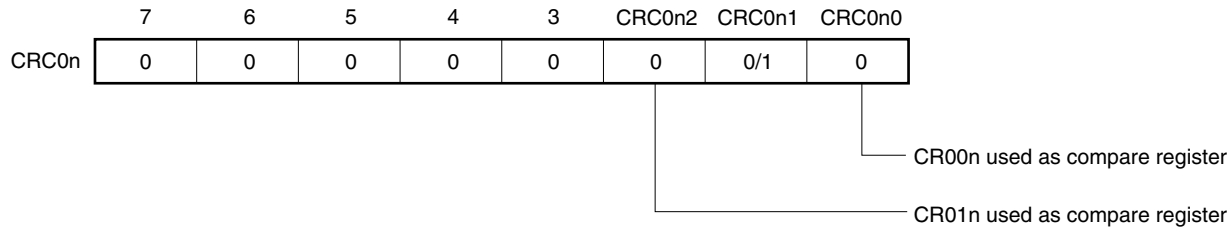
Remark n = 0 to 3

Figure 7-51. Control Register Settings for One-Shot Pulse Output with External Trigger (with Rising Edge Specified)

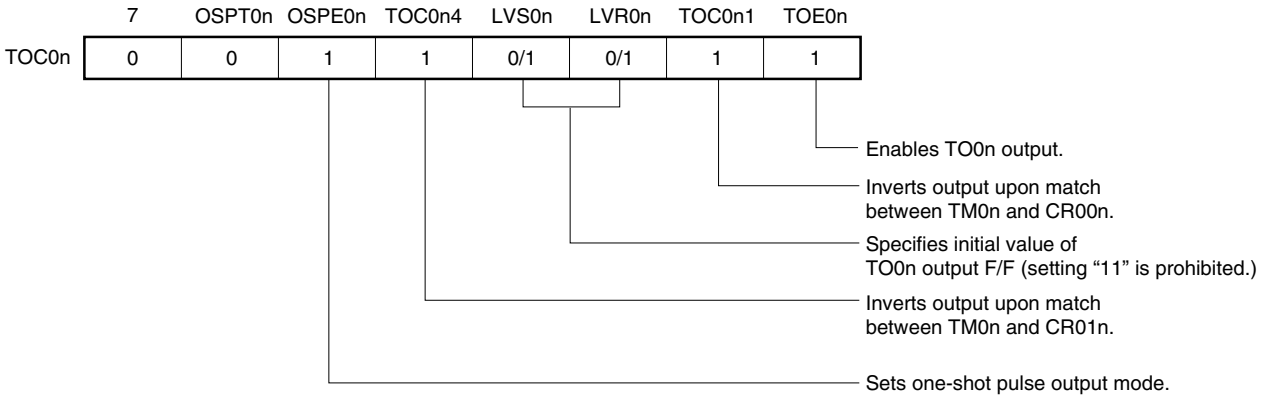
(a) 16-bit timer mode control register 0n (TMC0n)



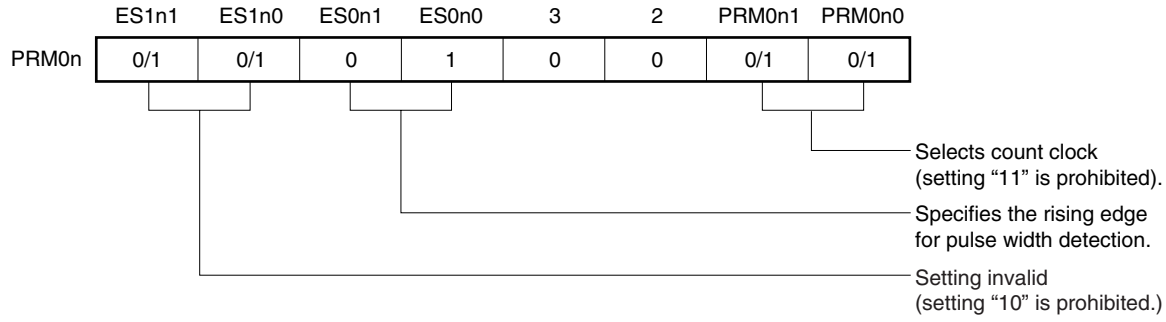
(b) Capture/compare control register 0n (CRC0n)



(c) 16-bit timer output control register 0n (TOC0n)



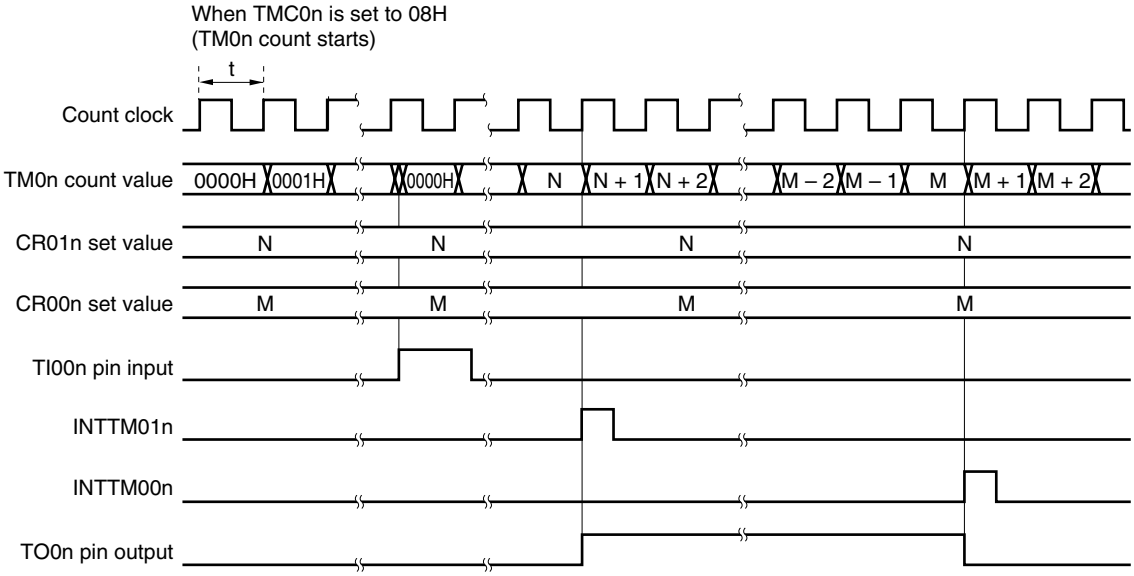
(d) Prescaler mode register 0n (PRM0n)



Caution Do not set the CR00n and CR01n registers to 0000H.

Remark n = 0 to 3

Figure 7-52. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution 16-bit timer counter 0n starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC0n2 and TMC0n3 bits.

Remark N < M
n = 0 to 3

7.5 Special Use of TM0n

7.5.1 Rewriting CR01n during TM0n operation

In principle, rewriting CR00n and CR01n of the 78K0/FE2 when they are used as compare registers is prohibited while TM0n is operating (TMC0n3 and TMC0n2 = other than 00).

However, the value of CR01n can be changed, even while TM0n is operating, using the following procedure if CR01n is used for PPG output and the duty factor is changed (change the value of CR01n immediately after its value matches the value of TM0n. If the value of CR01n is changed immediately before its value matches TM0n, an unexpected operation may be performed).

Procedure for changing value of CR01n

<1> Disable interrupt INTTM01n (TMMK01n = 1).
<2> Disable reversal of the timer output when the value of TM0n matches that of CR01n (TOC0n4 = 0).
<3> Change the value of CR01n.
<4> Wait for one cycle of the count clock of TM0n.
<5> Enable reversal of the timer output when the value of TM0n matches that of CR01n (TOC0n4 = 1).
<6> Clear the interrupt flag of INTTM01n (TMIF01n = 0) to 0.
<7> Enable interrupt INTTM01n (TMMK01n = 0).

Remark For TMIF01n and TMMK01n, see **CHAPTER 17 INTERRUPT FUNCTIONS**.

7.5.2 Setting LVS0n and LVR0n

(1) Usage of LVS0n and LVR0n

LVS0n and LVR0n are used to set the default value of the TO0n pin output and to invert the timer output without enabling the timer operation (TMC0n3 and TMC0n2 = 00). Clear LVS0n and LVR0n to 00 (default value: low-level output) when software control is unnecessary.

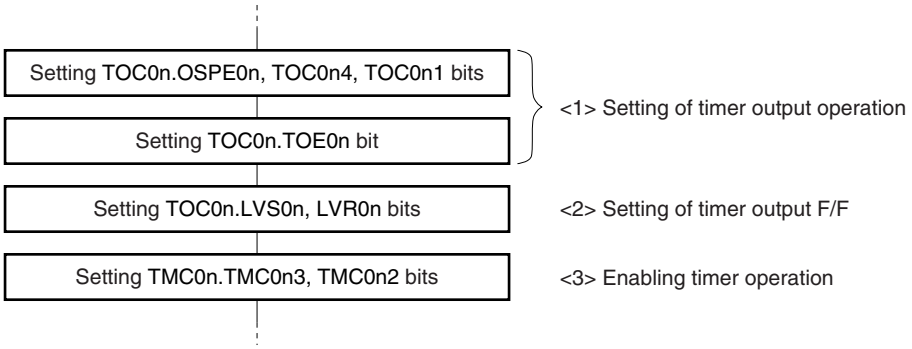
LVS0n	LVR0n	Timer Output Status
0	0	Not changed (low-level output)
0	1	Cleared (low-level output)
1	0	Set (high-level output)
1	1	Setting prohibited

Remark n = 0 to 3

(2) Setting LVS0n and LVR0n

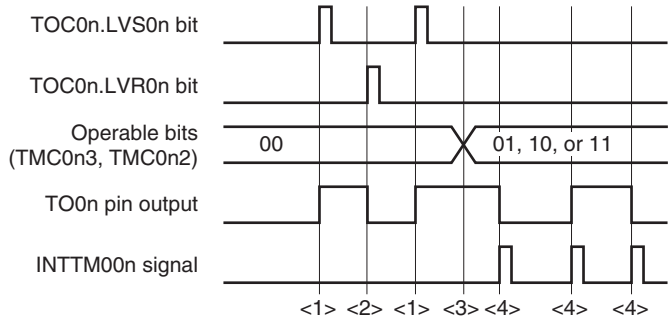
Set LVS0n and LVR0n using the following procedure.

Figure 7-53. Example of Flow for Setting LVS0n and LVR0n Bits



Caution Be sure to set LVS0n and LVR0n following steps <1>, <2>, and <3> above. Step <2> can be performed after <1> and before <3>.

Figure 7-54. Timing Example of LVR0n and LVS0n



<1> The TO0n pin output goes high when LVS0n and LVR0n = 10.

<2> The TO0n pin output goes low when LVS0n and LVR0n = 01 (the pin output remains unchanged from the high level even if LVS0n and LVR0n are cleared to 00).

<3> The timer starts operating when TMC0n3 and TMC0n2 are set to 01, 10, or 11. Because LVS0n and LVR0n were set to 10 before the operation was started, the TO0n pin output starts from the high level. After the timer starts operating, setting LVS0n and LVR0n is prohibited until TMC0n3 and TMC0n2 = 00 (disabling the timer operation).

<4> The output level of the TO0n pin is inverted each time an interrupt signal (INTTM00n) is generated.

Remark n = 0 to 3

7.6 Cautions for 16-Bit Timer/Event Counters 00 to 03

(1) Restrictions for each channel of 16-bit timer/event counter 0n

Table 7-5 shows the restrictions for each channel.

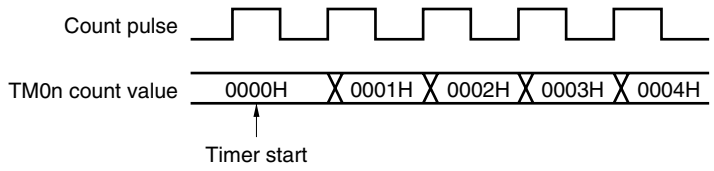
Table 7-5. Restrictions for Each Channel of 16-Bit Timer/Event Counter 0n

Operation	Restriction
As interval timer	–
As square-wave output	
As external event counter	
As clear & start mode entered by TI00n pin valid edge input	Using timer output (TO0n) is prohibited when detection of the valid edge of the TI01n pin is used. (TOC0n = 00H)
As free-running timer	–
As PPG output	0000H ≤ CP01n < CR00n ≤ FFFFH
As one-shot pulse output	Setting the same value to CR00n and CP01n is prohibited.
As pulse width measurement	Using timer output (TO0n) is prohibited (TOC0n = 00H)

(2) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because counting TM0n is started asynchronously to the count pulse.

Figure 7-55. Start Timing of TM0n Count



(3) Setting of CR00n and CR01n (clear & start mode entered upon a match between TM0n and CR00n)

Set a value other than 0000H to CR00n and CR01n (TM0n cannot count one pulse when it is used as an external event counter).

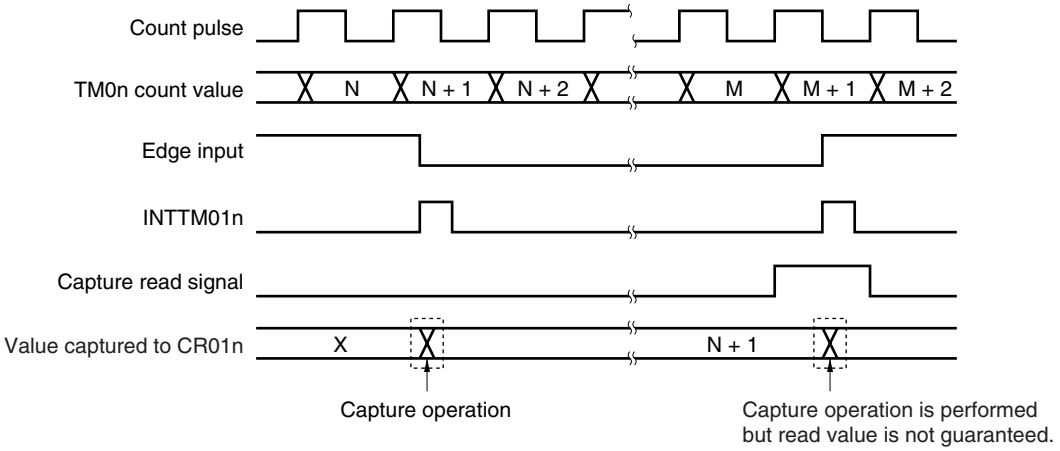
Remark n = 0 to 3

(4) Timing of holding data by capture register

(a) When the valid edge is input to the TI00n/TI01n pin and the reverse phase of the TI00n pin is detected while CR00n/CR01n is read, CR01n performs a capture operation but the read value of CR00n/CR01n is not guaranteed. At this time, an interrupt signal (INTTM00n/INTTM01n) is generated when the valid edge of the TI00n/TI01n pin is detected (the interrupt signal is not generated when the reverse-phase edge of the TI00n pin is detected).

When the count value is captured because the valid edge of the TI00n/TI01n pin was detected, read the value of CR00n/CR01n after INTTM00n/INTTM01n is generated.

Figure 7-56. Timing of Holding Data by Capture Register



(b) The values of CR00n and CR01n are not guaranteed after 16-bit timer/event counter 0n stops.

(5) Setting valid edge

Set the valid edge of the TI00n pin while the timer operation is stopped (TMC0n3 and TMC0n2 = 00). Set the valid edge by using ES0n0 and ES0n1.

(6) Re-triggering one-shot pulse

Make sure that the trigger is not generated while an active level is being output in the one-shot pulse output mode. Be sure to input the next trigger after the current active level is output.

Remark n = 0 to 3

(7) Operation of OVF0n flag

(a) Setting OVF0n flag (1)

The OVF0n flag is set to 1 in the following case, as well as when TM0n overflows.

Select the clear & start mode entered upon a match between TM0n and CR00n.

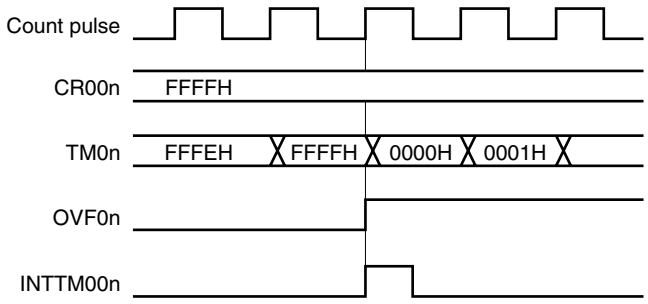
↓

Set CR00n to FFFFH.

↓

When TM0n matches CR00n and TM0n is cleared from FFFFH to 0000H

Figure 7-57. Operation Timing of OVF0n Flag



(b) Clearing OVF0n flag

Even if the OVF0n flag is cleared to 0 after TM0n overflows and before the next count clock is counted (before the value of TM0n becomes 0001H), it is set to 1 again and clearing is invalid.

(8) One-shot pulse output

One-shot pulse output operates correctly in the free-running timer mode or the clear & start mode entered by the TI00n pin valid edge. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM0n and CR00n.

Remark n = 0 to 3

(9) Capture operation

(a) When valid edge of TI00n is specified as count clock

When the valid edge of TI00n is specified as the count clock, the capture register for which TI00n is specified as a trigger does not operate correctly.

(b) Pulse width to accurately capture value by signals input to TI01n and TI00n pins

To accurately capture the count value, the pulse input to the TI00n and TI01n pins as a capture trigger must be wider than two count clocks selected by PRM0n (see **Figure 7-13**).

(c) Generation of interrupt signal

The capture operation is performed at the falling edge of the count clock but the interrupt signals (INTTM00n and INTTM01n) are generated at the rising edge of the next count clock (see **Figure 7-13**).

(d) Note when CRC0n1 (bit 1 of capture/compare control register 0n (CRC0n)) is set to 1

When the count value of the TM0n register is captured to the CR00n register in the phase reverse to the signal input to the TI00n pin, the interrupt signal (INTTM00n) is not generated after the count value is captured. If the valid edge is detected on the TI01n pin during this operation, the capture operation is not performed but the INTTM00n signal is generated as an external interrupt signal. Mask the INTTM00n signal when the external interrupt is not used.

(10) Edge detection

(a) Specifying valid edge after reset

If the operation of the 16-bit timer/event counter 0n is enabled after reset and while the TI00n or TI01n pin is at high level and when the rising edge or both the edges are specified as the valid edge of the TI00n or TI01n pin, then the high level of the TI00n or TI01n pin is detected as the rising edge. Note this when the TI00n or TI01n pin is pulled up. However, the rising edge is not detected when the operation is once stopped and then enabled again.

(b) Sampling clock for eliminating noise

The sampling clock for eliminating noise differs depending on whether the valid edge of TI00n is used as the count clock or capture trigger. In the former case, the sampling clock is fixed to f_{PRS}. In the latter, the count clock selected by PRM0n is used for sampling.

When the signal input to the TI00n pin is sampled and the valid level is detected two times in a row, the valid edge is detected. Therefore, noise having a short pulse width can be eliminated (see **Figure 7-13**).

(11) Timer operation

The signal input to the TI00n/TI01n pin is not acknowledged while the timer is stopped, regardless of the operation mode of the CPU.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. n = 0 to 3

CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 50 AND 51

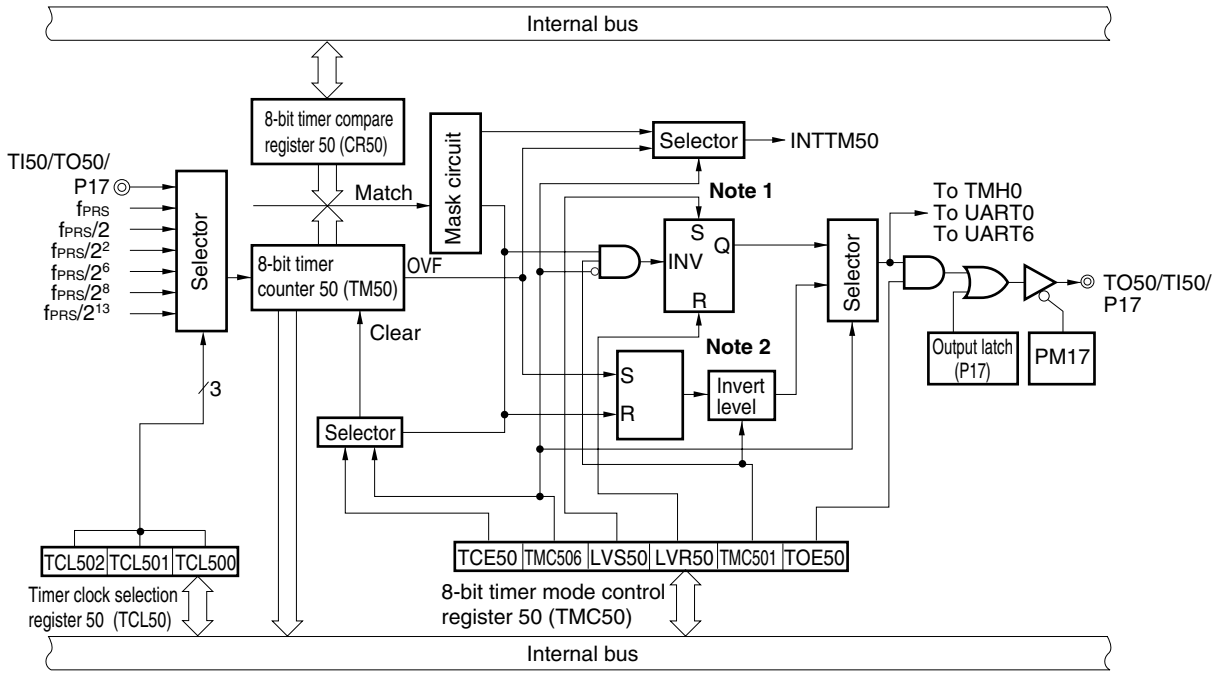
8.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

Figures 8-1 and 8-2 show the block diagrams of 8-bit timer/event counters 50 and 51.

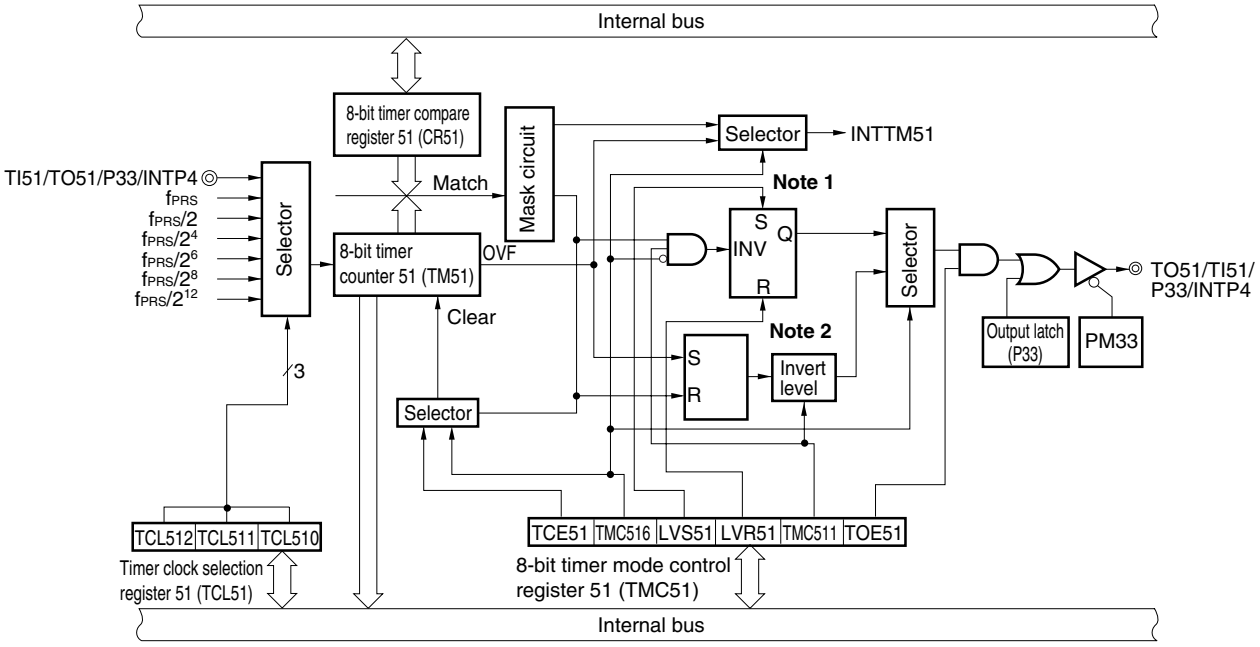
Figure 8-1. Block Diagram of 8-Bit Timer/Event Counter 50



Notes 1. Timer output F/F

2. PWM output F/F

Figure 8-2. Block Diagram of 8-Bit Timer/Event Counter 51



Notes 1. Timer output F/F

2. PWM output F/F

8.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 include the following hardware.

Table 8-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer input	TI5n
Timer output	TO5n
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 1 (PM1) or port mode register 3 (PM3) Port register 1 (P1) or port register 3 (P3)

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit register that counts the count pulses and is read-only.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 8-3. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF16H (TM50), FF1FH (TM51) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0, 1)								

In the following situations, the count value is cleared to 00H.

<1> Reset signal generation

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

Remark n = 0, 1

(2) 8-bit timer compare register 5n (CR5n)

CR5n can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match.

In PWM mode, when the TO5n pin becomes active due to a TM5n overflow and the values of TM5n and CR5n match, the TO5n pin becomes inactive.

The value of CR5n can be set within 00H to FFH.

Reset signal generation clears CR5n to 00H.

Figure 8-4. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF17H (CR50), FF41H (CR51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0, 1)								

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

8.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51

The following four registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 1 (PM1) or port mode register 3 (PM3)
- Port register 1 (P1) or port register 3 (P3)

(1) Timer clock selection register 5n (TCL5n)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of the TI5n pin input.

TCL5n can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears TCL5n to 00H.

Remark n = 0, 1

Figure 8-5. Format of Timer Clock Selection Register 50 (TCL50)

Address: FF6AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection				
				f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	TI50 pin falling edge[Note 1]				
0	0	1	TI50 pin rising edge[Note 2]				
0	1	0	f_{PRS}	4 MHz	8 MHz	10 MHz	20 MHz
0	1	1	$f_{PRS}/2$	2 MHz	4 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^2$	1 MHz	2 MHz	2.5 MHz	5 MHz
1	0	1	$f_{PRS}/2^6$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	15.62 kHz	31.25 kHz	39.06 kHz	78.13 kHz
1	1	1	$f_{PRS}/2^{13}$	0.48 kHz	0.97 kHz	1.22 kHz	2.44 kHz

Notes 1. In the on-board mode, the FLMD0 pin falling edge is selected.

2. In the on-board mode, the FLMD0 pin rising edge is selected.

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

2. Be sure to set bits 3 to 7 to 0.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 8-6. Format of Timer Clock Selection Register 51 (TCL51)

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection				
				f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	TI51 pin falling edge				
0	0	1	TI51 pin rising edge				
0	1	0	f_{PRS}	4 MHz	8 MHz	10 MHz	20 MHz
0	1	1	$f_{PRS}/2$	2 MHz	4 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^4$	500 kHz	1 MHz	625 kHz	1.25 MHz
1	0	1	$f_{PRS}/2^6$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	15.62 kHz	31.25 kHz	39.06 kHz	78.13 kHz
1	1	1	$f_{PRS}/2^{12}$	0.97 kHz	1.95 kHz	2.44 kHz	4.88 kHz

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

2. Be sure to set bits 3 to 7 to 0.

Remark f_{PRS}: Peripheral hardware clock frequency

(2) 8-bit timer mode control register 5n (TMC5n)

TMC5n is a register that performs the following five types of settings.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Timer output F/F (flip flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode.
<5> Timer output control

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 8-7. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF6BH After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Mode in which clear & start occurs on a match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE50	Timer output control
0	Output disabled (TM50 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

(Refer to **Cautions** and **Remarks** on the next page.)

Figure 8-8. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FF43H After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TM51 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

Cautions 1. The settings of LVS5n and LVR5n are valid in other than PWM mode.

2. Perform <1> to <4> below in the following order, not at the same time.

<1> Set TMC5n1, TMC5n6: Operation mode setting

<2> Set TOE5n to enable output: Timer output enable

<3> Set LVS5n, LVR5n (see Caution 1): Timer F/F setting

<4> Set TCE5n

3. Stop operation before rewriting TMC5n6.

Remarks 1. In PWM mode, PWM output is made inactive by clearing TCE5n to 0.

2. If LVS5n and LVR5n are read, the value is 0.

3. The values of the TMC5n6, LVS5n, LVR5n, TMC5n1, and TOE5n bits are reflected at the TO5n pin regardless of the value of TCE5n.

4. n = 0, 1

(3) Port mode registers 1 and 3 (PM1, PM3)

These registers set port 1 and 3 input/output in 1-bit units.

When using the P17/TO50/TI50 and P33/TO51/TI51/INTP4 pins for timer output, clear PM17 and PM33 and the output latches of P17 and P33 to 0.

When using the P17/TO50/TI50 and P33/TO51/TI51/INTP4 pins for timer input, set PM17 and PM33 to 1. The output latches of P17 and P33 at this time may be 0 or 1.

PM1 and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 8-9. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 8-10. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operations of 8-Bit Timer/Event Counters 50 and 51

8.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

Setting

<1> Set the registers.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don't care)

<2> After TCE5n = 1 is set, the count operation starts.

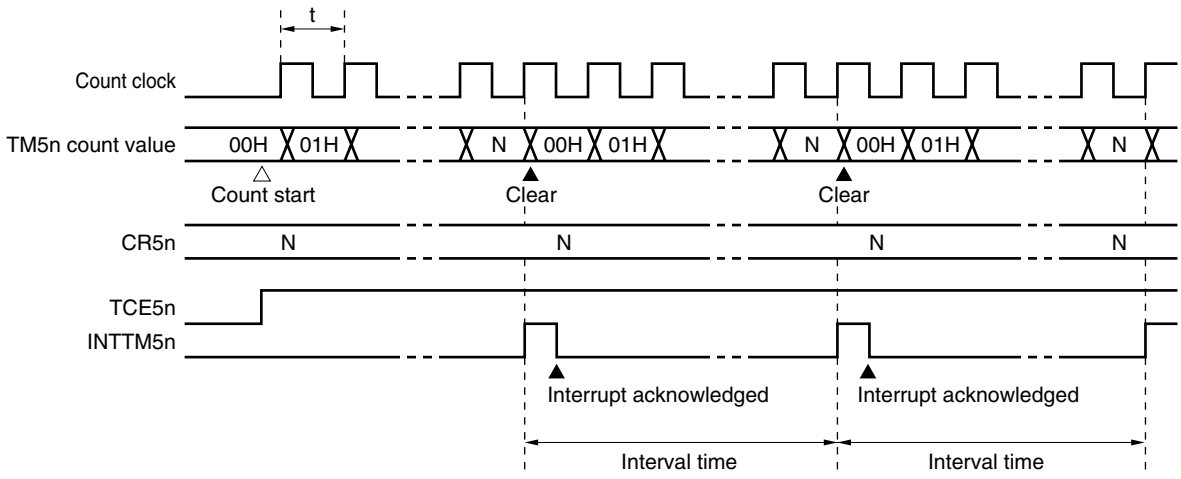
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n is generated repeatedly at the same interval.
Set TCE5n to 0 to stop the count operation.

Caution Do not write other values to CR5n during operation.

Figure 8-11. Interval Timer Operation Timing (1/2)

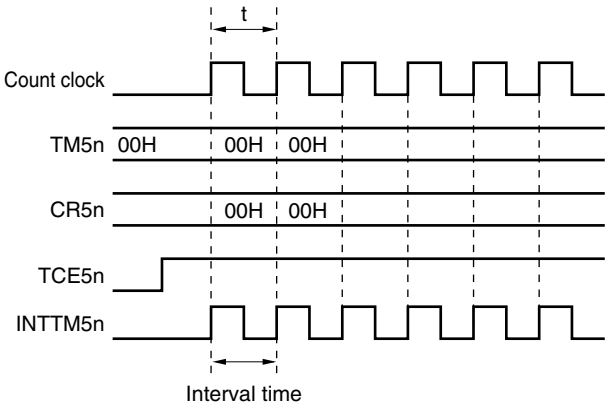
(a) Basic operation



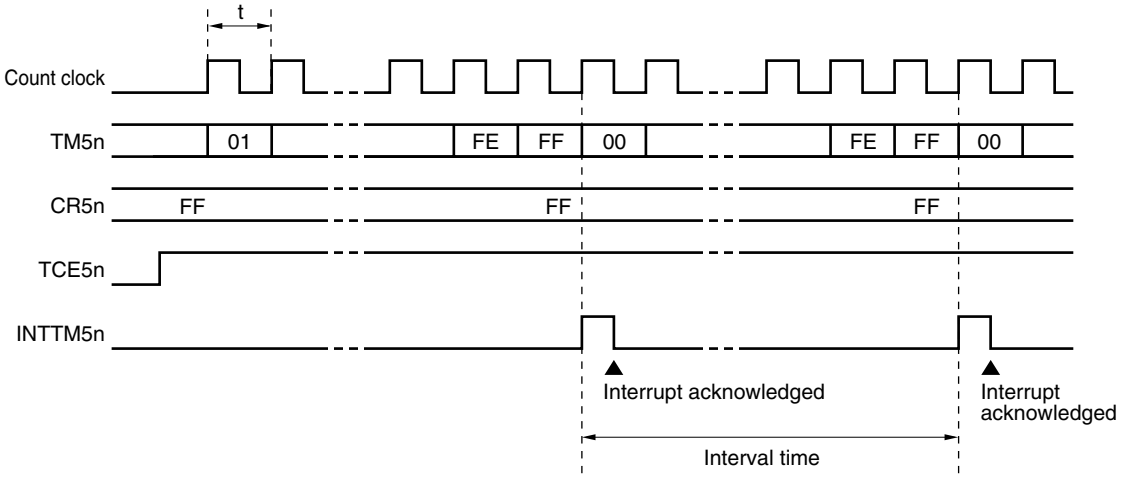
Remark Interval time = (N + 1) × t
N = 00H to FFH
n = 0, 1

Figure 8-11. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0, 1

8.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TI5n pin by 8-bit timer counter 5n (TM5n).

TM5n is incremented each time the valid edge specified by timer clock selection register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

Whenever the TM5n value matches the value of CR5n, INTTM5n is generated.

Setting

<1> Set each register.

- Set the port mode register (PM17 or PM33)[Note] to 1.
- TCL5n: Select TI5n pin input edge.
  TI5n pin falling edge → TCL5n = 00H
  TI5n pin rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 0000××00B × = Don't care)

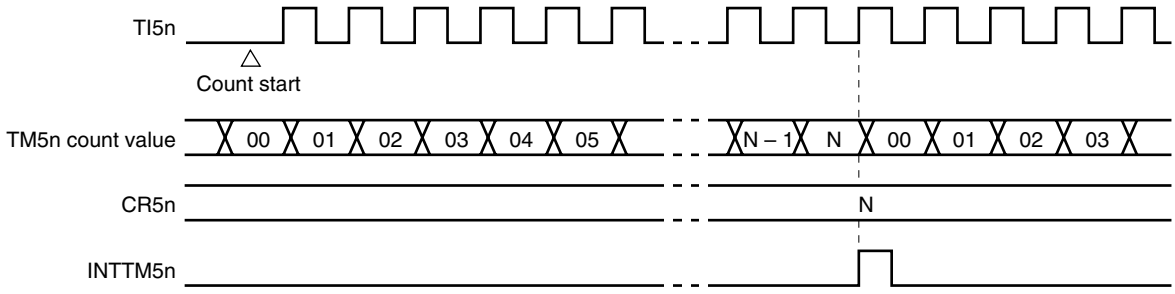
<2> When TCE5n = 1 is set, the number of pulses input from the TI5n pin is counted.

<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

Note 8-bit timer/event counter 50: PM17
8-bit timer/event counter 51: PM33

Figure 8-12. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0, 1

8.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is inverted at intervals determined by the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- Clear the port output latch (P17 or P33)[Note] and port mode register (PM17 or PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F inversion enabled
Timer output enabled
(TMC5n = 00001011B or 00000111B)

<2> After TCE5n = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n.

The frequency is as follows.

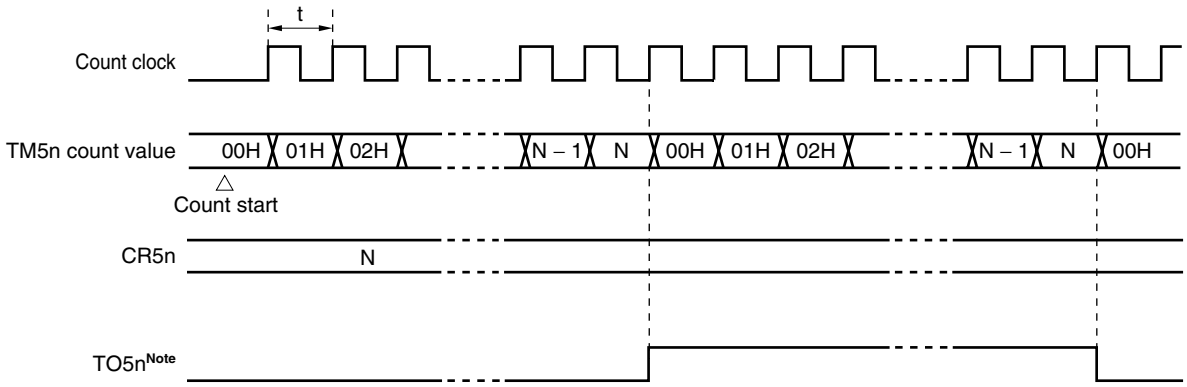
Frequency = 1/2t (N + 1)
(N: 00H to FFH)

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

Caution Do not write other values to CR5n during operation.

Remark n = 0, 1

Figure 8-13. Square-Wave Output Operation Timing



Note The initial value of TO5n output can be set by bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

8.4.4 PWM output operation

8-bit timer/event counter 5n operates as a PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

PWM output can be enabled/disabled with bit 0 (TOE5n) of TMC5n.

Caution In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

(1) PWM output basic operation

Setting

<1> Set each register.

- Clear the port output latch (P17 or P33)[Note] and port mode register (PM17 or PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Clear TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

PWM output operation

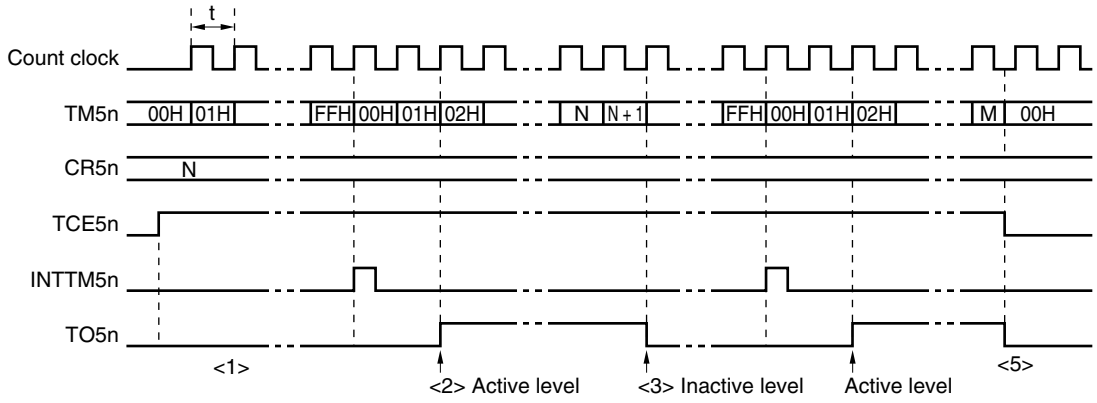
<1> PWM output (output from TO5n) outputs an inactive level until an overflow occurs.
<2> When an overflow occurs, the active level is output. The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).
<3> After the CR5n matches the count value, the inactive level is output until an overflow occurs again.
<4> Operations <2> and <3> are repeated until the count operation stops.
<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.
For details of timing, see **Figures 8-14** and **8-15**.
The cycle, active-level width, and duty are as follows.

- Cycle = 2^8t
- Active-level width = Nt
- Duty = N/2^8
  (N = 00H to FFH)

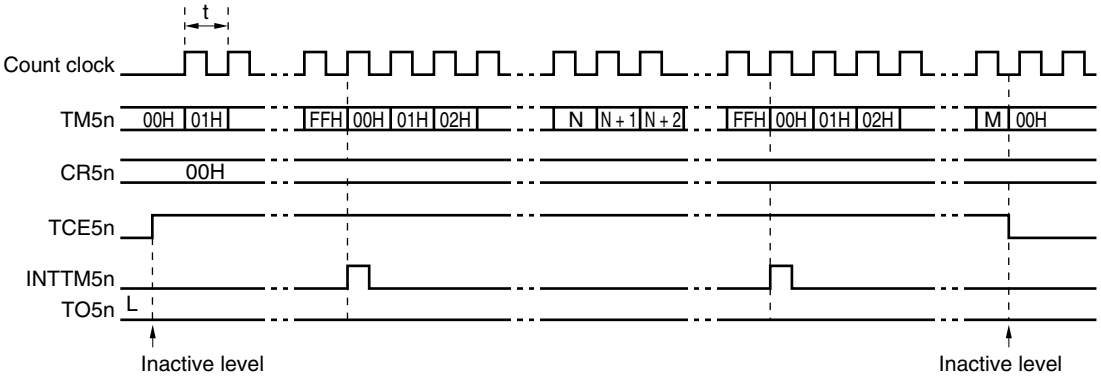
Remark n = 0, 1

Figure 8-14. PWM Output Operation Timing

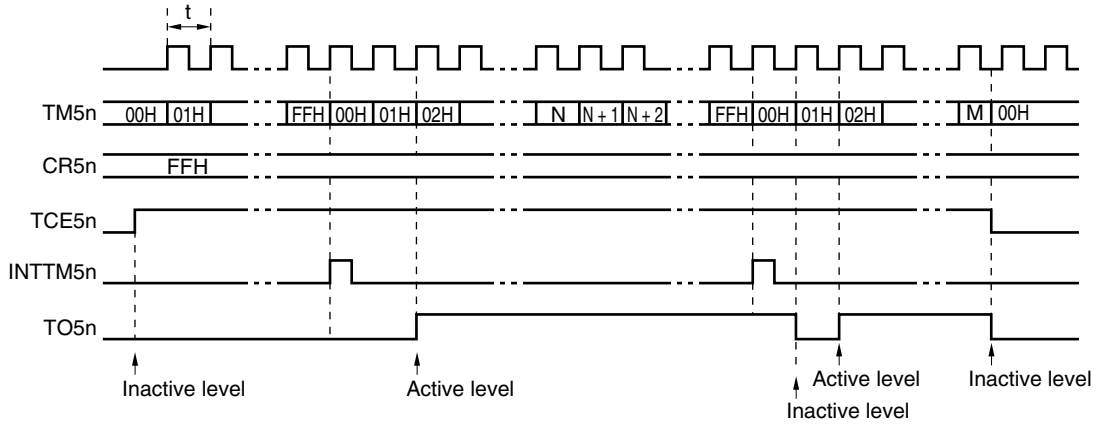
(a) Basic operation (active level = H)



(b) CR5n = 00H



(c) CR5n = FFH



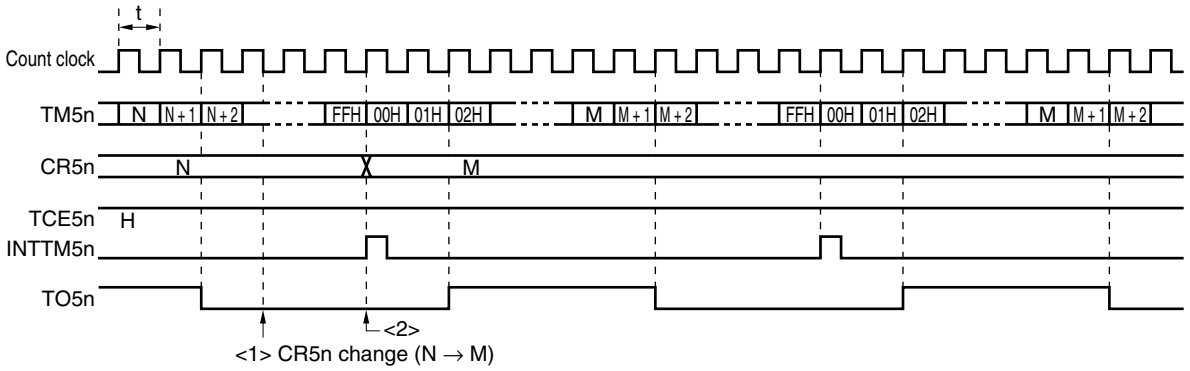
Remarks 1. <1> to <3> and <5> in Figure 8-14 (a) correspond to <1> to <3> and <5> in PWM output operation in **8.4.4 (1) PWM output basic operation**.

2. n = 0, 1

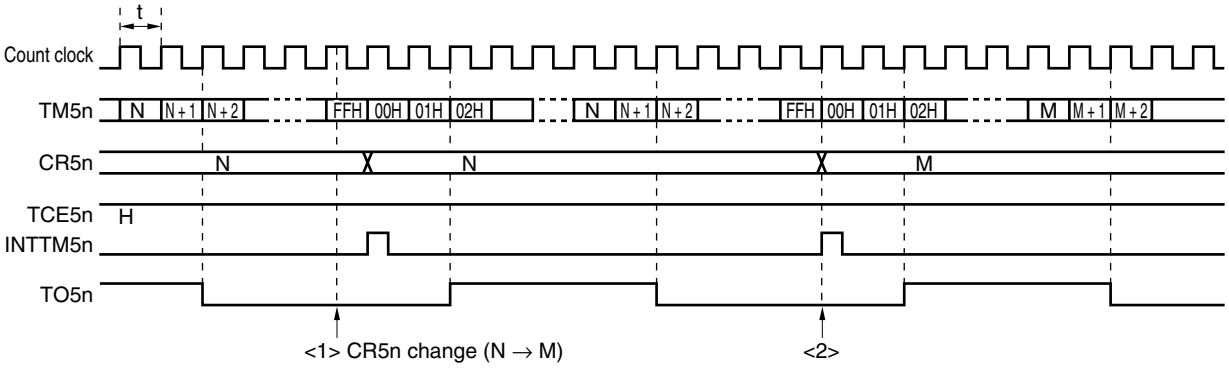
(2) Operation with CR5n changed

Figure 8-15. Timing of Operation with CR5n Changed

(a) CR5n value is changed from N to M before clock rising edge of FFH
→ Value is transferred to CR5n at overflow immediately after change.



(b) CR5n value is changed from N to M after clock rising edge of FFH
→ Value is transferred to CR5n at second overflow.



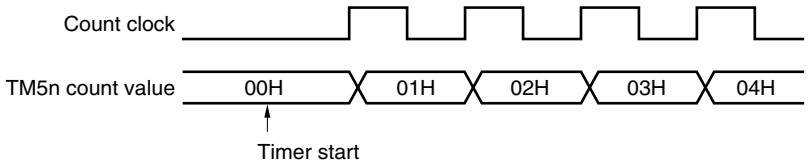
Caution When reading from CR5n between <1> and <2> in Figure 8-15, the value read differs from the actual value (read value: M, actual value of CR5n: N).

8.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counters 50 and 51 (TM50, TM51) are started asynchronously to the count clock.

Figure 8-16. 8-Bit Timer Counter 5n Start Timing



Remark n = 0, 1

CHAPTER 9 8-BIT TIMERS H0 AND H1

9.1 Functions of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 have the following functions.

- Interval timer
- PWM output mode
- Square-wave output
- Carrier generator mode (8-bit timer H1 only)

9.2 Configuration of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 include the following hardware.

Table 9-1. Configuration of 8-Bit Timers H0 and H1

Item	Configuration
Timer register	8-bit timer counter Hn
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	TOHn
Control registers	8-bit timer H mode register n (TMHMDn) 8-bit timer H carrier control register 1 (TMCYC1)[Note] Port mode register 1 (PM1) Port register 1 (P1)

Note 8-bit timer H1 only

Remark n = 0, 1

Figures 9-1 and 9-2 show the block diagrams.

Figure 9-1. Block Diagram of 8-Bit Timer H0

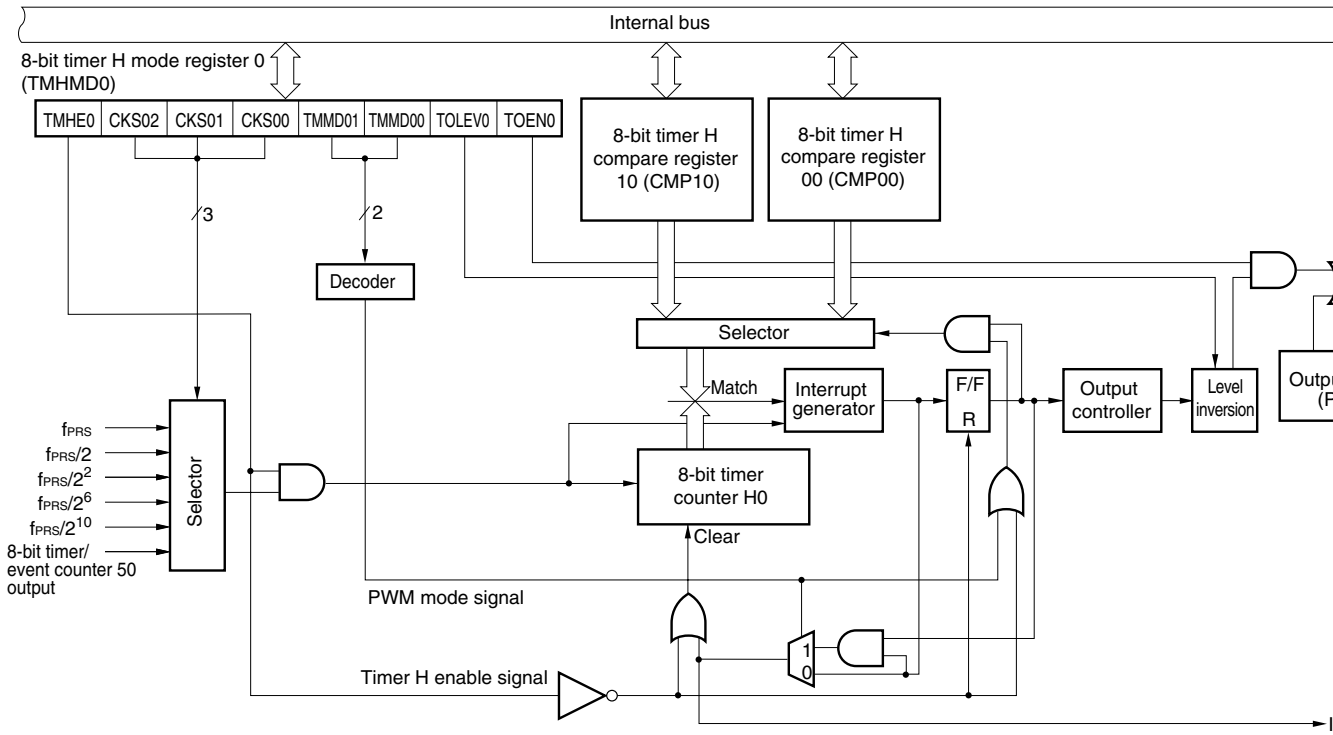
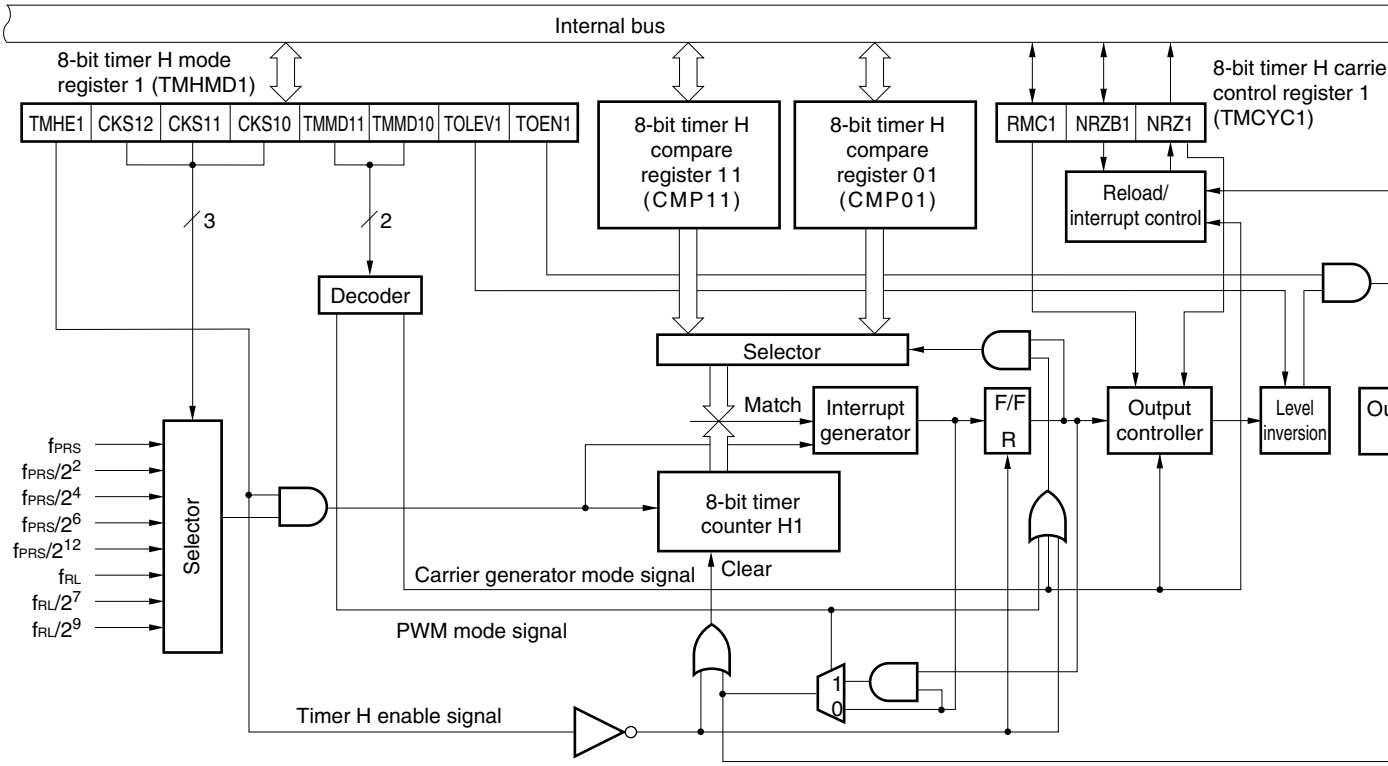


Figure 9-2. Block Diagram of 8-Bit Timer H1



(1) 8-bit timer H compare register 0n (CMP0n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in all of the timer operation modes.

This register constantly compares the value set to CMP0n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn) and inverts the output level of TOHn.

Rewrite the value of CMP0n while the timer is stopped (TMHEn = 0).

A reset signal generation clears this register to 00H.

Figure 9-3. Format of 8-Bit Timer H Compare Register 0n (CMP0n)

Address: FF02H (CMP00), FF1AH (CMP01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP0n (n = 0, 1)								

Caution CMP0n cannot be rewritten during timer count operation. CMP0n can be refreshed (the same value is written) during timer count operation.

(2) 8-bit timer H compare register 1n (CMP1n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in the PWM output mode and carrier generator mode.

In the PWM output mode, this register constantly compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, inverts the output level of TOHn. No interrupt request signal is generated.

In the carrier generator mode, the CMP1n register always compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn). At the same time, the count value is cleared.

CMP1n can be refreshed (the same value is written) and rewritten during timer count operation.

If the value of CMP1n is rewritten while the timer is operating, the new value is latched and transferred to CMP1n when the count value of the timer matches the old value of CMP1n, and then the value of CMP1n is changed to the new value. If matching of the count value and the CMP1n value and writing a value to CMP1n conflict, the value of CMP1n is not changed.

A reset signal generation clears this register to 00H.

Figure 9-4. Format of 8-Bit Timer H Compare Register 1n (CMP1n)

Address: FF0EH (CMP10), FF1BH (CMP11) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP1n (n = 0, 1)								

Caution In the PWM output mode and carrier generator mode, be sure to set CMP1n when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to CMP1n).

Remark n = 0, 1

9.3 Registers Controlling 8-Bit Timers H0 and H1

The following four registers are used to control 8-bit timers H0 and H1.

- 8-bit timer H mode register n (TMHMDn)
- 8-bit timer H carrier control register 1 (TMCYC1)[Note]
- Port mode register 1 (PM1)
- Port register 1 (P1)

Note 8-bit timer H1 only

(1) 8-bit timer H mode register n (TMHMDn)

This register controls the mode of timer H.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 9-5. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation (Counter is cleared to 0)
1	Enables timer count operation (Count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock selection				
				f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	f_{PRS}	4 MHz	8 MHz	10 MHz	20 MHz
0	0	1	$f_{PRS}/2$	2 MHz	4 MHz	5 MHz	10 MHz
0	1	0	$f_{PRS}/2^2$	1 MHz	2 MHz	2.5 MHz	5 MHz
0	1	1	$f_{PRS}/2^6$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
1	0	0	$f_{PRS}/2^{10}$	3.90 kHz	7.81 kHz	9.77 kHz	19.54 kHz
1	0	1	TM50 output[Note]				
Other than above			Setting prohibited				

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

Note When TM50 output as the count clock.

- Set to PWM mode (TMC506 = 1) after the following order to bellow.

<1>Set the count clock to make the duty = 50%.

<2>Start the operation of 8-bit timer/event counter 50.

- Set to Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0) after the following order to bellow.

<1>Enable the timer F/F inversion operation (TMC501 = 1).

<2>Start the operation of 8-bit timer/event counter 50.

It is not necessary to enable the TO50 pin as a timer output pin in any mode.

Cautions 1. When TMHE0 = 1, setting the other bits of TMHMD0 is prohibited. However, TMHMD0 can be refreshed (the same value is written).

2. In the PWM output mode, be sure to set 8-bit timer H compare register 10 (CMP10) when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP10).

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)

TMC501: Bit 1 of TMC50

Figure 9-6. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FFFAH After reset: 00H R/W

Symbol	<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} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	f_{PRS}	4 MHz	8 MHz	10 MHz	20 MHz
0	0	1	$f_{PRS}/2^2$	1 MHz	2 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^4$	500 kHz	1 MHz	625 kHz	1.25 MHz
0	1	1	$f_{PRS}/2^6$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz
1	0	0	$f_{PRS}/2^{12}$	0.97 kHz	1.95 kHz	2.44 kHz	4.88 kHz
1	0	1	$f_{RL}/2^7$	1.88 kHz (TYP.)			
1	1	0	$f_{RL}/2^9$	0.47 kHz (TYP.)			
1	1	1	f_{RL}	240 kHz (TYP.)			

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
0	1	Carrier generator mode
1	0	PWM output mode
0	0	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 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.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

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

(2) 8-bit timer H carrier control register 1 (TMCYC1)

This register controls the remote control output and carrier pulse output status of 8-bit timer H1.

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

Reset signal generation clears this register to 00H.

Figure 9-7. Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)

Address: FFEEH 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.

(3) Port mode register 1 (PM1)

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

When using the P15/TOH0 and P16/TOH1/INTP5 pins for timer output, clear PM15 and PM16 and the output latches of P15 and P16 to 0.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 9-8. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

9.4 Operation of 8-Bit Timers H0 and H1

9.4.1 Operation as interval timer/square-wave output

When 8-bit timer counter Hn and compare register 0n (CMP0n) match, an interrupt request signal (INTTMHn) is generated and 8-bit timer counter Hn is cleared to 00H.

Compare register 1n (CMP1n) is not used in interval timer mode. Since a match of 8-bit timer counter Hn and the CMP1n register is not detected even if the CMP1n register is set, timer output is not affected.

By setting bit 0 (TOENn) of timer H mode register n (TMHMDn) to 1, a square wave of any frequency (duty = 50%) is output from TOHn.

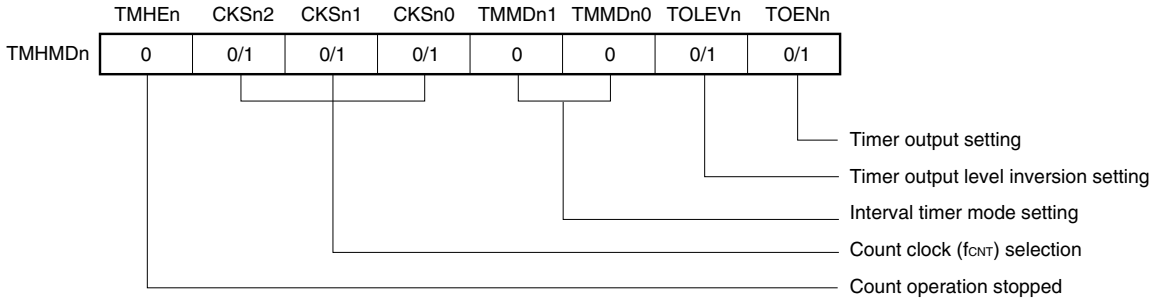
(1) Usage

Generates the INTTMHn signal repeatedly at the same interval.

<1> Set each register.

Figure 9-9. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register n (TMHMDn)



(ii) CMP0n register setting

- Compare value (N)

<2> Count operation starts when TMHEn = 1.

<3> When the values of 8-bit timer counter Hn and the CMP0n register match, the INTTMHn signal is generated and 8-bit timer counter Hn is cleared to 00H.

$$\text{Interval time} = (N + 1)/f_{CNT}$$

<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, clear TMHEn to 0.

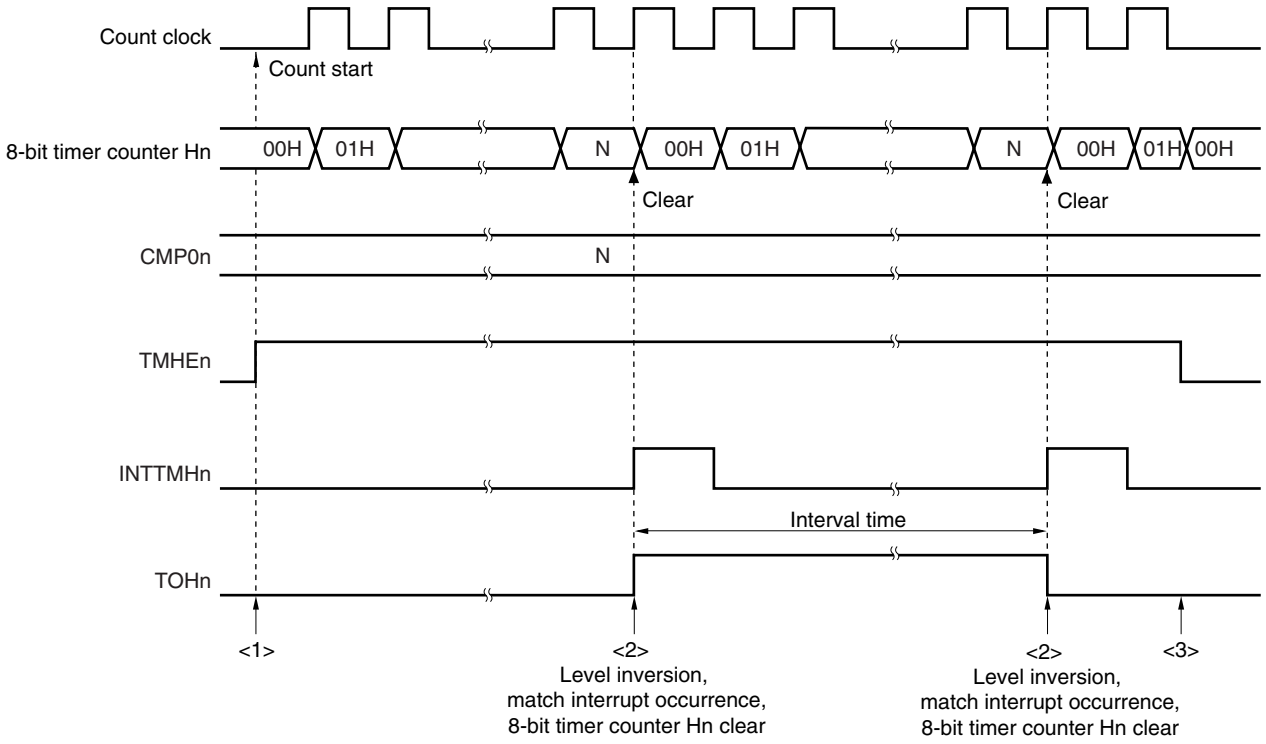
Remark n = 0, 1

(2) Timing chart

The timing of the interval timer/square-wave output operation is shown below.

Figure 9-10. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

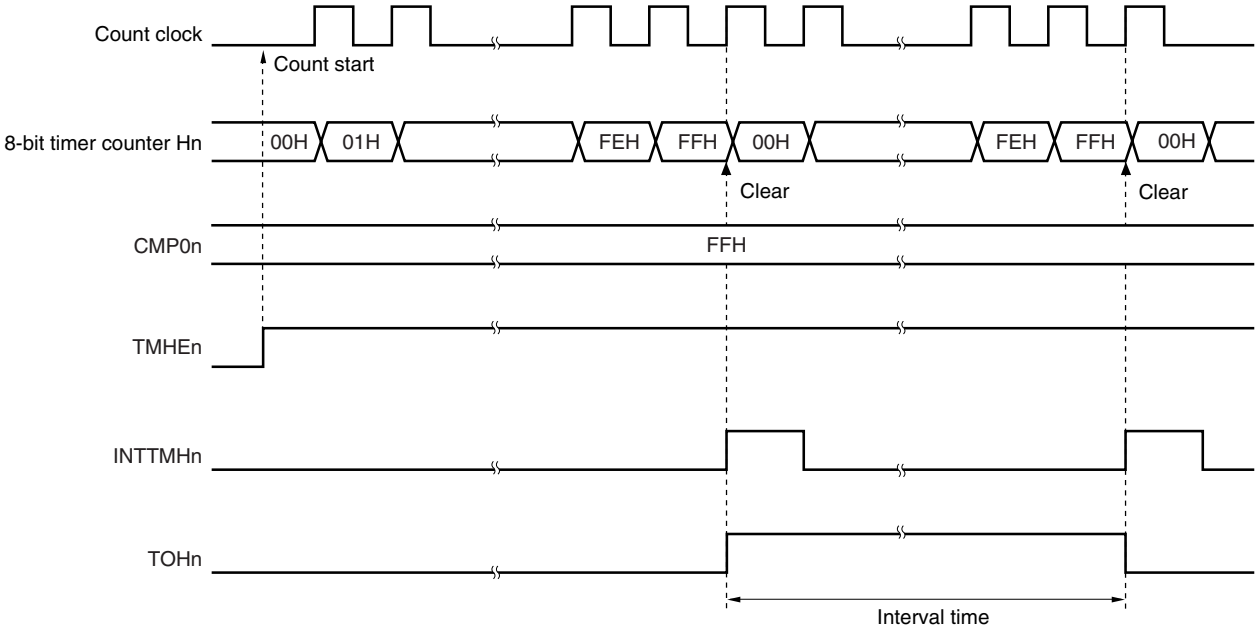
<2> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, the TOHn output level is inverted, and the INTTMHn signal is output.

<3> The INTTMHn signal and TOHn output become inactive by clearing the TMHEn bit to 0 during timer Hn operation. If these are inactive from the first, the level is retained.

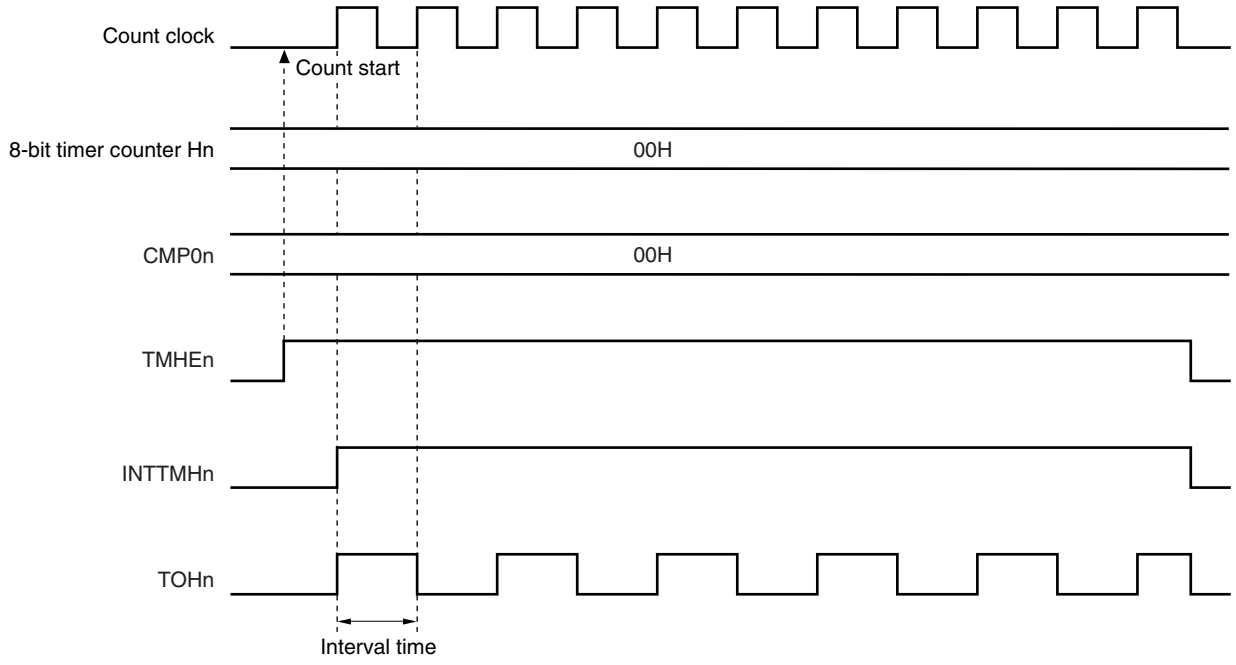
Remark n = 0, 1
N = 01H to FEH

Figure 9-10. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



Remark n = 0, 1

9.4.2 Operation as PWM output mode

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

8-bit timer compare register 0n (CMP0n) controls the cycle of timer output (TOHn). Rewriting the CMP0n register during timer operation is prohibited.

8-bit timer compare register 1n (CMP1n) controls the duty of timer output (TOHn). Rewriting the CMP1n register during timer operation is possible.

The operation in PWM output mode is as follows.

TOHn output becomes active and 8-bit timer counter Hn is cleared to 0 when 8-bit timer counter Hn and the CMP0n register match after the timer count is started. TOHn output becomes inactive when 8-bit timer counter Hn and the CMP1n register match.

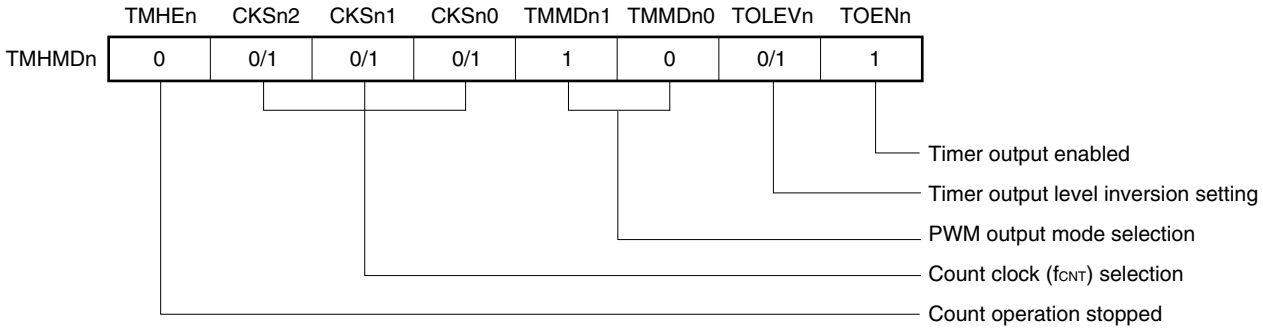
(1) Usage

In PWM output mode, a pulse for which an arbitrary duty and arbitrary cycle can be set is output.

<1> Set each register.

Figure 9-11. Register Setting in PWM Output Mode

(i) Setting timer H mode register n (TMHMDn)



(ii) Setting CMP0n register

- Compare value (N): Cycle setting

(iii) Setting CMP1n register

- Compare value (M): Duty setting

Remarks 1. n = 0, 1

2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of 8-bit timer counter Hn and the CMP0n register match, 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, and TOHn output becomes active. At the same time, the compare register to be compared with 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

<4> When 8-bit timer counter Hn and the CMP1n register match, TOHn output becomes inactive and the compare register to be compared with 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHEn = 0.

If the setting value of the CMP0n register is N, the setting value of the CMP1n register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

PWM pulse output cycle = $(N + 1)/f_{CNT}$
Duty = Active width : Total width of PWM = (M + 1) : (N + 1)

Cautions 1. In PWM output mode, three operation clocks (signal selected using the CKSn2 to CKSn0 bits of the TMHMDn register) are required to transfer the CMP1n register value after rewriting the register.

2. Be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

Remark n = 0, 1

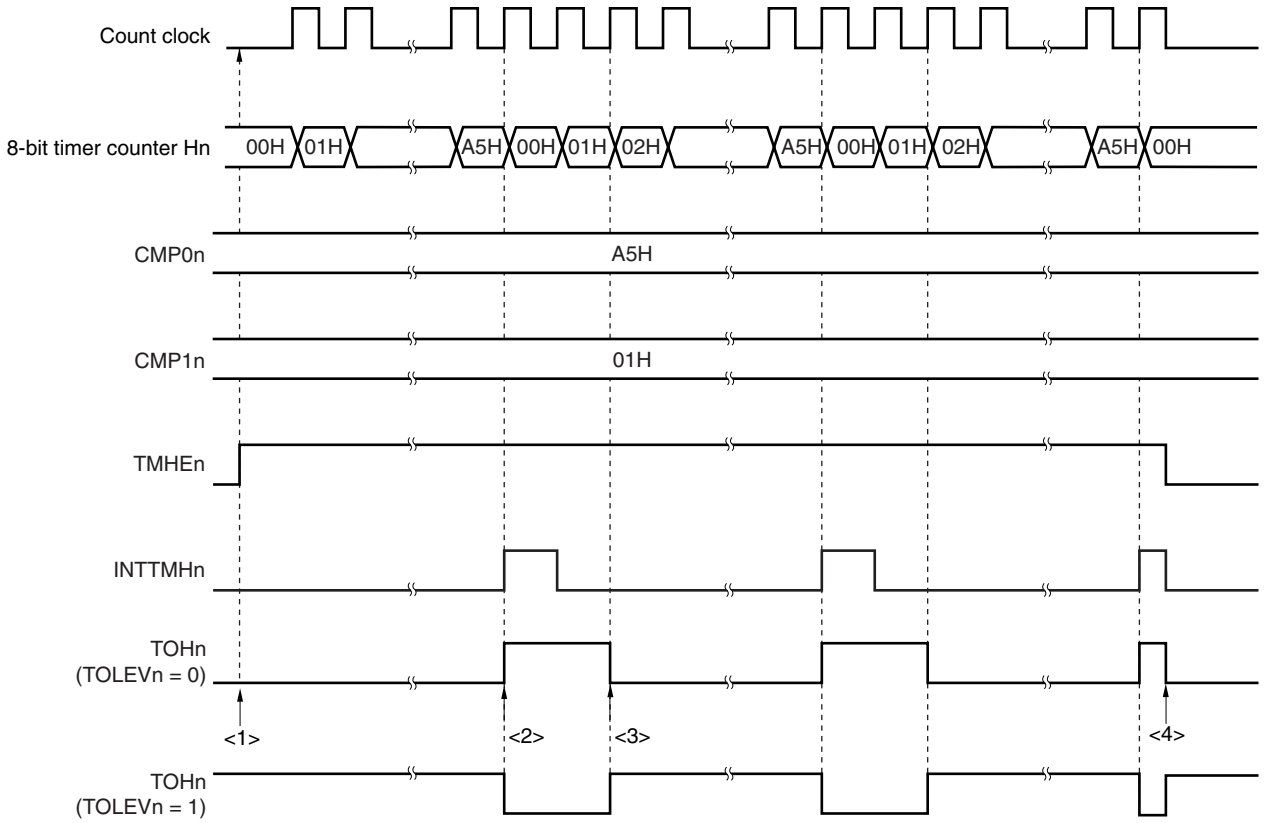
(2) Timing chart

The operation timing in PWM output mode is shown below.

Caution Make sure that the CMP1n register setting value (M) and CMP0n register setting value (N) are within the following range.
00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

Figure 9-12. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, TOHn output remains inactive (when TOLEVn = 0).

<2> When the values of 8-bit timer counter Hn and the CMP0n register match, the TOHn output level is inverted, the value of 8-bit timer counter Hn is cleared, and the INTTMHn signal is output.

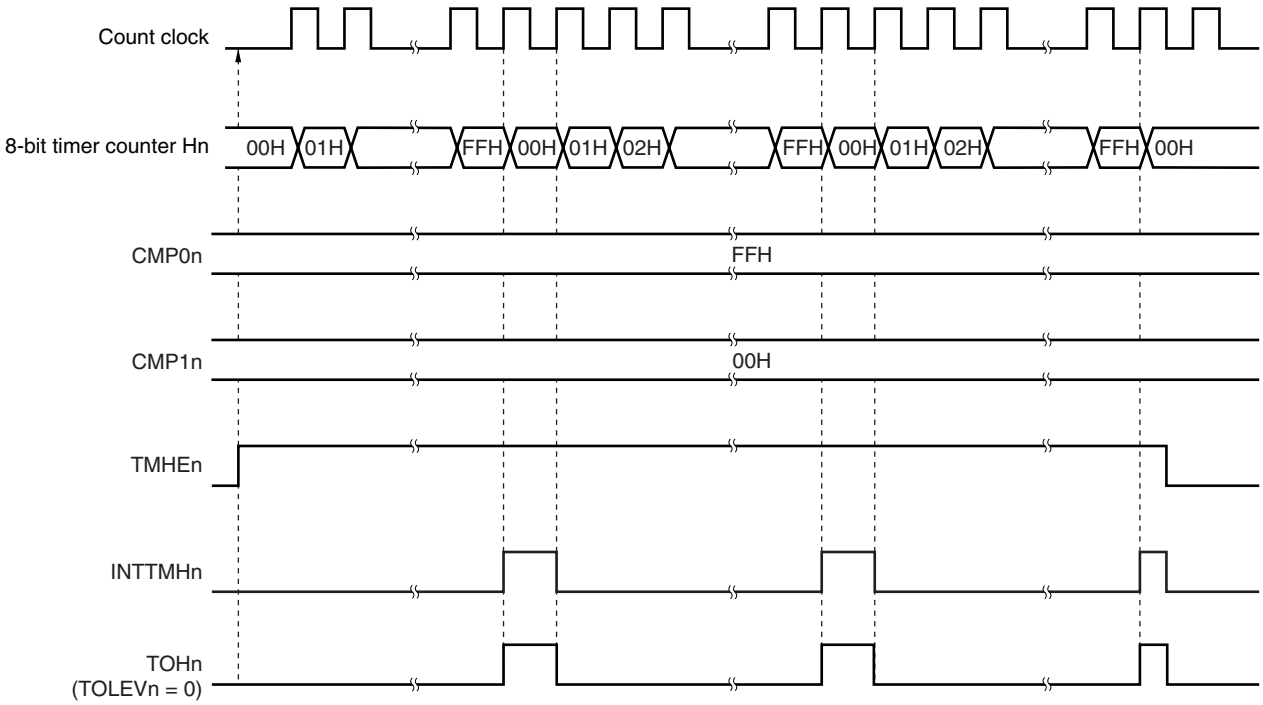
<3> When the values of 8-bit timer counter Hn and the CMP1n register match, the level of the TOHn output is returned. At this time, the 8-bit timer counter value is not cleared and the INTTMHn signal is not output.

<4> Clearing the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

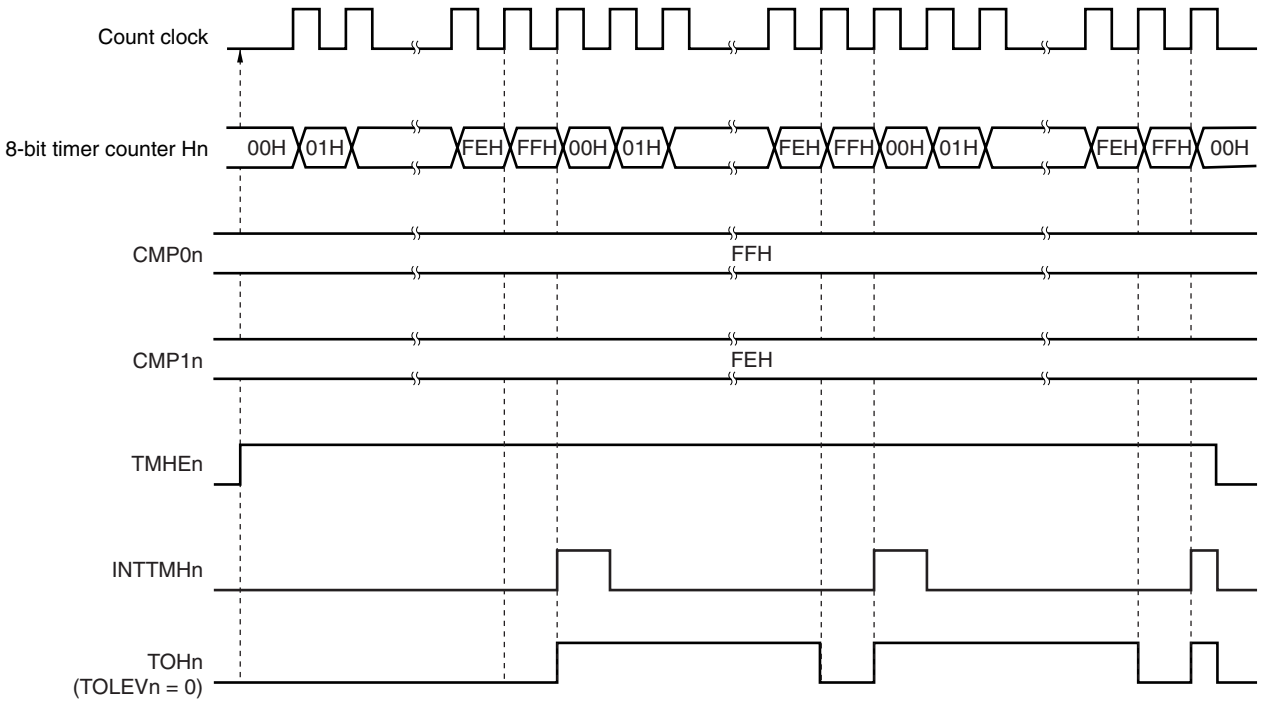
Remark n = 0, 1

Figure 9-12. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP0n = FFH, CMP1n = 00H



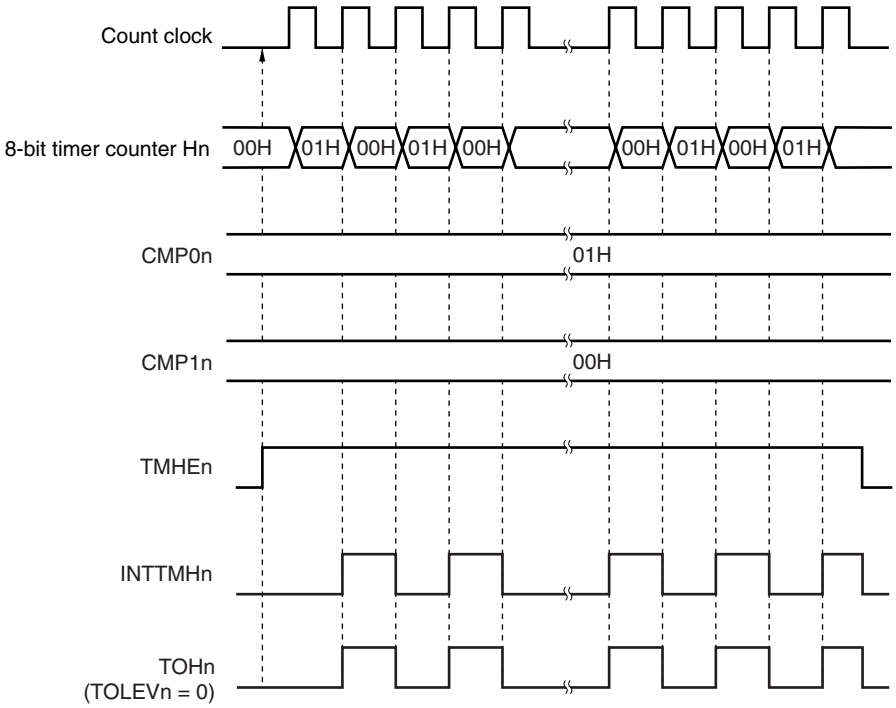
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0, 1

Figure 9-12. Operation Timing in PWM Output Mode (3/4)

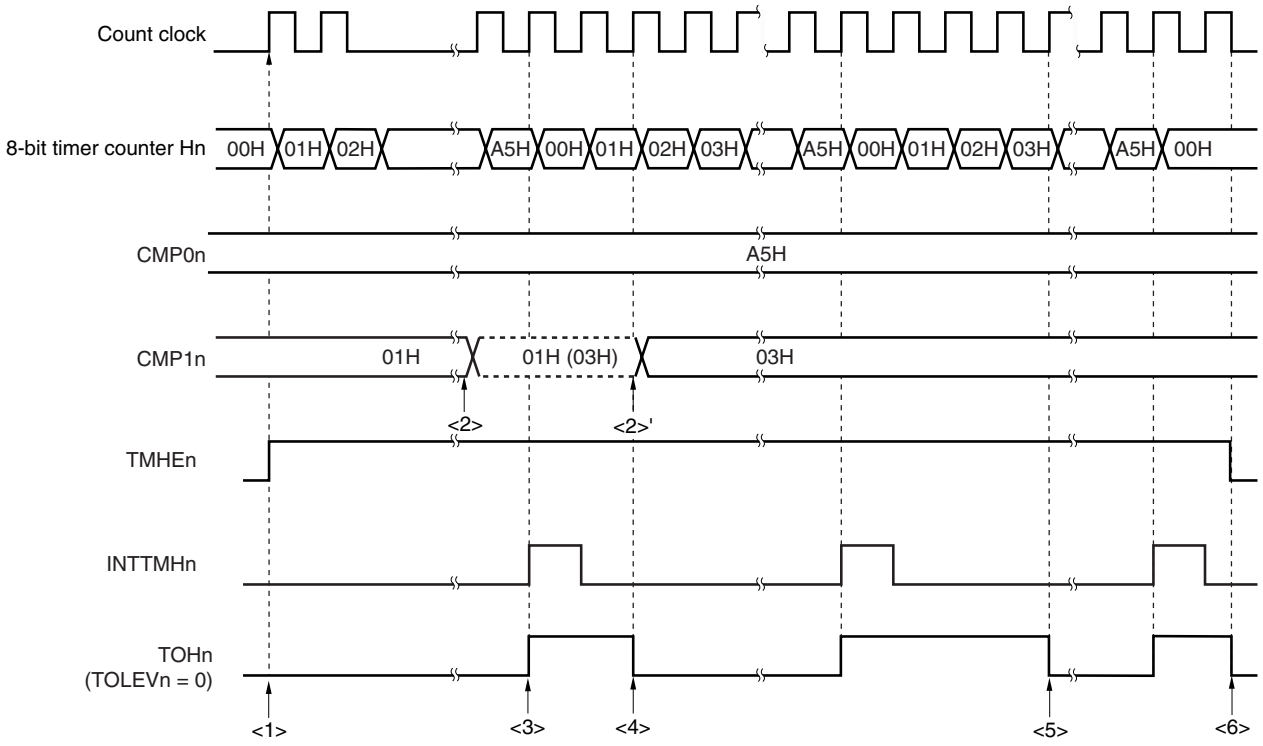
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0, 1

Figure 9-12. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP1n (CMP1n = 01H → 03H, CMP0n = A5H)



<1> The count operation is enabled by setting TMHEn = 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, the TOHn output remains inactive (when TOLEVn = 0).

<2> The CMP1n register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, the TOHn output becomes active, and the INTTMHn signal is output.

<4> If the CMP1n register value is changed, the value is latched and not transferred to the register. When the values of 8-bit timer counter Hn and the CMP1n register before the change match, the value is transferred to the CMP1n register and the CMP1n register value is changed (<2>').

However, three count clocks or more are required from when the CMP1n register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of 8-bit timer counter Hn and the CMP1n register after the change match, the TOHn output becomes inactive. 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<6> Clearing the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

Remark n = 0, 1

9.4.3 Carrier generator mode operation (8-bit timer H1 only)

The carrier clock generated by 8-bit timer H1 is output in the cycle set by 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

(1) Carrier generation

In carrier generator mode, 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

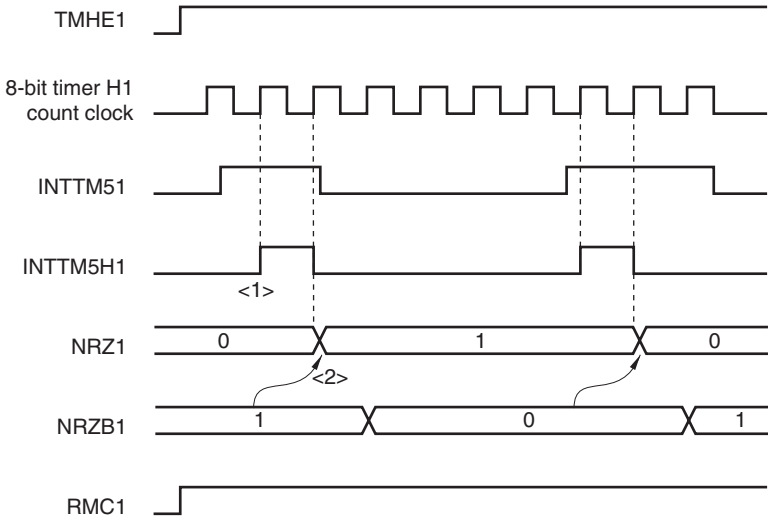
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of 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
1	0	Low-level output
1	1	Carrier pulse output

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 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 9-13. Transfer Timing



<1> The INTTM51 signal is synchronized with the count clock of 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.

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 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When 8-bit timer/event counter 51 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

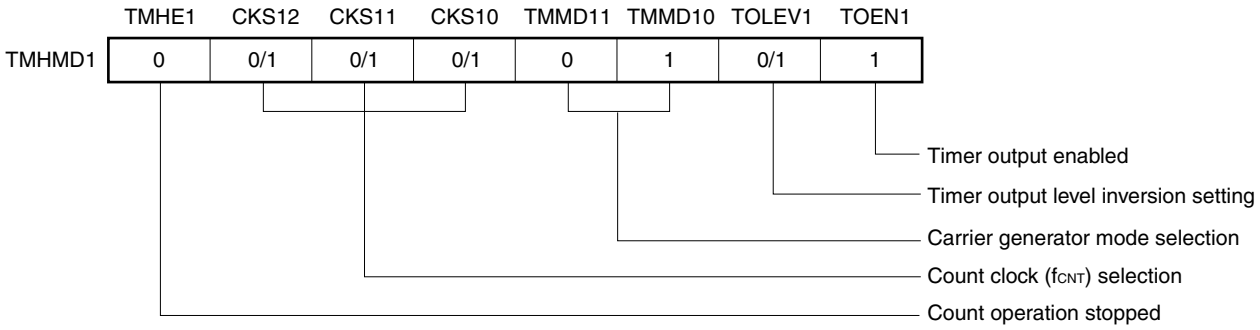
(3) Usage

Outputs an arbitrary carrier clock from the TOH1 pin.

<1> Set each register.

Figure 9-14. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)



(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... carrier output enable bit

(v) TCL51 and TMC51 register setting

- See **8.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51**.

<2> When TMHE1 = 1, 8-bit timer H1 starts counting.

<3> When TCE51 of 8-bit timer mode control register 51 (TMC51) is set to 1, 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 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared, and at the same time, the compare register to be compared with 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared, and at the same time, the compare register to be compared with 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 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> When the NRZ1 bit is high level, a carrier clock is output from the TOH1 pin.

<9> 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.

(4) Timing chart

The carrier output control timing is shown below.

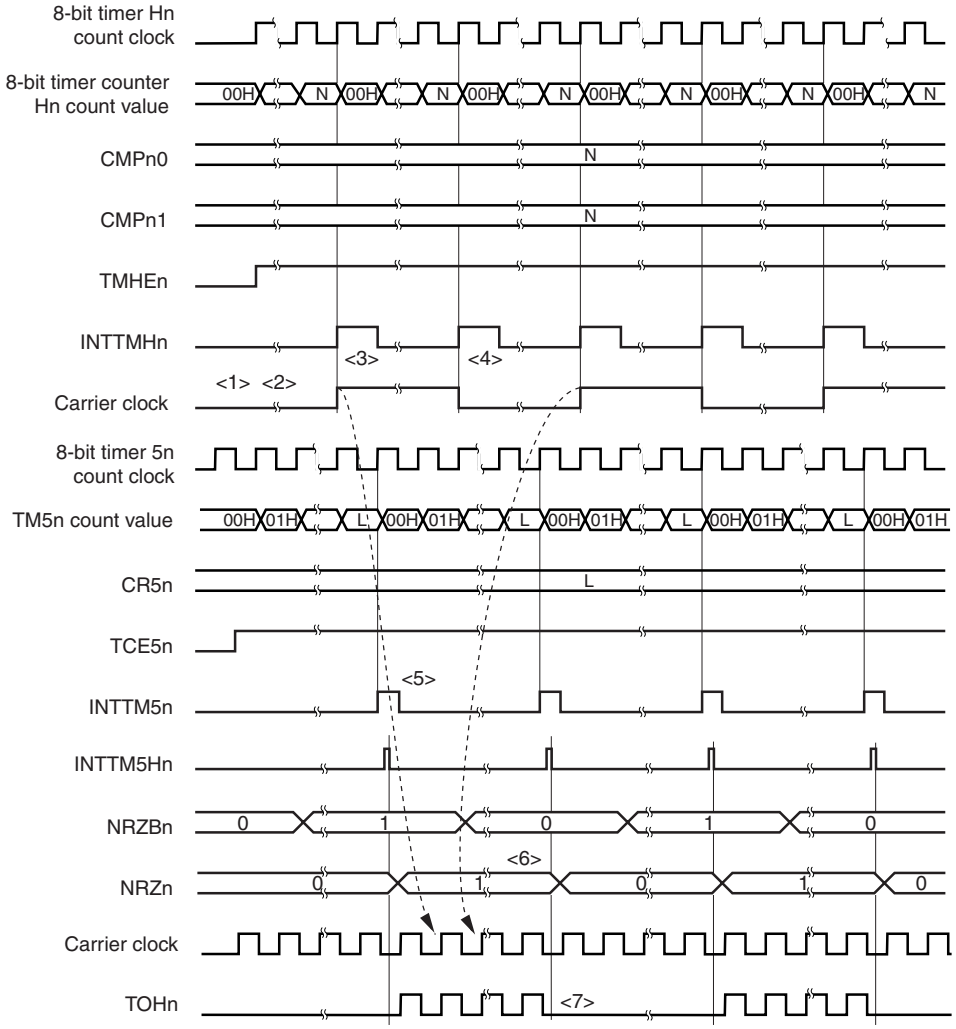
Cautions 1. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

2. In the carrier generator mode, three operating clocks (signal selected by CKS12 to CKS10 bits of TMHMD1 register) or more are required from when the CMP11 register value is changed to when the value is transferred to the register.

3. Be sure to set the RMC1 bit before the count operation is started.

Figure 9-15. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = N, CMP11 = N



<1> When TMHE1 = 0 and TCE51 = 0, 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, 8-bit timer counter H1 starts a count operation. At that time, the carrier clock is held at the inactive level.

<3> When the count value of 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 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of 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 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. 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.

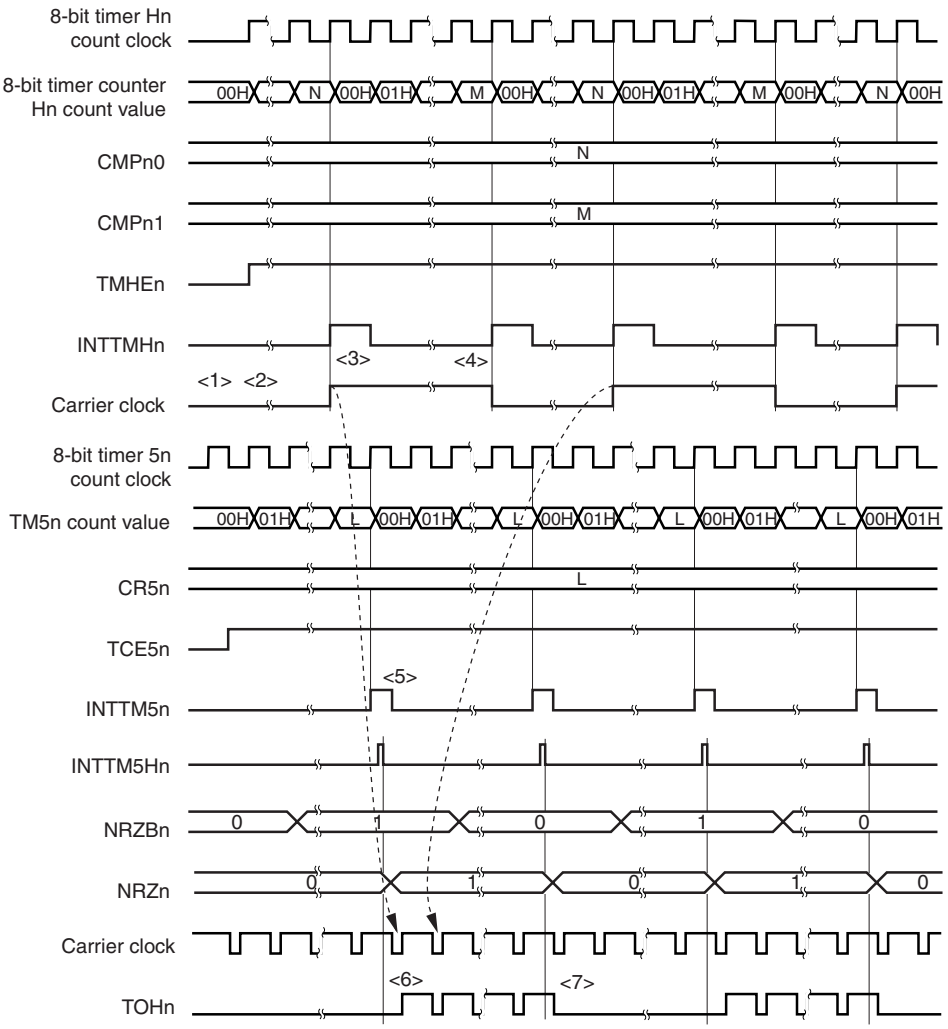
<5> When the INTTM51 signal is generated, it is synchronized with 8-bit timer H1 count clock and output as the INTTM5H1 signal.

<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.

Figure 9-15. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = N, CMP11 = M



<1> When TMHE1 = 0 and TCE51 = 0, 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, 8-bit timer counter H1 starts a count operation. At that time, the carrier clock is held at the inactive level.

<3> When the count value of 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 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of 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 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. 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.

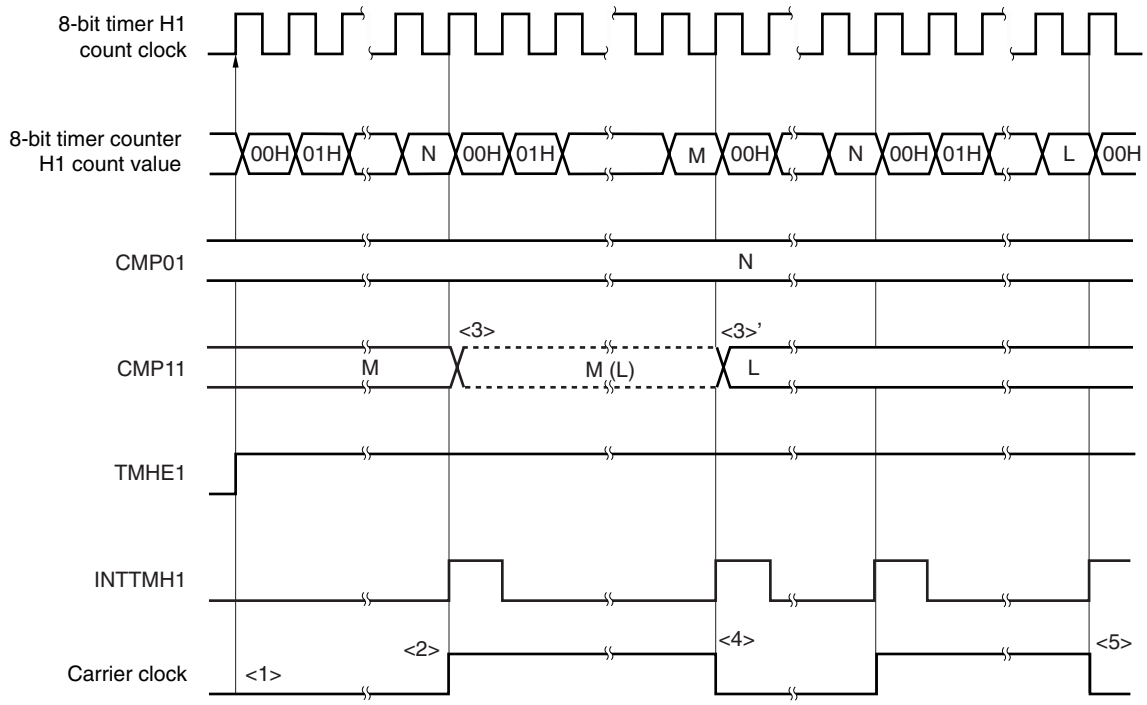
<5> When the INTTM51 signal is generated, it is synchronized with 8-bit timer H1 count clock and output as the INTTM5H1 signal.

<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).

Figure 9-15. Carrier Generator Mode Operation Timing (3/3)

(c) Operation when CMP11 is changed



<1> When TMHE1 = 1 is set, 8-bit timer H1 starts a count operation. At that time, the carrier clock is held at the inactive level.

<2> When the count value of 8-bit timer counter H1 matches the CMP01 register value, 8-bit timer counter H1 is cleared and the INTTMH1 signal is output.

<3> The CMP11 register can be rewritten during 8-bit timer H1 operation, however, the changed value (L) is latched. The CMP11 register is changed when the count value of 8-bit timer counter H1 and the CMP11 register value before the change (M) match (<3>').

<4> When the count value of 8-bit timer counter H1 and the CMP11 register value before the change (M) match, the INTTMH1 signal is output, the carrier signal is inverted, and 8-bit timer counter H1 is cleared to 00H.

<5> The timing at which the count value of 8-bit timer counter H1 and the CMP11 register value match again is indicated by the value after the change (L).

CHAPTER 10 WATCH TIMER

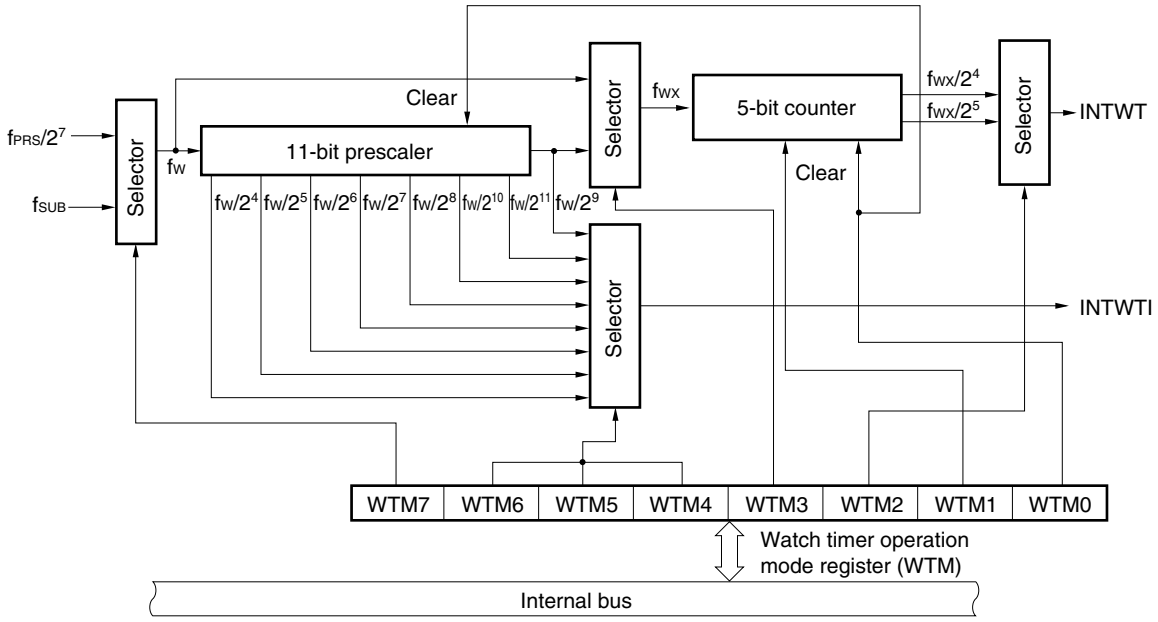
10.1 Functions of Watch Timer

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 10-1 shows the watch timer block diagram.

Figure 10-1. Block Diagram of Watch Timer



Remark f_{PRS}: Peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency
f_W: Watch timer clock frequency ($f_{PRS}/2^7$ or f_{SUB})
f_{WX}: f_W or $f_W/2^9$

(1) Watch timer

When the high-speed system clock or subsystem clock is used, interrupt requests (INTWT) are generated at preset intervals.

Table 10-1. Watch Timer Interrupt Time

Interrupt Time	When Operated at f_{SUB} = 32.768 kHz	When Operated at f_{PRS} = 4 MHz	When Operated at f_{PRS} = 5 MHz	When Operated at f_{PRS} = 10 MHz	When Operated at f_{PRS} = 20 MHz
$2^4/f_W$	488 μs	0.51 ms	410 μs	205 μs	102 μs
$2^5/f_W$	977 μs	1.03 ms	819 μs	410 μs	205 μs
$2^{13}/f_W$	0.25 s	0.26 s	0.210 s	0.105 s	520 μs
$2^{14}/f_W$	0.5 s	0.53 s	0.419 s	0.210 s	0.105 s

Remark f_{PRS}: Peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency
f_W: Watch timer clock frequency ($f_{PRS}/2^7$ or f_{SUB})

(2) Interval timer

Interrupt requests (INTWTI) are generated at preset time intervals.

Table 10-2. Interval Timer Interval Time

Interrupt Time	When Operated at f_{SUB} = 32.768 kHz	When Operated at f_{PRS} = 4 MHz	When Operated at f_{PRS} = 5 MHz	When Operated at f_{PRS} = 10 MHz	When Operated at f_{PRS} = 20 MHz
$2^4/f_W$	488 μs	0.51 ms	410 μs	205 μs	102 μs
$2^5/f_W$	977 μs	1.03 ms	820 μs	410 μs	205 μs
$2^6/f_W$	1.95 ms	2.05 ms	1.64 ms	820 μs	410 μs
$2^7/f_W$	3.91 ms	4.1 ms	3.28 ms	1.64 ms	820 μs
$2^8/f_W$	7.81 ms	8.2 ms	6.55 ms	3.28 ms	1.64 ms
$2^9/f_W$	15.6 ms	16.4 ms	13.1 ms	6.55 ms	3.28 ms
$2^{10}/f_W$	31.3 ms	32.75 ms	26.2 ms	13.1 ms	6.55 ms
$2^{11}/f_W$	62.5 ms	65.55 ms	52.4 ms	26.2 ms	13.1 ms

Remark f_{PRS}: Peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency
f_W: Watch timer clock frequency ($f_{PRS}/2^7$ or f_{SUB})

10.2 Configuration of Watch Timer

The watch timer includes the following hardware.

Table 10-3. Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control register	Watch timer operation mode register (WTM)

10.3 Register Controlling Watch Timer

The watch timer is controlled by the watch timer operation mode register (WTM).

- **Watch timer operation mode register (WTM)**
  This register sets the watch timer count clock, enables/disables operation, prescaler interval time, and 5-bit counter operation control.
  WTM is set by a 1-bit or 8-bit memory manipulation instruction.
  Reset signal generation clears WTM to 00H.

Figure 10-2. Format of Watch Timer Operation Mode Register (WTM)

Address: FF8FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	Watch timer count clock selection (f_W)					
		f_{SUB} = 32.768 kHz	f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	$f_{PRS}/2^7$	–	31.25 kHz	62.5 kHz	78.125 kHz	156.25 kHz
1	f_{SUB}	32.768 kHz	–			

WTM6	WTM5	WTM4	Prescaler interval time selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
1	1	0	$2^{10}/f_W$
1	1	1	$2^{11}/f_W$

WTM3	WTM2	Interrupt time selection
0	0	$2^{14}/f_W$
0	1	$2^{13}/f_W$
1	0	$2^5/f_W$
1	1	$2^4/f_W$

WTM1	5-bit counter operation control
0	Clear after operation stop
1	Start

WTM0	Watch timer operation enable
0	Operation stop (clear both prescaler and 5-bit counter)
1	Operation enable

Caution Do not change the count clock and interval time (by setting bits 4 to 7 (WTM4 to WTM7) of WTM) during watch timer operation.

Remarks 1. f_W: Watch timer clock frequency ($f_{PRS}/2^7$ or f_{SUB})

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{SUB}: Subsystem clock frequency

10.4 Watch Timer Operations

10.4.1 Watch timer operation

The watch timer generates an interrupt request signal (INTWT) at a specific time interval by using the peripheral hardware clock or subsystem clock.

When bit 0 (WTM0) and bit 1 (WTM1) of the watch timer operation mode register (WTM) are set to 1, the count operation starts. When these bits are cleared to 0, the 5-bit counter is cleared and the count operation stops.

When the interval timer is simultaneously operated, zero-second start can be achieved only for the watch timer by clearing WTM1 to 0. In this case, however, the 11-bit prescaler is not cleared. Therefore, an error up to $2^9 \times 1/f_W$ seconds occurs in the first overflow (INTWT) after zero-second start.

The interrupt request is generated at the following time intervals.

Table 10-4. Watch Timer Interrupt Time

WTM3	WTM2	Interrupt Time Selection	When Operated at f_{SUB} = 32.768 kHz (WTM7 = 1)	When Operated at f_{PRS} = 4 MHz (WTM7 = 0)	When Operated at f_{PRS} = 5 MHz (WTM7 = 0)	When Operated at f_{PRS} = 10 MHz (WTM7 = 0)	When Operated at f_{PRS} = 20 MHz (WTM7 = 0)
0	0	$2^{14}/f_W$	0.5 s	0.53 s	0.419 s	0.210 s	0.105 s
0	1	$2^{13}/f_W$	0.25 s	0.26 s	0.210 s	0.105 s	52.5 ms
1	0	$2^5/f_W$	977 μs	1.03 ms	819 μs	410 μs	205 μs
1	1	$2^4/f_W$	488 μs	0.51 ms	410 μs	205 μs	102 μs

Remarks 1. f_W: Watch timer clock frequency ($f_{PRS}/2^7$ or f_{SUB})

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{SUB}: Subsystem clock frequency

10.4.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt requests (INTWTI) repeatedly at an interval of the preset count value.

The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer operation mode register (WTM).

When bit 0 (WTM0) of the WTM is set to 1, the count operation starts. When this bit is set to 0, the count operation stops.

Table 10-5. Interval Timer Interval Time

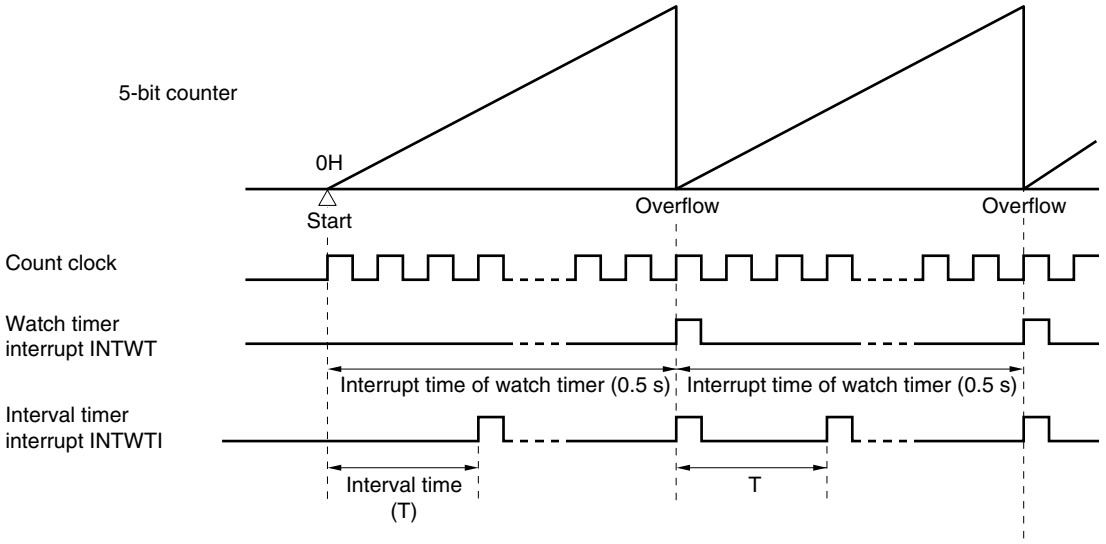
WTM6	WTM5	WTM4	Interval Time	When Operated at f_{SUB} = 32.768 kHz (WTM7 = 1)	When Operated at f_{PRS} = 4 MHz (WTM7 = 0)	When Operated at f_{PRS} = 5 MHz (WTM7 = 0)	When Operated at f_{PRS} = 10 MHz (WTM7 = 0)	When Operated at f_{PRS} = 20 MHz (WTM7 = 0)
0	0	0	$2^4/f_W$	488 μs	0.51 ms	410 μs	205 μs	102 μs
0	0	1	$2^5/f_W$	977 μs	1.03 ms	820 μs	410 μs	205 μs
0	1	0	$2^6/f_W$	1.95 ms	2.05 ms	1.64 ms	820 μs	410 μs
0	1	1	$2^7/f_W$	3.91 ms	4.1 ms	3.28 ms	1.64 ms	820 μs
1	0	0	$2^8/f_W$	7.81 ms	8.2 ms	6.55 ms	3.28 ms	1.64 ms
1	0	1	$2^9/f_W$	15.6 ms	16.4 ms	13.1 ms	6.55 ms	3.28 ms
1	1	0	$2^{10}/f_W$	31.3 ms	32.75 ms	26.2 ms	13.1 ms	6.55 ms
1	1	1	$2^{11}/f_W$	62.5 ms	65.55 ms	52.4 ms	26.2 ms	13.1 ms

Remarks 1. f_W: Watch timer clock frequency ($f_{PRS}/2^7$ or f_{SUB})

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{SUB}: Subsystem clock frequency

Figure 10-3. Operation Timing of Watch Timer/Interval Timer



Remark f_W: Watch timer clock frequency

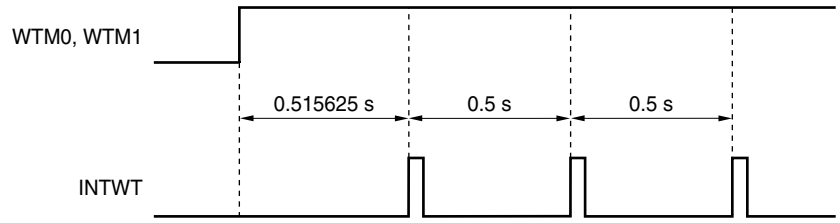
Figures in parentheses are for operation with f_W = 32.768 kHz (WTM7 = 1, WTM3, WTM2 = 0, 0)

10.5 Cautions for Watch Timer

When operation of the watch timer and 5-bit counter is enabled by the watch timer mode control register (WTM) (by setting bits 0 (WTM0) and 1 (WTM1) of WTM to 1), the interval until the first interrupt request (INTWT) is generated after the register is set does not exactly match the specification made with bits 2 and 3 (WTM2, WTM3) of WTM. Subsequently, however, the INTWT signal is generated at the specified intervals.

Figure 10-4. Example of Generation of Watch Timer Interrupt Request (INTWT) (When Interrupt Period = 0.5 s)

It takes 0.515625 seconds for the first INTWT to be generated ($2^9 \times 1/32768 = 0.015625$ s longer). INTWT is then generated every 0.5 seconds.



CHAPTER 11 WATCHDOG TIMER

11.1 Functions of Watchdog Timer

The watchdog timer operates on the internal low-speed oscillator clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to WDTE
- If data is written to WDTE during a window close period
- If the instruction is fetched from an area not set by the IMS and IXS registers (detection of an invalid check while the CPU hangs up)
- If the CPU accesses an area that is not set by the IMS and IXS registers (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, see **CHAPTER 19 RESET FUNCTION**.

11.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 11-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

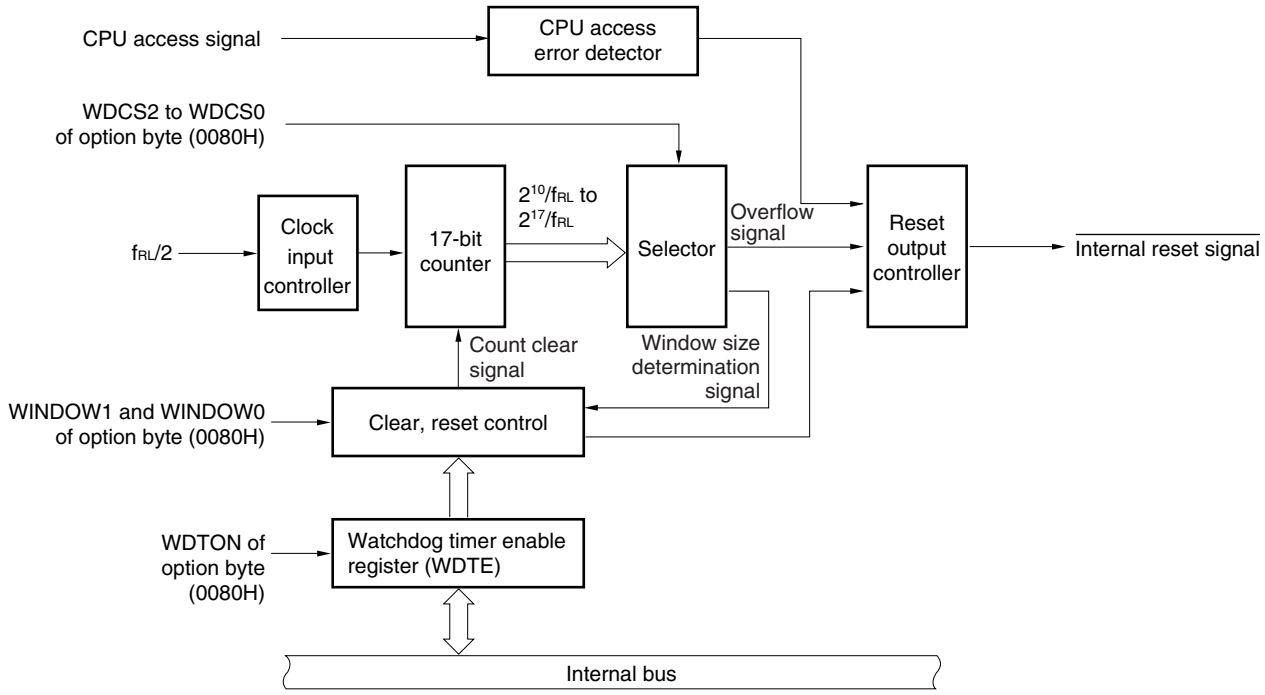
How the counter operation is controlled, overflow time, and window open period are set by the option byte.

Table 11-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (0080H)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)

Remark For the option byte, see **CHAPTER 23 OPTION BYTE**.

Figure 11-1. Block Diagram of Watchdog Timer



11.3 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 11-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FF9BH 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)).

11.4 Operation of Watchdog Timer

11.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (0080H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (0080H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 23**).

<R>

WDTON	Operation Control of Watchdog Timer Counter/Illegal Access Detection
0	Counter operation disabled (counting stopped after reset), Illegal access detection operation disabled.
1	Counter operation enabled (counting started after reset), Illegal access detection operation enabled.

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H) (for details, see **11.4.2** and **CHAPTER 23**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, see **11.4.3** 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 period other than the window open period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, an internal reset signal is generated. An internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
   - If data other than “ACH” is written to WDTE
   - If the instruction is fetched from an area not set by the IMS and IXS registers (detection of an invalid check during a CPU program loop)
   - If the CPU accesses an area not set by the IMS and IXS registers (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 $2/f_{RL}$ seconds.

3. The watchdog timer can be cleared immediately before the count value overflows (FFFFH).

Cautions 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (LSROSC) of the option byte.

	LSROSC = 0 (Internal Low-Speed Oscillator Can Be Stopped by Software)	LSROSC = 1 (Internal Low-Speed Oscillator Cannot Be Stopped)
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If LSROSC = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is not cleared to 0 but starts counting from the value at which it was stopped.

If oscillation of the internal low-speed oscillator is stopped by setting LSRSTOP (bit 1 of the internal oscillator mode register (RCM) = 1) when LSROSC = 0, the watchdog timer stops operating. At this time, the counter is not cleared to 0.

5. The watchdog timer continues its operation during self-programming and EEPROM™ emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

11.4.2 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.

If an overflow occurs, an internal reset signal is generated. If “ACH” is written to WDTE during the window open period before the overflow time, the present count is cleared and the watchdog timer starts counting again.

The following overflow time is set.

Table 11-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

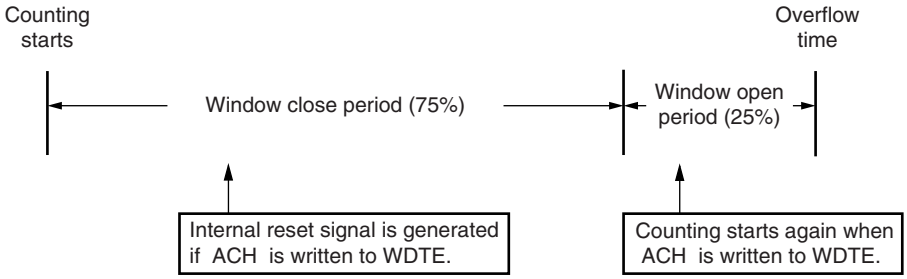
2. ( ): f_{RL} = 264 kHz (MAX.)

11.4.3 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 11-4. 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 $2^{10}/f_{RL}$, the window close time and open time are as follows.

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 3.56 ms	0 to 2.37 ms	0 to 0.119 ms	None
Window open time	3.56 to 3.88 ms	2.37 to 3.88 ms	0.119 to 3.88 ms	0 to 3.88 ms

<When window open period is 25%>

- Overflow time:
  $2^{10}/f_{RL}$ (MAX.) = 2^{10}/264 kHz (MAX.) = 3.88 ms
- Window close time:
  0 to $2^{10}/f_{RL}$ (MIN.) × (1 − 0.25) = 0 to 2^{10}/216 kHz (MIN.) × 0.75 = 0 to 3.56 ms
- Window open time:
  $2^{10}/f_{RL}$ (MIN.) × (1 − 0.25) to $2^{10}/f_{RL}$ (MAX.) = 2^{10}/216 kHz (MIN.) × 0.75 to 2^{10}/264 kHz (MAX.)
  = 3.56 to 3.88 ms

CHAPTER 12 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

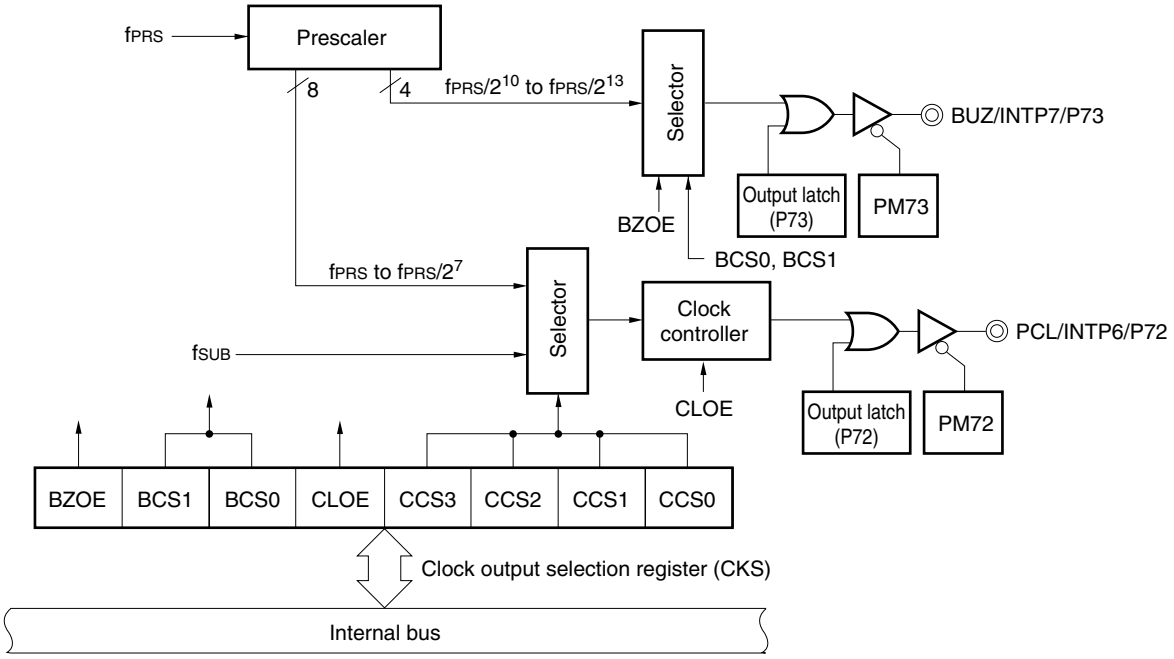
12.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

In addition, the buzzer output is intended for square-wave output of buzzer frequency selected with CKS.

Figure 12-1 shows the block diagram of clock output/buzzer output controller.

Figure 12-1. Block Diagram of Clock Output/Buzzer Output Controller



12.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 12-1. Clock Output/Buzzer Output Controller Configuration

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 7 (PM7) Port register 7 (P7)

12.3 Register Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output selection register (CKS)
- Port mode register 7 (PM7)

(1) Clock output selection register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

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

Reset signal generation clears CKS to 00H.

Figure 12-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	<7>	6	5	<4>	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Clock division circuit operation stopped. BUZ fixed to low level.
1	Clock division circuit operation enabled. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection		
			f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	$f_{PRS}/2^{10}$	9.77 kHz	19.54 kHz
0	1	$f_{PRS}/2^{11}$	4.88 kHz	9.77 kHz
1	0	$f_{PRS}/2^{12}$	2.44 kHz	4.88 kHz
1	1	$f_{PRS}/2^{13}$	1.22 kHz	2.44 kHz

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection[Note]			
					f_{SUB} = 32.768 kHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}[Note1]	–	10 MHz	Setting prohibited[Note2]
0	0	0	1	$f_{PRS}/2$		5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$		2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$		1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$		625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$		312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$		156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$		78.125 kHz	156.25 kHz
1	0	0	0	f_{SUB}	32.768 kHz	–	
Other than above				Setting prohibited			

Notes 1. If the peripheral hardware clock operates on the internal high-speed oscillation clock when 1.8 V ≤ V_{DD} < 2.7 V, setting CCS3 = CCS2 = CCS1 = CCS0 = 0 (output clock of PCL: f_{PRS}) is prohibited.

2. The PCL output clock prohibits settings if they exceed 10 MHz.

Cautions 1. Set BCS1 and BCS0 when the buzzer output operation is stopped (BZOE = 0).

2. Set CCS3 to CCS0 while the clock output operation is stopped (CLOE = 0).

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{SUB}: Subsystem clock frequency

(2) Port mode register 7 (PM7)

This register sets port 7 input/output in 1-bit units.

When using the P72/INTP6/PCL pin for clock output and the P73/INTP7/BUZ pin for buzzer output, set PM72, PM73 and the output latch of P72, P73 to 0.

PM7 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM7 to FFH.

Figure 12-3. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 Clock Output/Buzzer Output Controller Operations

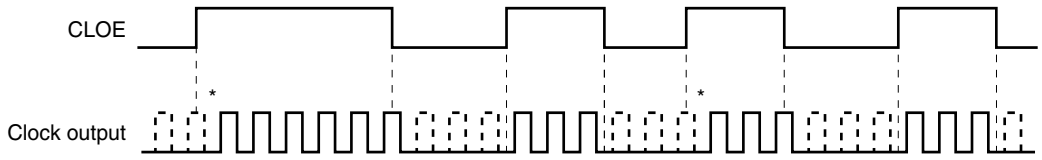
12.4.1 Clock output operation

The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).
<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 12-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after the high-level period of the clock.

Figure 12-4. Remote Control Output Application Example



12.4.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output selection register (CKS) (buzzer output in disabled status).
<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

CHAPTER 13 A/D CONVERTER

13.1 Function of A/D Converter

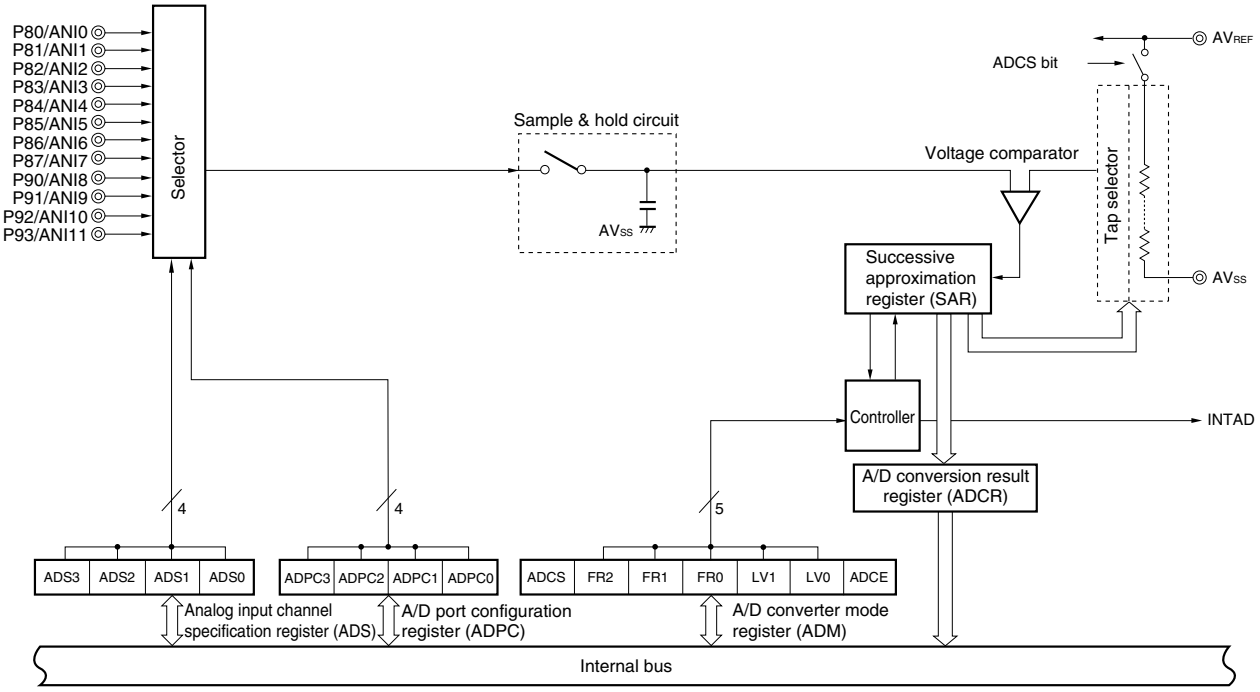
The A/D converter converts an analog input signal into a digital value, and consists of up to twelve channels (ANI0 to ANI11) with a resolution of 10 bits.

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 channel selected from analog inputs ANI0 to ANI11. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 13-1. Block Diagram of A/D Converter



13.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI11 pins

These are the analog input pins of the 12-channel A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

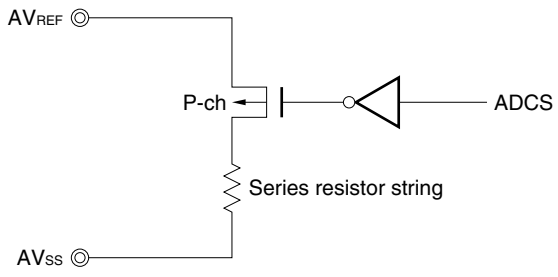
(2) Sample & hold circuit

The sample & hold circuit samples the input voltage of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled voltage value during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared with the sampled voltage value.

Figure 13-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled voltage value and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register converts the result of comparison by the voltage comparator, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

Caution When data is read from ADCR and ADCRH, a wait cycle is generated. Do not read data from ADCR and ADCRH when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

(9) AV_{REF} pin

<R> This pin inputs an analog power/reference voltage to the A/D converter. Make this pin the same potential as the V_{DD} pin when port 8 and port 9 are used as a digital port.

The signal input to ANI0 to ANI11 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(10) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

(11) A/D converter mode register (ADM)

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

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

This register switches the P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 pins to analog input of A/D converter or

<R> digital I/O of port.

(13) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

(14) Port mode register 8 (PM8)

This register switches the P80/ANI0 to P87/ANI7 pins to input or output.

(15) Port mode register 9 (PM9)

This register switches the P90/ANI8 to P93/ANI11 pins to input or output.

13.3 Registers Used in A/D Converter

The A/D converter uses the following seven registers.

- A/D converter mode register (ADM)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- Port mode register 8 (PM8)
- Port mode register 9 (PM9)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

(1) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

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

Reset signal generation clears this register to 00H.

Figure 13-3. Format of A/D Converter Mode Register (ADM)

Address: FF2AH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2Note 1	FR1Note 1	FR0Note 1	LV1Note 1	LV0Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADCE	Comparator operation controlNote 2
0	Stops comparator operation
1	Enables comparator operation (comparator: 1/2AV_{REF} operation)

Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 13-2 A/D Conversion Time Selection**.

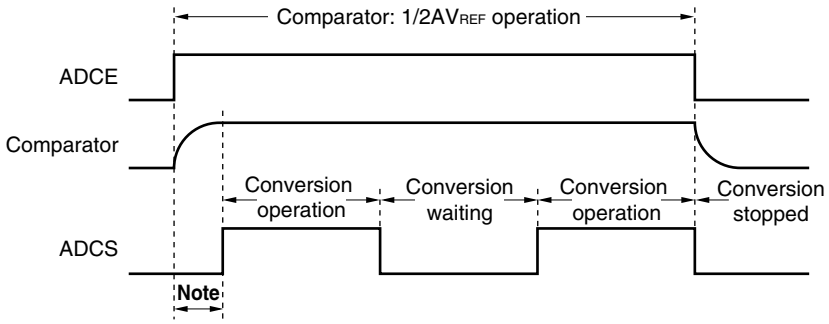
2. The operation of the comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 13-1. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (comparator: 1/2AV$_{REF}$ operation, only comparator consumes power)
1	0	Conversion mode (comparator operation stopped[Note])
1	1	Conversion mode (comparator: 1/2AV$_{REF}$ operation)

Note Ignore data of the first conversion because it is not guaranteed range.

Figure 13-4. Timing Chart When Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 μs or longer.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR2, LV1, and LV0 to values other than the identical data.

2. If data is written to ADM, a wait cycle is generated. Do not write data to ADM when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

Table 13-2. A/D Conversion Time Selection

(1) 2.7 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register (ADM)					Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{PRS} = 4 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	0	0	264/f_{PRS}	Setting prohibited	26.4 μs	13.2 μs	f_{PRS}/12
0	0	1	0	0	176/f_{PRS}		17.6 μs	8.8 μs[Note]	f_{PRS}/8
0	1	0	0	0	132/f_{PRS}	33.0 μs	13.2 μs	6.6 μs[Note]	f_{PRS}/6
0	1	1	0	0	88/f_{PRS}	22.0 μs	8.8 μs[Note]	Setting prohibited	f_{PRS}/4
1	0	0	0	0	66/f_{PRS}	16.5 μs	6.6 μs[Note]		f_{PRS}/3
1	0	1	0	0	44/f_{PRS}	11.0 μs[Note]	Setting prohibited		f_{PRS}/2
Other than above					Setting prohibited				

Note This can be set only when 4.0 V ≤ AV_{REF} ≤ 5.5 V.

(2) 2.3 V ≤ AV_{REF} < 2.7 V

A/D Converter Mode Register (ADM)					Conversion Time Selection			Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	
0	0	0	0	1	480/f_{PRS}	Setting prohibited	Setting prohibited	f_{PRS}/12
0	0	1	0	1	320/f_{PRS}		64.0 μs	f_{PRS}/8
0	1	0	0	1	240/f_{PRS}	60.0 μs	48.0 μs	f_{PRS}/6
0	1	1	0	1	160/f_{PRS}	40.0 μs	32.0 μs	f_{PRS}/4
1	0	0	0	1	120/f_{PRS}	30.0 μs	Setting prohibited	f_{PRS}/3
Other than above					Setting prohibited			

Cautions 1. Set the conversion times with the following conditions.

- **4.0 V ≤ AV_{REF} ≤ 5.5 V: f_{AD} = 0.6 to 3.6 MHz**
- **2.7 V ≤ AV_{REF} < 4.0 V: f_{AD} = 0.6 to 1.8 MHz**
- **2.3 V ≤ AV_{REF} < 2.7 V: f_{AD} = 0.6 to 1.48 MHz**

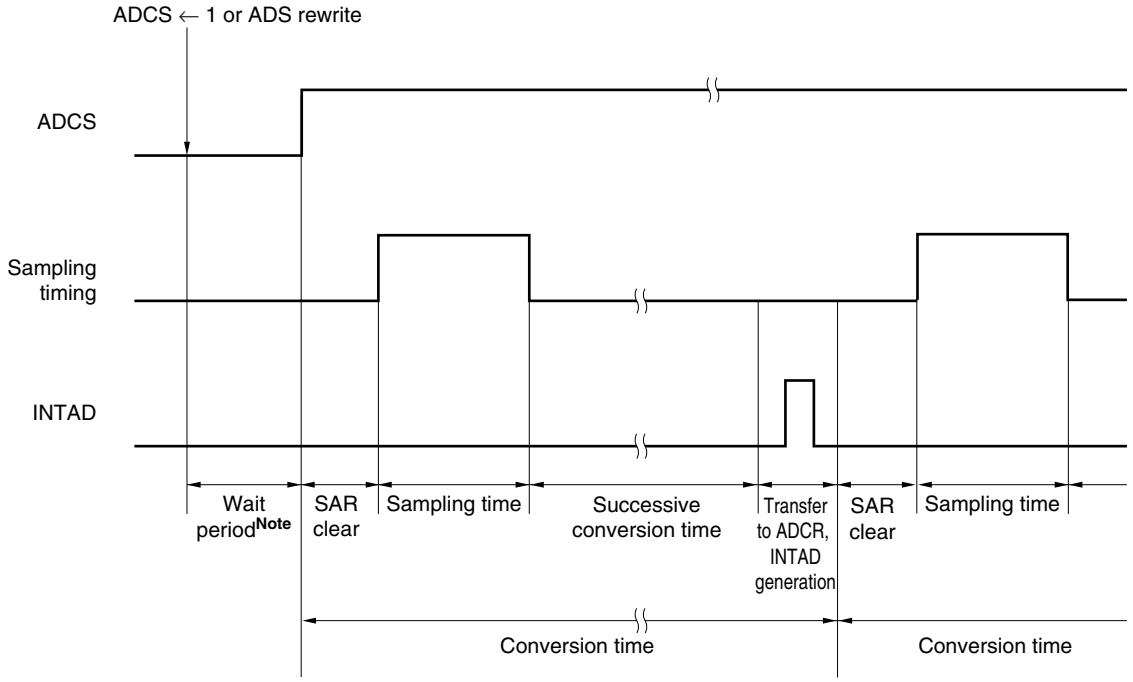
2. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

3. Change LV1 and LV0 from the default value, when 2.3 V ≤ AV_{REF} < 2.7 V.

4. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 13-5. A/D Converter Sampling and A/D Conversion Timing



Note For details of wait period, see **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 lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register, and is stored in ADCR in order starting from bit 7 of FF19H. FF19H indicates the higher 8 bits of the conversion result, and FF18H indicates the lower 2 bits of the conversion result.

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

Reset signal generation clears this register to 0000H.

Figure 13-6. Format of 10-Bit A/D Conversion Result Register (ADCR)

Address: FF18H, FF19H After reset: 0000H R

Symbol	FF19H								FF18H							
ADCR											0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCR, a wait cycle is generated. Do not read data from ADCR when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

(3) 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Figure 13-7. Format of 8-Bit A/D Conversion Result Register (ADCRH)

Address: FF19H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Cautions 1. When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCRH, a wait cycle is generated. Do not read data from ADCRH when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

(4) Analog input channel specification register (ADS)

This register specifies the input port of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 13-8. Format of Analog Input Channel Specification Register (ADS)

Address: FF2BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

ADS3	ADS2	ADS1	ADS0	Analog input channel specification
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
0	1	1	0	ANI6
0	1	1	1	ANI7
1	0	0	0	ANI8
1	0	0	1	ANI9
1	0	1	0	ANI10
1	0	1	1	ANI11

Cautions 1. Be sure to clear bits 4 to 7 to 0.

2 Because ADS and ADPC do not control input and output, set the channel used for A/D conversion in the input mode by using port mode register 8, 9 (PM8, PM9). If the channel is set in the output mode, selection of ADPC is disabled.

3. Do not set a pin to be used as a digital input pin with ADPC with ADS.

4. If data is written to ADS, a wait cycle is generated. Do not write data to ADS when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

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

This register switches the P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation clears this register to 00H.

Figure 13-9. Format of A/D Port Configuration Register (ADPC)

Address: FF22H After reset: 00H R/W

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

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/ digital I/O (D) switching											
				P93/ ANI11	P92/ ANI10	P91/ ANI9	P90/ ANI8	P87/ ANI7	P86/ ANI6	P85/ ANI5	P84/ ANI4	P83/ ANI3	P82/ ANI2	P81/ ANI1	P80/ ANI0
0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
0	1	1	1	A	A	A	A	A	D	D	D	D	D	D	D
1	0	0	0	A	A	A	A	D	D	D	D	D	D	D	D
1	0	0	1	A	A	A	D	D	D	D	D	D	D	D	D
1	0	1	0	A	A	D	D	D	D	D	D	D	D	D	D
1	0	1	1	A	D	D	D	D	D	D	D	D	D	D	D
1	1	0	0	D	D	D	D	D	D	D	D	D	D	D	D
Other than above				Setting prohibited											

Cautions 1. Set the channel to be used for A/D conversion in the input mode by using port mode register 8, 9 (PM8, PM9).

2. If data is written to ADPC, a wait cycle is generated. Do not write data to ADPC when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

(6) Port mode register 8 (PM8)

When using the P80/ANI0 to P87/ANI7 pins for analog input port, set PM80 to PM87 to 1. The output latches of P80 to P87 at this time may be 0 or 1.

If PM80 to PM87 are set to 0, they cannot be used as analog input port pins.

PM8 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 13-10. Format of Port Mode Register 8 (PM8)

Address: FF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80

PM8n	P8n pin I/O mode selection (n = 0 to 7)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

(7) Port mode register 9 (PM9)

When using the P90/ANI8 to P93/ANI11 pins for analog input port, set PM90 to PM93 to 1. The output latches of P90 to P93 at this time may be 0 or 1.

If PM90 to PM93 are set to 0, they cannot be used as analog input port pins.

PM9 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 13-11. Format of Port Mode Register 9 (PM9)

Address: FF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	1	1	1	1	PM93	PM92	PM91	PM90

PM9n	P9n pin I/O mode selection (n = 0 to 3)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 pins are as shown below depending on the settings of ADPC, ADS, PM8 and PM9.

Table 13-3. Setting Functions of P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 Pins

ADPC	PM8, PM9	ADS	P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 Pins
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output

13.4 A/D Converter Operations

13.4.1 Basic operations of A/D converter

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1 to start the operation of the comparator.
<2> Set channels for A/D conversion to analog input by using the A/D port configuration register (ADPC) and set to input mode by using port mode register 8, 9 (PM8, PM9).
<3> Set A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM.
<4> Select one channel for A/D conversion using the analog input channel specification register (ADS).
<5> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.
(<6> to <12> are operations performed by hardware.)
<6> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.
<7> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.
<8> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.
<9> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.
<10> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.
- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

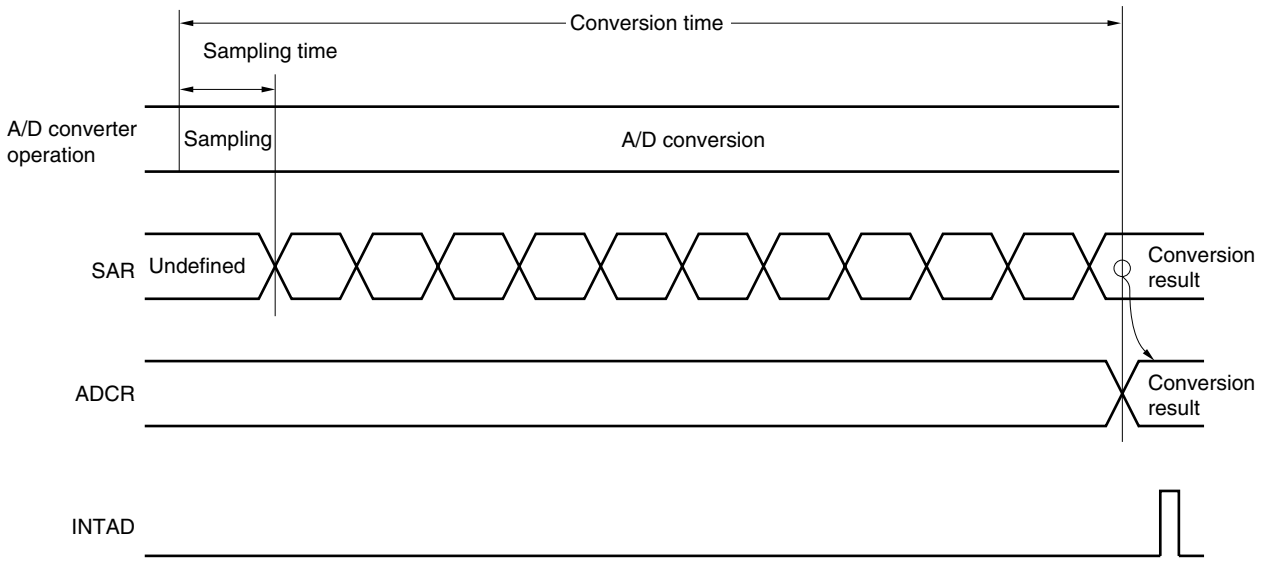
The voltage tap and sampled voltage are compared and bit 8 of SAR is manipulated as follows.
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<11> Comparison is continued in this way up to bit 0 of SAR.
<12> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.
<13> Repeat steps <6> to <12>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <5>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <5>. To change a channel of A/D conversion, start from <4>.

Caution Make sure the period of <1> to <5> is 1 μs or more.

Remark Two types of A/D conversion result registers are available.
- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

Figure 13-12. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

13.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI11) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

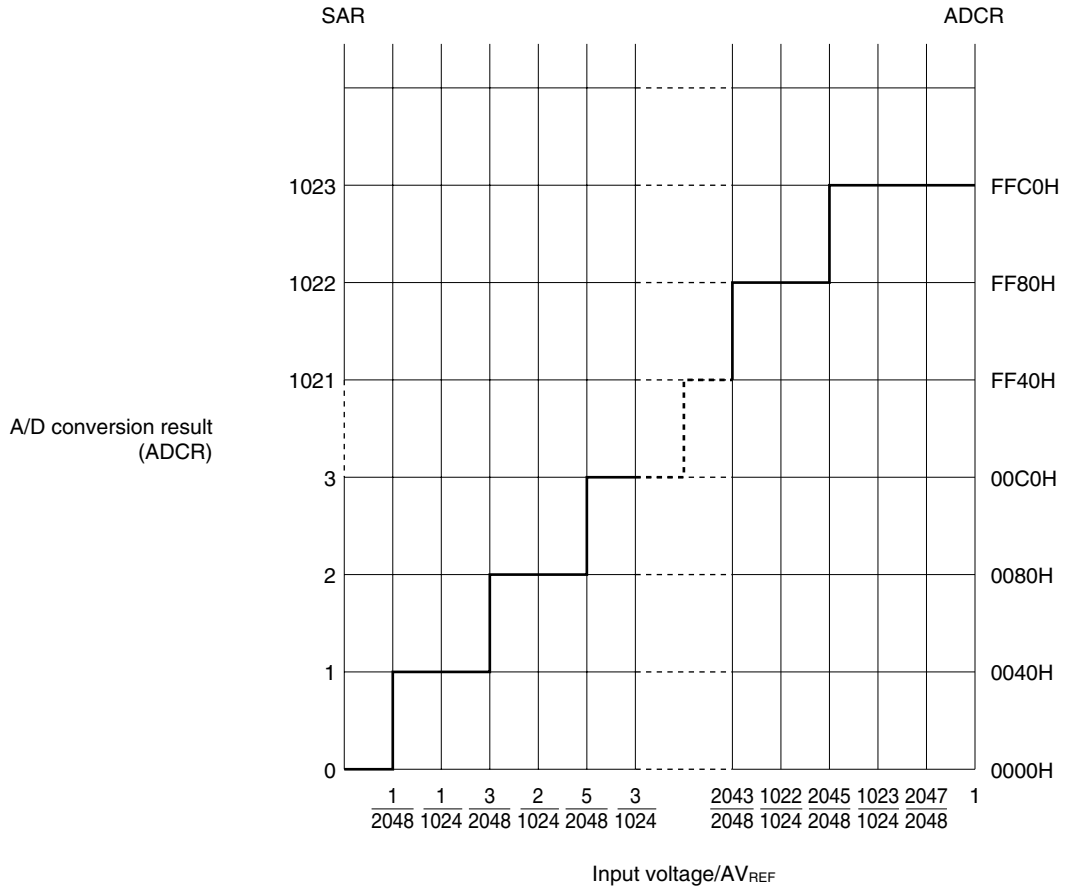
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 13-13 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 13-13. Relationship Between Analog Input Voltage and A/D Conversion Result



13.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI11 by the analog input channel specification register (ADS) and A/D conversion is executed.

(1) A/D conversion operation

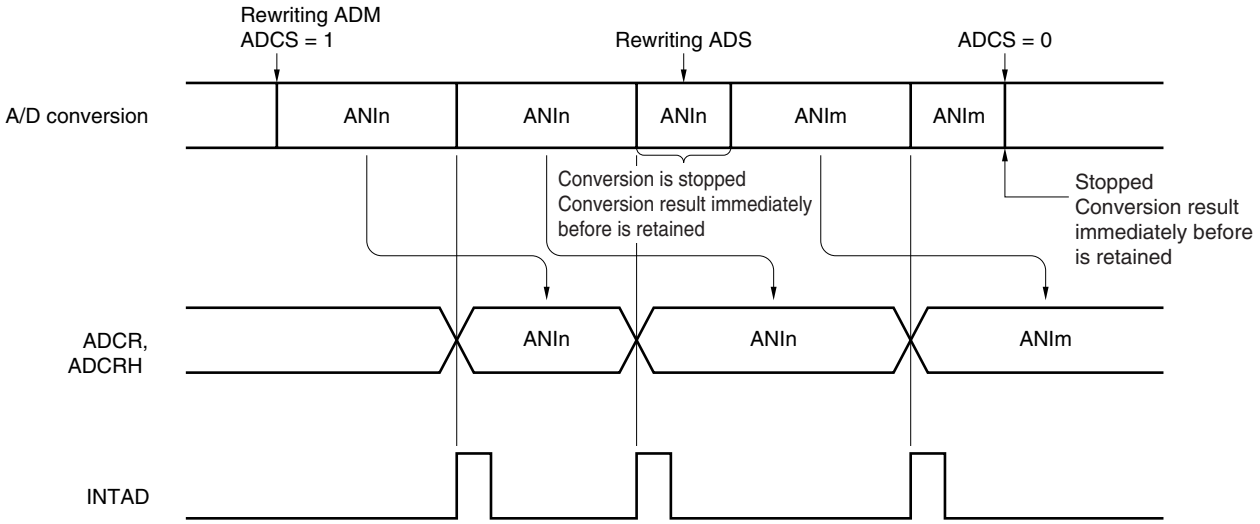
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 13-14. A/D Conversion Operation



Remarks 1. n = 0 to 11

2. m = 0 to 11

The setting methods are described below.

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
<2> Set the channel to be used in the analog input mode by using bits 3 to 0 (ADPC3 to ADPC0) of the A/D port configuration register (ADPC) and bits 7 to 0 (PM87 to PM80) of port mode register 8 (PM8), bits 3 to 0 (PM93 to PM90) of port mode register 9 (PM9).
<3> Select conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM.
<4> Select a channel to be used by using bits 3 to 0 (ADS3 to ADS0) of the analog input channel specification register (ADS).
<5> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.
<6> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<7> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<8> Change the channel using bits 3 to 0 (ADS3 to ADS0) of ADS to start A/D conversion.
<9> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<11> Clear ADCS to 0.
<12> Clear ADCE to 0.

Cautions 1. Make sure the period of <1> to <5> is 1 μs or more.
2. <1> may be done between <2> and <4>.
3. <1> can be omitted. However, ignore data of the first conversion after <5> in this case.
4. The period from <6> to <9> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <8> to <9> is the conversion time set using FR2 to FR0, LV1, and LV0.

13.5 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 13-15. Overall Error

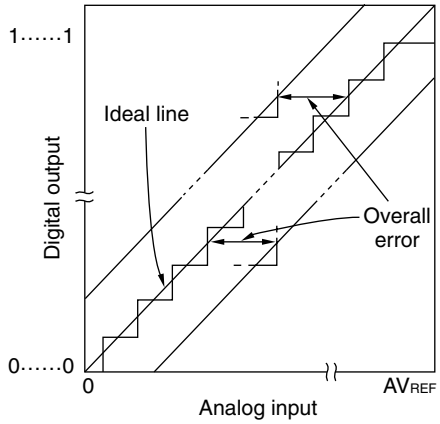
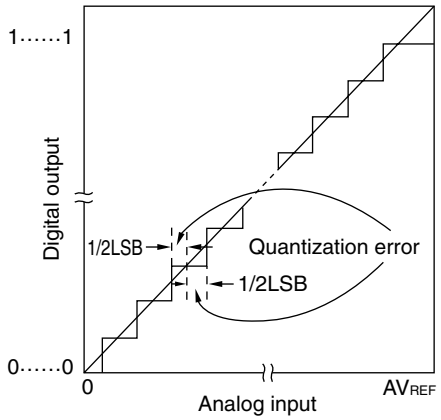


Figure 13-16. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale − 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(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 13-17. Zero-Scale Error

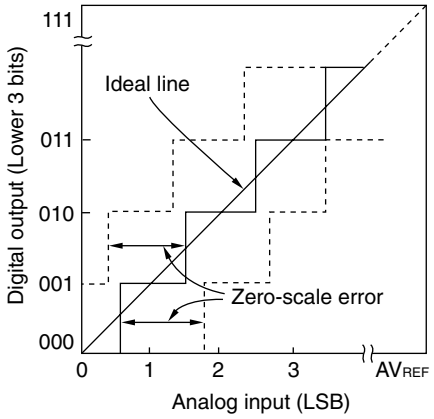


Figure 13-18. Full-Scale Error

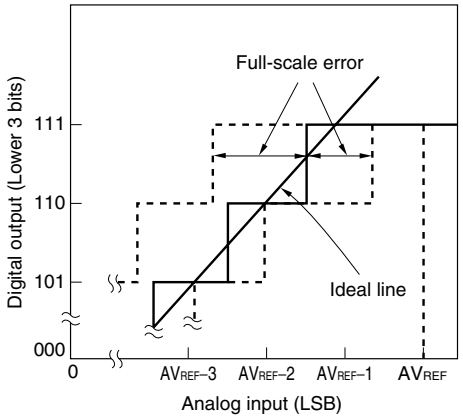


Figure 13-19. Integral Linearity Error

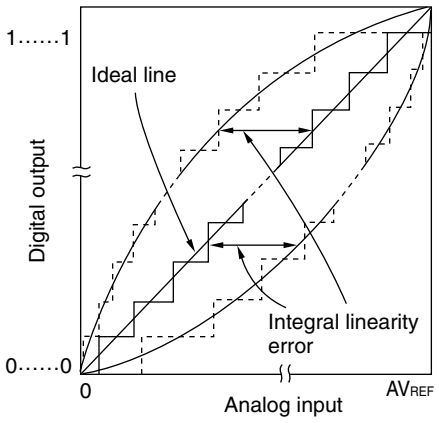
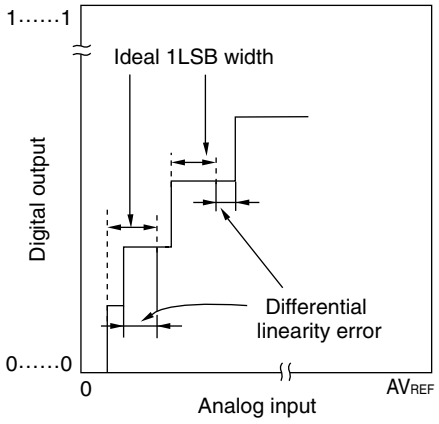


Figure 13-20. Differential Linearity Error



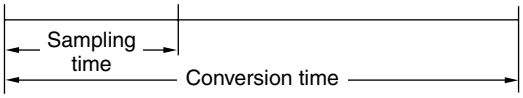
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



13.6 Cautions for A/D Converter

(1) Operating current in STOP mode

The A/D converter stops operating in the STOP mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0.

To restart from the standby status, clear bit 6 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of ANI0 to ANI11

Observe the rated range of the ANI0 to ANI11 input voltage. If a voltage of AV_{REF} or higher and AV_{SS} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion

ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.

<2> Conflict between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

ADM, ADS, or ADPC write has priority. ADCR or ADCRH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REF} pin and pins ANI0 to ANI11.

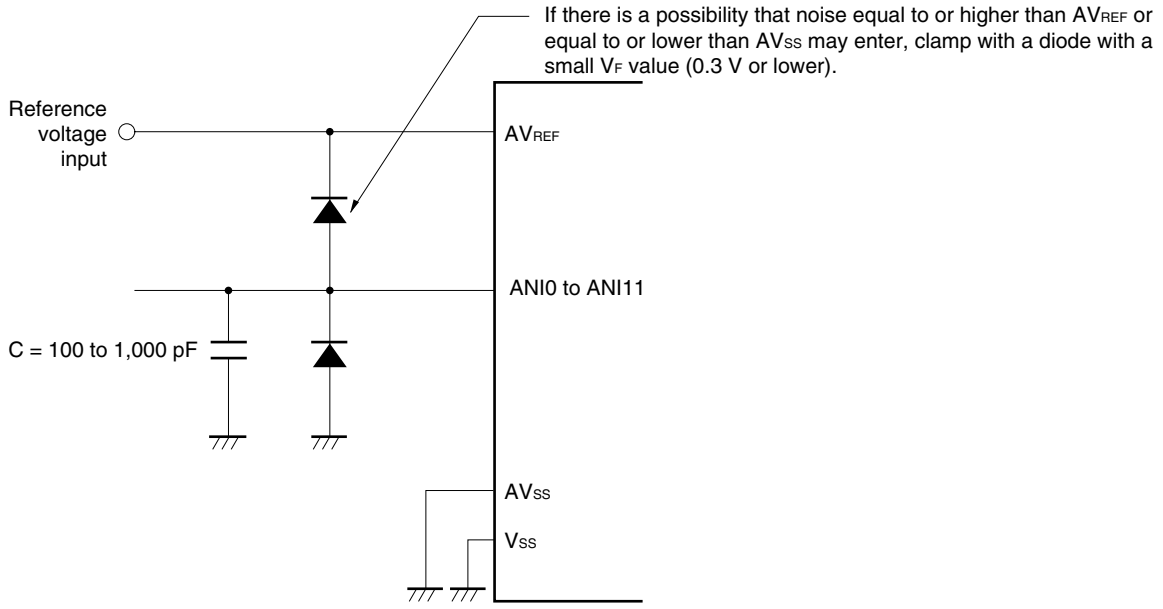
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 13-21 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 13-21. Analog Input Pin Connection



(5) P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11

<1> The analog input pins (ANI0 to ANI11) are also used as I/O port pins (P80 to P87, P90 to P93).
When A/D conversion is performed with any of ANI0 to ANI11 selected, do not access P80 to P87, P90 to P93 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P80 to P87, P90 to P93 starting with the P80/ANI0 that is the furthest from AVREF.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI0 to ANI11 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 10 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI11 pins (see **Figure 13-21**).

(7) AVREF pin input impedance

A series resistor string of several tens of kΩ is connected between the AVREF and AVSS pins.
Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AVREF and AVSS pins, resulting in a large reference voltage error.

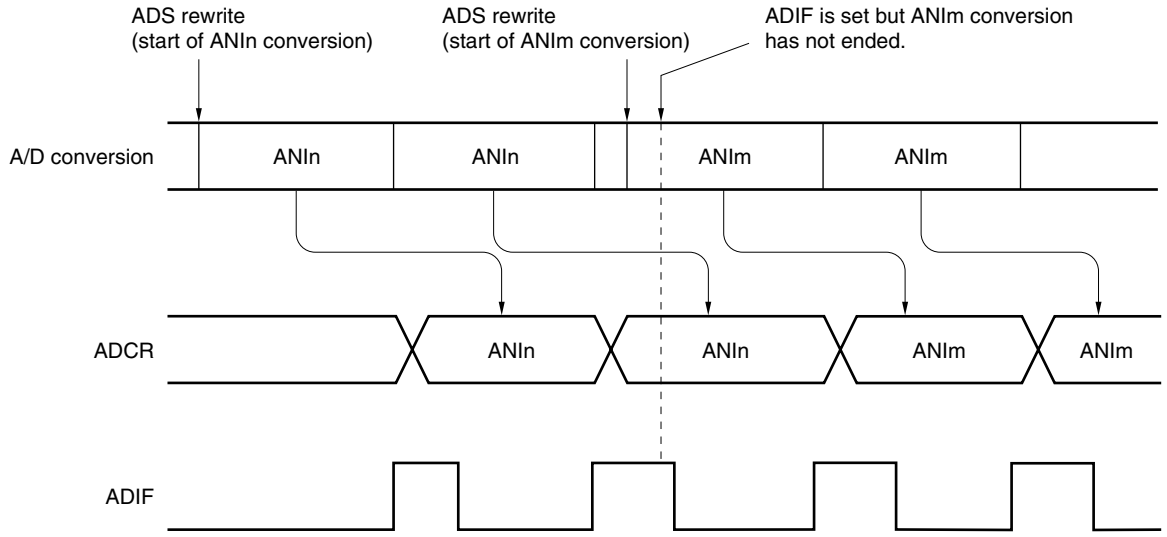
(8) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 13-22. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 11

2. m = 0 to 11

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 13-23. Internal Equivalent Circuit of ANIn Pin

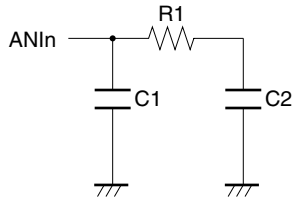


Table 13-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	C1	C2
$4.0\ V \leq AV_{REF} \leq 5.5\ V$	8.1 kΩ	8 pF	5 pF
$2.7\ V \leq AV_{REF} < 4.0\ V$	31 kΩ	8 pF	5 pF
$2.3\ V \leq AV_{REF} < 2.7\ V$	381 kΩ	8 pF	5 pF

Remarks 1. The resistance and capacitance values shown in Table 13-4 are not guaranteed values.

2. n = 0 to 11

CHAPTER 14 SERIAL INTERFACES UART60 AND UART61

The 78K0/FE2 incorporate serial interfaces UART60 and UART61.

14.1 Functions of Serial Interfaces UART60 and UART61

Serial interfaces UART60 and UART61 have the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, see **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode supports the LIN (Local Interconnect Network)-bus. The functions of this mode are outlined below.

For details, see **14.4.2 Asynchronous serial interface (UART) mode** and **14.4.3 Dedicated baud rate generator**.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD6n: Transmit data output pin
  RxD6n: Receive data input pin
- Data length of communication data can be selected from 7 or 8 bits.
- Dedicated internal 8-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently (full-duplex operation).
- Twelve operating clock inputs selectable
- 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 TxD6n output inversion function inverts only the transmission side and not the reception side. To use this function, the reception side must be ready for reception of inverted data.

2. If clock supply to serial interfaces UART60 and UART61 are not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interfaces UART60 and UART61 are stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6n pins 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 by setting POWER6n = 0, RXE6n = 0, and TXE6n = 0.

3. Set POWER6n = 1 and then set TXE6n = 1 (transmission) or RXE6n = 1 (reception) to start communication.

4. TXE6n and RXE6n are synchronized by the base clock (f_{XCLK6}) set by CKSR6n. To enable transmission or reception again, set TXE6n or RXE6n to 1 at least two clocks of the base clock after TXE6n or RXEn6 has been cleared to 0. If TXE6n or RXE6n is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXB6n at least one base clock (f_{XCLK6}) after setting TXE6n = 1.

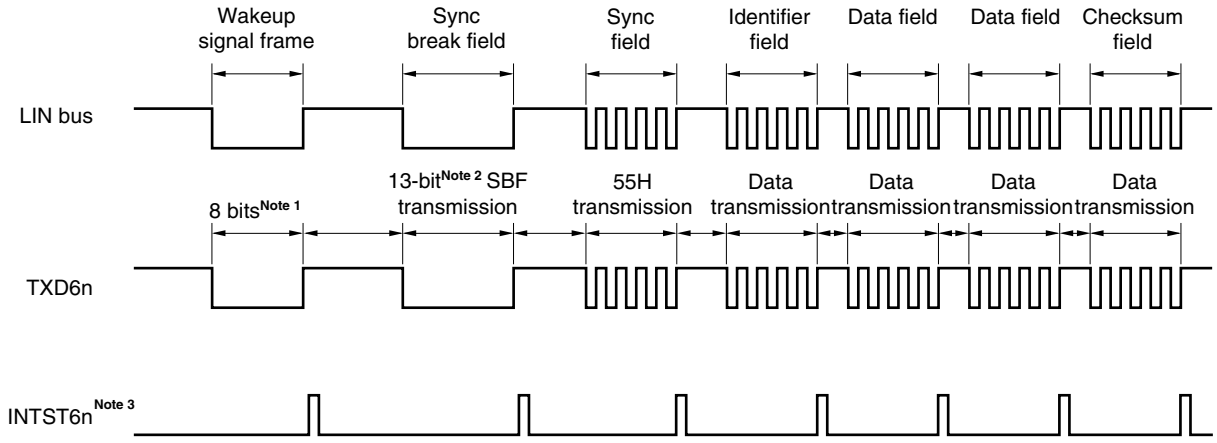
Cautions 6. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is used in LIN communication operation.

Remarks 1. 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.
n 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.

2. n = 0, 1

Figures 14-1 and 14-2 outline the transmission and reception operations of LIN.

Figure 14-1. LIN Transmission Operation



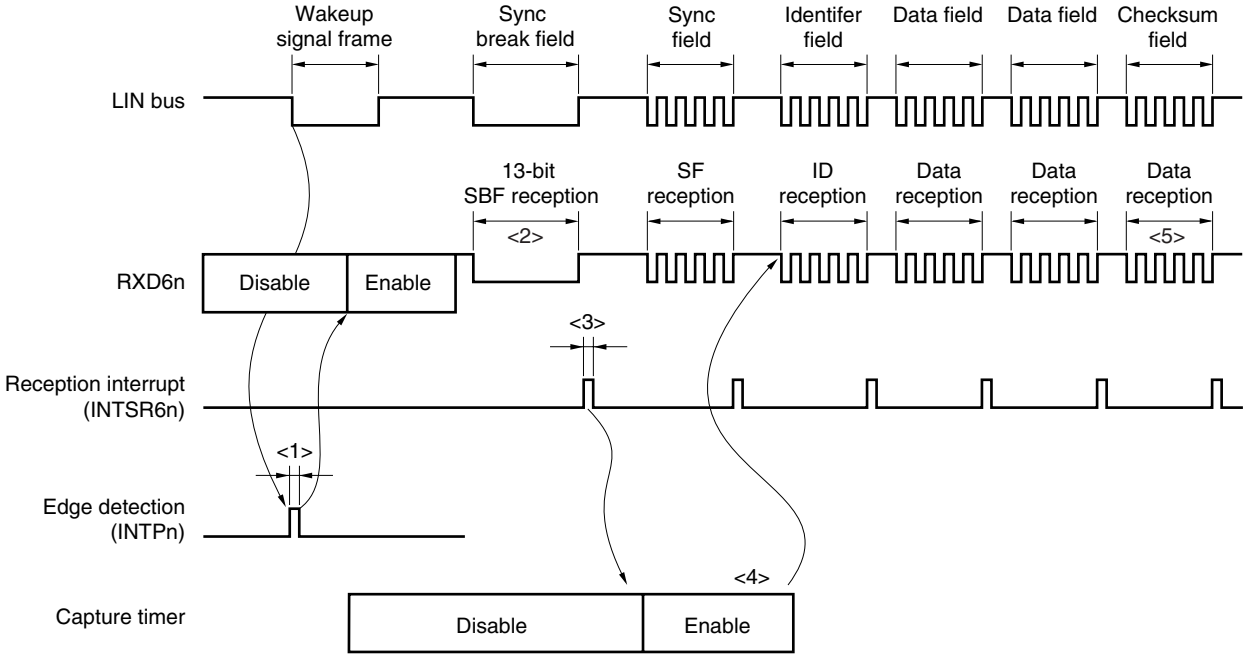
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62n to SBL60n) of asynchronous serial interface control register 6n (ASICL6n). If more precise output width adjustment is necessary, use baud rate generator control register 6n (BRGC6n)
(see **14.4.2 (2) (h) SBF transmission**).

3. INTST6n 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.
n = 0, 1

Figure 14-2. LIN Reception Operation



Reception processing is as follows.

<1> The wakeup signal is detected at the edge of the pin, and enables UART6n and sets the SBF reception mode.

<2> Reception continues until the STOP bit is detected. When an SBF with low-level data of 11 bits or more has been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If an SBF with low-level data of less than 11 bits has been detected, it is assumed that an SBF reception error has occurred. The interrupt signal is not output and the SBF reception mode is restored.

<3> If SBF reception has been completed correctly, an interrupt signal is output. Start 16-bit timer/event counter 00 by the SBF reception end interrupt servicing and measure the bit interval (pulse width) of the sync field (see **7.4.3 Pulse width measurement operation**). Detection of errors OVE6n, PE6n, and FE6n is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6n is not performed. The shift register holds the reset value FFH.

<4> Calculate the baud rate error from the bit length of the sync field, disable UART6n after SF reception, and then re-set baud rate generator control register 6n (BRGC6n).

<5> Distinguish the checksum field by software. Also perform processing by software to initialize UART6n after reception of the checksum field and to set the SBF reception mode again.

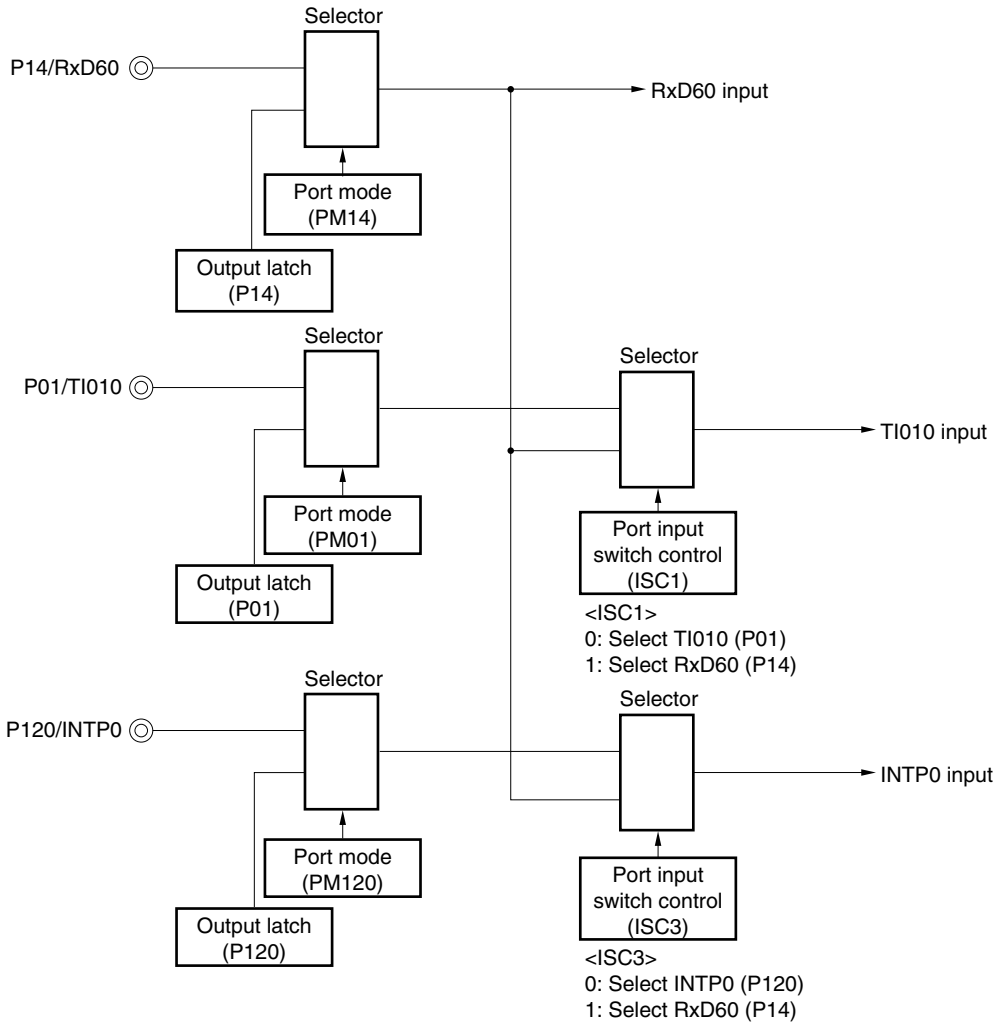
Remark n = 0, 1

Figure 14-3 and 14-4 show 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 and INTP1). 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, 01, and the baud rate error can be calculated.

The input source of the reception port input (RxD60 and RxD61) can be input to the external interrupt (INTP0 and INTP1) and 16-bit timer/event counter 00, 01 by port input switch control (ISC), without connecting RxD60, RxD61, INTP0, INTP1, TI010 and TI001 externally.

Figure 14-3. Port Configuration for LIN Reception Operation (UART60)



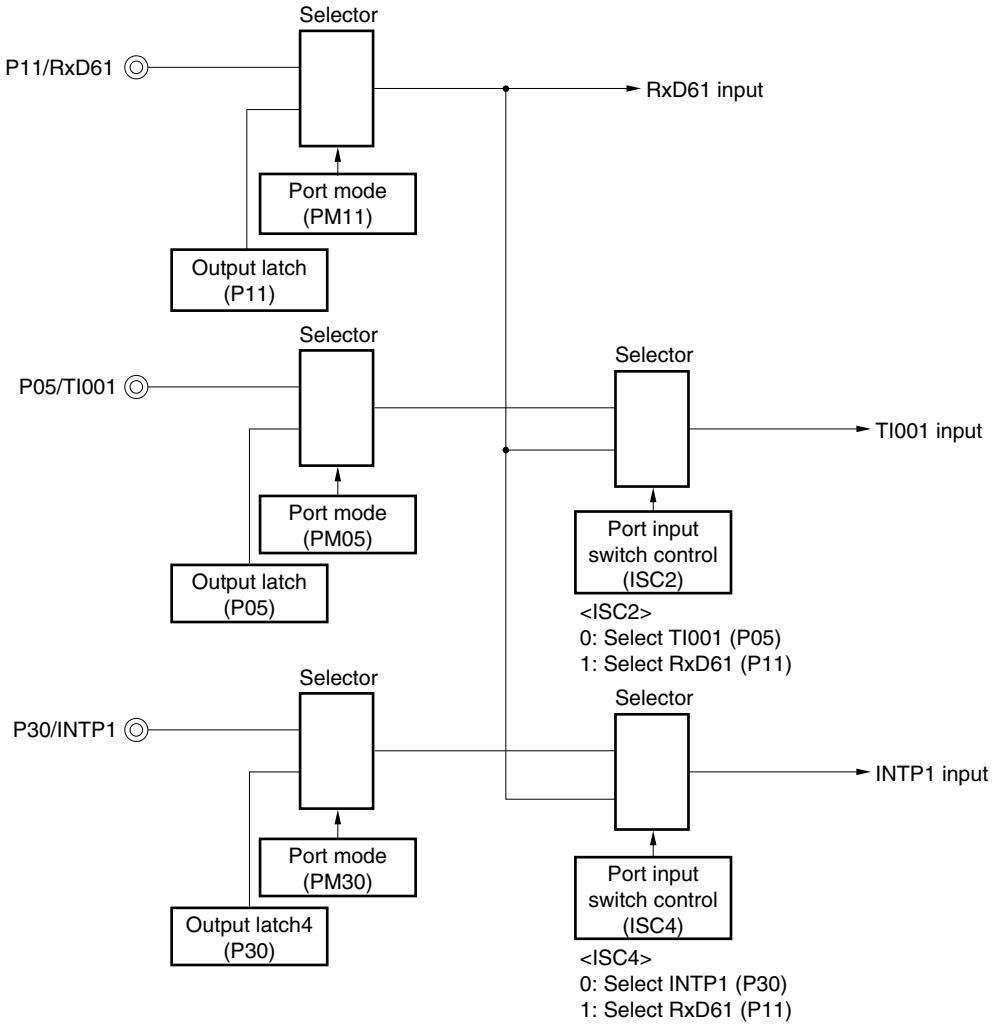
Remark ISC1, ISC3: Bits 1 and 3 of the input switch control register (ISC) (see **Figure 14-19**)

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 (TI010); baud rate error detection
  Use: Detects the baud rate error (measures the TI010 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART60.

Figure 14-4. Port Configuration for LIN Reception Operation (UART61)



Remark ISC2, ISC4: Bits 2 and 4 of the input switch control register (ISC) (see **Figure 14-19**)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTP1); wakeup signal detection
  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter 00 (TI001); baud rate error detection
  Use: Detects the baud rate error (measures the TI001 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART61.

14.2 Configurations of Serial Interface UART60 and UART61

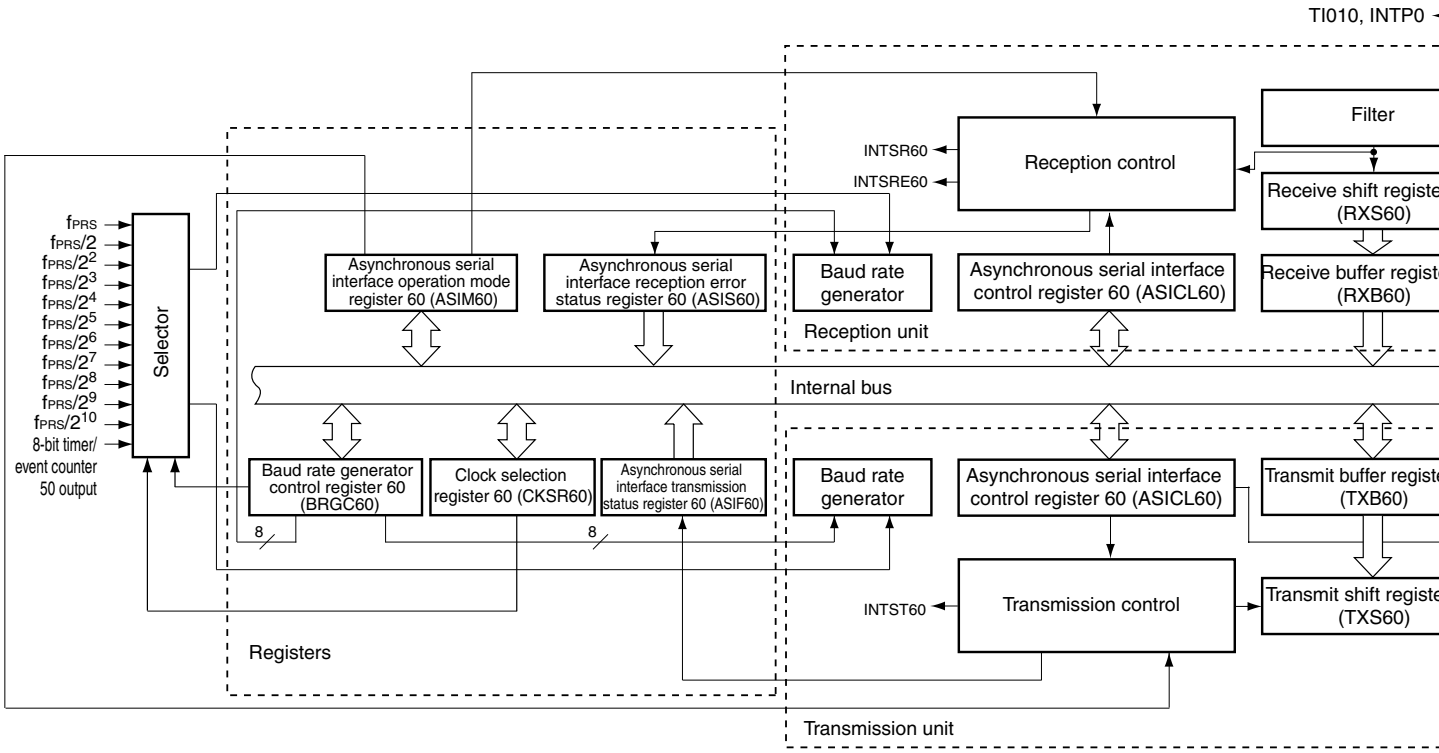
Serial interfaces UART60 and UART61 include the following hardware.

Table 14-1. Configurations of Serial Interface UART60 and UART61

Item	Configuration
Registers	Receive buffer register 6n (RXB6n) Receive shift register 6n (RXS6n) Transmit buffer register 6n (TXB6n) Transmit shift register 6n (TXS6n)
Control registers	Asynchronous serial interface operation mode register 6n (ASIM6n) Asynchronous serial interface reception error status register 6n (ASIS6n) Asynchronous serial interface transmission status register 6n (ASIF6n) Clock selection register 6n (CKSR6n) Baud rate generator control register 6n (BRGC6n) Asynchronous serial interface control register 6n (ASICL6n) Input switch control register (ISC) Port mode register 1 (PM1) Port register 1 (P1)

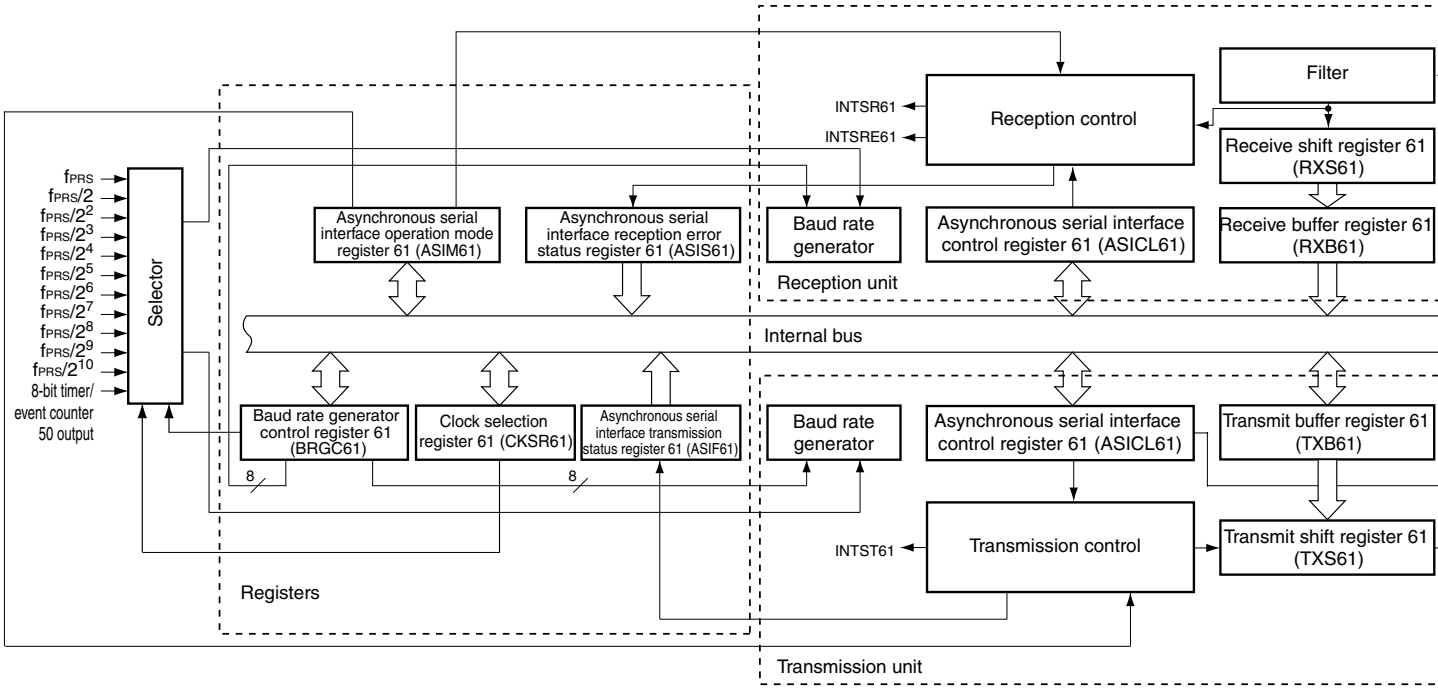
Remark n = 0, 1

Figure 14-5. Block Diagram of Serial Interface UART60



Note Selectable with input switch control register (ISC)

Figure 14-6. Block Diagram of Serial Interface UART61



(1) Receive buffer register 6n (RXB6n)

This 8-bit register stores parallel data converted by receive shift register 6n (RXS6n).

Each time 1 byte of data has been received, new receive data is transferred to this register from RXS6n. If the data length is set to 7 bits, data is transferred as follows.

- In LSB-first reception, the receive data is transferred to bits 0 to 6 of RXB6n and the MSB of RXB6n is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6n and the LSB of RXB6n is always 0.

If an overrun error (OVE6n) occurs, the receive data is not transferred to RXB6n.

RXB6n can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

Reset signal generation sets this register to FFH.

(2) Receive shift register 6n (RXS6n)

This register converts the serial data input to the RxD6n pins into parallel data.

RXS6n cannot be directly manipulated by a program.

(3) Transmit buffer register 6n (TXB6n)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6n.

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

Reset signal generation sets this register to FFH.

Cautions 1. Do not write data to TXB6n when bit 1 (TXBF6n) of asynchronous serial interface transmission status register 6n (ASIF6n) is 1.

2. Do not refresh (write the same value to) TXB6n by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) are 1 or when bits 7 and 5 (POWER6n, RXE6n) of ASIM6n are 1).

3. Set transmit data to TXB6n at least one base clock (f_{XCLK6}) after setting TXE6n = 1.

(4) Transmit shift register 6n (TXS6n)

This register transmits the data transferred from TXB6n from the TxD6n pins as serial data. Data is transferred from TXB6n immediately after TXB6n is written for the first transmission, or immediately before INTST6n occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6n and transmitted from the TxD6n pins at the falling edge of the base clock.

TXS6n cannot be directly manipulated by a program.

Remark n = 0, 1

14.3 Registers Controlling Serial Interfaces UART60 and UART61

Serial interfaces UART60 and UART61 are controlled by the following nine registers.

- Asynchronous serial interface operation mode register 6n (ASIM6n)
- Asynchronous serial interface reception error status register 6n (ASIS6n)
- Asynchronous serial interface transmission status register 6n (ASIF6n)
- Clock selection register 6n (CKSR6n)
- Baud rate generator control register 6n (BRGC6n)
- Asynchronous serial interface control register 6n (ASICL6n)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Asynchronous serial interface operation mode register 6n (ASIM6n)

This 8-bit register controls the serial communication operations of serial interface UART60 and UART61.

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

Reset signal generation sets this register to 01H.

Remarks 1. ASIM6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

2. n = 0, 1

Figure 14-7. Format of Asynchronous Serial Interface Operation Mode Register 60 (ASIM60) (1/2)

Address: FF2EH After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM60	POWER60	TXE60	RXE60	PS610	PS600	CL60	SL60	ISRM60

POWER60	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

TXE60	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE60	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. The output of the TxD60 pins goes high level and the input from the RxD60 pins is fixed to the high level when POWER60 = 0 during transmission.

2. Asynchronous serial interface reception error status register 60 (ASIS60), asynchronous serial interface transmission status register 60 (ASIF60), bit 7 (SBRF60) and bit 6 (SBRT60) of asynchronous serial interface control register 60 (ASICL60), and receive buffer register 60 (RXB60) are reset.

Figure 14-7. Format of Asynchronous Serial Interface Operation Mode Register 60 (ASIM60) (2/2)

PS610	PS600	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.

CL60	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL60	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM60	Enables/disables occurrence of reception completion interrupt in case of error
0	“INTSRE60” occurs in case of error (at this time, INTSR60 does not occur).
1	“INTSR60” occurs in case of error (at this time, INTSRE60 does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE60) of asynchronous serial interface reception error status register 60 (ASIS60) is not set and the error interrupt does not occur.

Cautions 1. To start the transmission, set POWER60 to 1 and then set TXE60 to 1. To stop the transmission, clear TXE60 to 0, and then clear POWER60 to 0.

2. To start the reception, set POWER60 to 1 and then set RXE60 to 1. To stop the reception, clear RXE60 to 0, and then clear POWER60 to 0.

3. Set POWER60 to 1 and then set RXE60 to 1 while a high level is input to the RxD60 pins. If POWER60 is set to 1 and RXE60 is set to 1 while a low level is input, reception is started.

4. TXE60 and RXE60 are synchronized by the base clock (f_{XCLK6}) set by CKSR60. To enable transmission or reception again, set TXE60 or RXE60 to 1 at least two clocks of the base clock after TXE60 or RXE60 has been cleared to 0. If TXE60 or RXE60 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXB60 at least one base clock (f_{XCLK6}) after setting TXE60 = 1.

6. Clear the TXE60 and RXE60 bits to 0 before rewriting the PS610, PS600, and CL60 bits.

7. Fix the PS610 and PS600 bits to 0 when used in LIN communication operation.

8. Clear TXE60 to 0 before rewriting the SL60 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL60 bit.

9. Make sure that RXE60 = 0 when rewriting the ISRM60 bit.

Figure 14-8. Format of Asynchronous Serial Interface Operation Mode Register 61 (ASIM61) (1/2)

Address: FF2FH After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM61	POWER61	TXE61	RXE61	PS611	PS601	CL61	SL61	ISRM61

POWER61	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

TXE61	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE61	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. The output of the TxD61 pins goes high level and the input from the RxD61 pins is fixed to the high level when POWER61 = 0 during transmission.

2. Asynchronous serial interface reception error status register 61 (ASIS61), asynchronous serial interface transmission status register 61 (ASIF61), bit 7 (SBRF61) and bit 6 (SBRT61) of asynchronous serial interface control register 61 (ASICL61), and receive buffer register 61 (RXB61) are reset.

Figure 14-8. Format of Asynchronous Serial Interface Operation Mode Register 61 (ASIM61) (2/2)

PS611	PS601	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.

CL61	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL61	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM61	Enables/disables occurrence of reception completion interrupt in case of error
0	“INTSRE61” occurs in case of error (at this time, INTSR61 does not occur).
1	“INTSR61” occurs in case of error (at this time, INTSRE61 does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE61) of asynchronous serial interface reception error status register 61 (ASIS61) is not set and the error interrupt does not occur.

Cautions 1. To start the transmission, set POWER61 to 1 and then set TXE61 to 1. To stop the transmission, clear TXE61 to 0, and then clear POWER61 to 0.

2. To start the reception, set POWER61 to 1 and then set RXE61 to 1. To stop the reception, clear RXE61 to 0, and then clear POWER61 to 0.

3. Set POWER61 to 1 and then set RXE61 to 1 while a high level is input to the RxD61 pins. If POWER61 is set to 1 and RXE61 is set to 1 while a low level is input, reception is started.

4. TXE61 and RXE61 are synchronized by the base clock (f_{XCLK6}) set by CKSR61. To enable transmission or reception again, set TXE61 or RXE61 to 1 at least two clocks of the base clock after TXE61 or RXE61 has been cleared to 0. If TXE61 or RXE61 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

5. Set transmit data to TXB61 at least one base clock (f_{XCLK6}) after setting TXE61 = 1.

6. Clear the TXE61 and RXE61 bits to 0 before rewriting the PS611, PS601, and CL61 bits.

7. Fix the PS611 and PS601 bits to 0 when used in LIN communication operation.

8. Clear TXE61 to 0 before rewriting the SL61 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL61 bit.

9. Make sure that RXE61 = 0 when rewriting the ISRM61 bit.

(2) Asynchronous serial interface reception error status register 6n (ASIS6n)

This register indicates an error status on completion of reception by serial interfaces UART60 and UART61. It includes three error flag bits (PE6n, FE6n, OVE6n).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6n) or bit 5 (RXE6n) of ASIM6n to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS6n and then read receive buffer register 6n (RXB6n) to clear the error flag.

Figure 14-9. Format of Asynchronous Serial Interface Reception Error Status Register 60 (ASIS60)

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS60	0	0	0	0	0	PE60	FE60	OVE60

PE60	Status flag indicating parity error
0	If POWER60 = 0 and RXE60 = 0, or if ASIS60 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE60	Status flag indicating framing error
0	If POWER60 = 0 and RXE60 = 0, or if ASIS60 register is read
1	If the stop bit is not detected on completion of reception

OVE60	Status flag indicating overrun error
0	If POWER60 = 0 and RXE60 = 0, or if ASIS60 register is read
1	If receive data is set to the RXB60 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE60 bit differs depending on the set values of the PS610 and PS600 bits of asynchronous serial interface operation mode register 60 (ASIM60).

2. The first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 60 (RXB60) but discarded.

4. If data is read from ASIS60, a wait cycle is generated. Do not read data from ASIS60 when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

Figure 14-10. Format of Asynchronous Serial Interface Reception Error Status Register 61 (ASIS61)

Address: FF2FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS61	0	0	0	0	0	PE61	FE61	OVE61

PE61	Status flag indicating parity error
0	If POWER61 = 0 and RXE61 = 0, or if ASIS61 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE61	Status flag indicating framing error
0	If POWER61 = 0 and RXE61 = 0, or if ASIS61 register is read
1	If the stop bit is not detected on completion of reception

OVE61	Status flag indicating overrun error
0	If POWER61 = 0 and RXE61 = 0, or if ASIS61 register is read
1	If receive data is set to the RXB61 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE61 bit differs depending on the set values of the PS611 and PS601 bits of asynchronous serial interface operation mode register 61 (ASIM61).

2. The first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 61 (RXB61) but discarded.

4. If data is read from ASIS61, a wait cycle is generated. Do not read data from ASIS6 when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, see CHAPTER 31 CAUTIONS FOR WAIT.

(3) Asynchronous serial interface transmission status register 6n (ASIF6n)

This register indicates the status of transmission by serial interfaces UART60 and UART61. It includes two status flag bits (TXBF6n and TXSF6n).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6n register after data has been transferred from the TXB6n register to the TXS6n register.

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6n) or bit 6 (TXE6n) of ASIM6n to 0 clears this register to 00H.

Figure 14-11. Format of Asynchronous Serial Interface Transmission Status Register 60 (ASIF60)

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF60	0	0	0	0	0	0	TXBF60	TXSF60

TXBF60	Transmit buffer data flag
0	If POWER60 = 0 or TXE60 = 0, or if data is transferred to transmit shift register 60 (TXS60)
1	If data is written to transmit buffer register 60 (TXB60) (if data exists in TXB60)

TXSF60	Transmit shift register data flag
0	If POWER60 = 0 or TXE60 = 0, or if the next data is not transferred from transmit buffer register 60 (TXB60) after completion of transfer
1	If data is transferred from transmit buffer register 60 (TXB60) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB60 register. Be sure to check that the TXBF60 flag is "0". If so, write the next transmit data (second byte) to the TXB60 register. If data is written to the TXB60 register while the TXBF60 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 TXSF60 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF60 flag is "1", the transmit data cannot be guaranteed.

Figure 14-12. Format of Asynchronous Serial Interface Transmission Status Register 61 (ASIF61)

Address: FF38H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF61	0	0	0	0	0	0	TXBF61	TXSF61

TXBF61	Transmit buffer data flag
0	If POWER61 = 0 or TXE61 = 0, or if data is transferred to transmit shift register 61 (TXS61)
1	If data is written to transmit buffer register 61 (TXB61) (if data exists in TXB61)

TXSF61	Transmit shift register data flag
0	If POWER61 = 0 or TXE61 = 0, or if the next data is not transferred from transmit buffer register 61 (TXB61) after completion of transfer
1	If data is transferred from transmit buffer register 61 (TXB61) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB61 register. Be sure to check that the TXBF61 flag is "0". If so, write the next transmit data (second byte) to the TXB61 register. If data is written to the TXB61 register while the TXBF61 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 TXSF61 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF61 flag is "1", the transmit data cannot be guaranteed.

(4) Clock selection register 6n (CKSR6n)

This register selects the base clocks of serial interface UART60 and UART61.

CKSR6n can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Remark CKSR6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

Figure 14-13. Format of Clock Selection Register 60 (CKSR60)

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR60	0	0	0	0	TPS630	TPS620	TPS610	TPS600

TPS630	TPS620	TPS610	TPS600	Base clock (f_{XCLK6}) selection				
					f_{PRS} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	4 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{PRS}/2$	2 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	1 MHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	500 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	250 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	125 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	62.5 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	31.25 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	15.625 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	7.813 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	3.906 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output[Note]				
Other than above				Setting prohibited				

Note Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable the TO50 pin as a timer output pin in any mode.

Caution Make sure POWER60 = 0 when rewriting TPS630 to TPS600.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

Figure 14-14. Format of Clock Selection Register 61 (CKSR61)

Address: FF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR61	0	0	0	0	TPS631	TPS621	TPS611	TPS601

TPS631	TPS621	TPS611	TPS601	Base clock (f_{XCLK6}) selection				
					f_{PRS} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	4 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{PRS}/2$	2 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	1 MHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	500 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	250 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	125 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	62.5 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	31.25 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	15.625 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	7.813 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	3.906 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output[Note]				
Other than above				Setting prohibited				

Note Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable the TO50 pin as a timer output pin in any mode.

Caution Make sure POWER61 = 0 when rewriting TPS631 to TPS601.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(5) Baud rate generator control register 6n (BRGC6n)

This register sets the division value of the 8-bit counters of serial interface UART60 and UART61.

BRGC6n can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Remark BRGC6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

Figure 14-15. Format of Baud Rate Generator Control Register 60 (BRGC60)

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC60	MDL670	MDL660	MDL650	MDL640	MDL630	MDL620	MDL610	MDL600

MDL670	MDL660	MDL650	MDL640	MDL630	MDL620	MDL610	MDL600	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 (TXE60) and bit 5 (RXE60) of the ASIM6n register = 0 when rewriting the MDL670 to MDL600 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 TPS630 to TPS600 bits of CKSR60 register

2. k: Value set by MDL670 to MDL600 bits (k = 4, 5, 6, ..., 255)

3. ×: Don't care

Figure 14-16. Format of Baud Rate Generator Control Register 61 (BRGC61)

Address: FF3EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC61	MDL671	MDL661	MDL651	MDL641	MDL631	MDL621	MDL611	MDL601

MDL671	MDL661	MDL651	MDL641	MDL631	MDL621	MDL611	MDL601	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 (TXE61) and bit 5 (RXE61) of the ASIM61 register = 0 when rewriting the MDL671 to MDL601 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 TPS631 to TPS601 bits of CKSR61 register

2. k: Value set by MDL671 to MDL601 bits (k = 4, 5, 6, ..., 255)

3. ×: Don't care

(6) Asynchronous serial interface control register 6n (ASICL6n)

This register controls the serial communication operations of serial interface UART60 and UART61.

ASICL6n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 16H.

Caution ASICL6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1). However, do not set both SBRT6n and SBTT6n to 1 by a refresh operation during SBF reception (SBRT6n = 1) or SBF transmission (until INTST6n occurs since SBTT6n has been set (1)), because it may re-trigger SBF reception or SBF transmission.

Figure 14-17. Format of Asynchronous Serial Interface Control Register 60 (ASICL60) (1/2)

Address: FF58H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL60	SBRF60	SBRT60	SBTT60	SBL620	SBL610	SBL600	DIR60	TXDLV60

SBRF60	SBF reception status flag
0	If POWER60 = 0 and RXE60 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT60	SBF reception trigger
0	–
1	SBF reception trigger

SBTT60	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 14-17. Format of Asynchronous Serial Interface Control Register 60 (ASICL60) (2/2)

SBL620	SBL610	SBL600	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.

DIR60	First-bit specification
0	MSB
1	LSB

TXDLV60	Enables/disables inverting TxD6n output
0	Normal output of TxD60
1	Inverted output of TxD60

Cautions 1. In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF60 flag is held (1).

2. Before setting the SBRT60 bit, make sure that bit 7 (POWER60) and bit 5 (RXE60) of ASIM60 = 1. After setting the SBRT60 bit to 1, do not clear it to 0 before SBF reception is completed (before an interrupt request signal is generated).

3. The read value of the SBRT60 bit is always 0. SBRT60 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT60 bit to 1, make sure that bit 7 (POWER60) and bit 6 (TXE60) of ASIM60 = 1. After setting the SBTT60 bit to 1, do not clear it to 0 before SBF transmission is completed (before an interrupt request signal is generated).

5. The read value of the SBTT60 bit is always 0. SBTT60 is automatically cleared to 0 at the end of SBF transmission.

6. Do not set the SBRT60 bit to 1 during reception, and do not set the SBTT60 bit to 1 during transmission.

7. Before rewriting the DIR60 and TXDLV60 bits, clear the TXE60 and RXE60 bits to 0.

Figure 14-18. Format of Asynchronous Serial Interface Control Register 61 (ASICL61) (1/2)

Address: FF3FH After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL61	SBRF61	SBRT61	SBTT61	SBL621	SBL611	SBL601	DIR61	TXDLV61

SBRF61	SBF reception status flag
0	If POWER61 = 0 and RXE61 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT61	SBF reception trigger
0	–
1	SBF reception trigger

SBTT61	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 14-18. Format of Asynchronous Serial Interface Control Register 61 (ASICL61) (2/2)

SBL621	SBL611	SBL601	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.

DIR61	First-bit specification
0	MSB
1	LSB

TXDLV61	Enables/disables inverting TxD6n output
0	Normal output of TxD6n
1	Inverted output of TxD6n

Cautions 1. In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF61 flag is held (1).

2. Before setting the SBRT61 bit, make sure that bit 7 (POWER61) and bit 5 (RXE61) of ASIM61 = 1. After setting the SBRT61 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 SBRT61 bit is always 0. SBRT61 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT61 bit to 1, make sure that bit 7 (POWER61) and bit 6 (TXE61) of ASIM61 = 1. After setting the SBTT61 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 SBTT61 bit is always 0. SBTT61 is automatically cleared to 0 at the end of SBF transmission.

6. Do not set the SBRT61 bit to 1 during reception, and do not set the SBTT61 bit to 1 during transmission.

7. Before rewriting the DIR61 and TXDLV61 bits, clear the TXE61 and RXE61 bits to 0.

(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 input source is switched by setting ISC.

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

Reset signal generation sets this register to 00H.

Figure 14-19. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	ISC7	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC7	Interrupt source selection
0	INTWTI
1	INTDMU

ISC4	INTP1 input source selection
0	INTP1 (P30)
1	RxD61(P11)

ISC3	INTP0 input source selection
0	INTP0 (P120)
1	RxD60 (P14)

ISC2	TI001 input source selection
0	TI001 (P06)
1	RxD61(P11)

ISC1	TI010 input source selection
0	TI010 (P01)
1	RxD60 (P14)

ISC0	TI000 input source selection
0	TI000 (P00)
1	TSOUT

(8) Port mode register 1 (PM1)

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

When using the P13/TxD60 and P10/SCK10/TxD61 pins for serial interface data output, clear PM13 and PM10 to 0 and set the output latch of P13 and P10 to 1.

When using the P14/RxD60 and P11/SI10/RxD61 in for serial interface data input, set PM14 and PM11 to 1. The output latch of P14 and P11 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 14-20. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 Operations of Serial Interface UART60 and UART61

Serial interfaces UART60 and UART61 have the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

14.4.1 Operation stop mode

In this mode, serial communication cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER6n, TXE6n, and RXE6n) of ASIM6n to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6n (ASIM6n).

ASIM6n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6n	POWER6n	TXE6n	RXE6n	PS61n	PS60n	CL6n	SL6n	ISRM6n

POWER6n	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE6n	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).

RXE6n	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The output of the TxD6n pins goes high and the input from the RxD6n pins is fixed to high level when POWER6n = 0.

2. Asynchronous serial interface reception error status register 6n (ASIS6n), asynchronous serial interface transmission status register 6n (ASIF6n), bit 7 (SBRF6n) and bit 6 (SBRT6n) of asynchronous serial interface control register 6n (ASICL6n), and receive buffer register 6n (RXB6n) are reset.

Caution Clear POWER6n to 0 after clearing TXE6n and RXE6n to 0 to stop the operation. To start the communication, set POWER6n to 1, and then set TXE6n and RXE6n to 1.

Remarks 1. To use the RxD60/P14, RxD61/P11/SI10, TxD60/P13 and TxD61/P10/SCK10 pins as general-purpose port pins, see **CHAPTER 5 PORT FUNCTIONS**.

2. n = 0, 1

14.4.2 Asynchronous serial interface (UART) mode

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 6n (ASIM6n)
- Asynchronous serial interface reception error status register 6n (ASIS6n)
- Asynchronous serial interface transmission status register 6n (ASIF6n)
- Clock selection register 6n (CKSR6n)
- Baud rate generator control register 6n (BRGC6n)
- Asynchronous serial interface control register 6n (ASICL6n)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6n register (see **Figure 14-13, 14-14**).
<2> Set the BRGC6n register (see **Figure 14-15, 14-16**).
<3> Set bits 0 to 4 (ISRM6n, SL6n, CL6n, PS60n, PS61n) of the ASIM6n register (see **Figure 14-7, 14-8**).
<4> Set bits 0 and 1 (TXDLV6n, DIR6n) of the ASICL6n register (see **Figure 14-17, 14-18**).
<5> Set bit 7 (POWER6n) of the ASIM6n register to 1.
<6> Set bit 6 (TXE6n) of the ASIM6n register to 1. → Transmission is enabled.
Set bit 5 (RXE6n) of the ASIM6n register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6n (TXB6n). → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

Remark n = 0, 1

The relationship between the register settings and pins is shown below.

Table 14-2. Relationship Between Register Settings and Pins

(a) UART60

POWER6n	TXE6n	RXE6n	PM13	P13	PM14	P14	UART60 Operation	Pin Function	
								TxD60/P13	RxD60/P14
0	0	0	×Note	×Note	×Note	×Note	Stop	P13	P14
1	0	1	×Note	×Note	1	×	Reception	P13	RxD60
	1	0	0	1	×Note	×Note	Transmission	TxD60	P14
	1	1	0	1	1	×	Transmission/ reception	TxD60	RxD60

(b) UART61

POWER6n	TXE6n	RXE6n	PM10	P10	PM11	P11	UART61 Operation	Pin Function	
								TxD61/P10/SCK61	RxD61/P11/SI10
0	0	0	×Note	×Note	×Note	×Note	Stop	P10	P11
1	0	1	×Note	×Note	1	×	Reception	P10	RxD61
	1	0	0	1	×Note	×Note	Transmission	TxD61	P11
	1	1	0	1	1	×	Transmission/ reception	TxD61	RxD61

Note Can be set as port function.

Remarks 1. ×: don't care
POWER6n: Bit 7 of asynchronous serial interface operation mode register 6n (ASIM6n)
TXE6n: Bit 6 of ASIM6n
RXE6n: Bit 5 of ASIM6n
PM1×: Port mode register
P1×: Port output latch

2. n = 0, 1

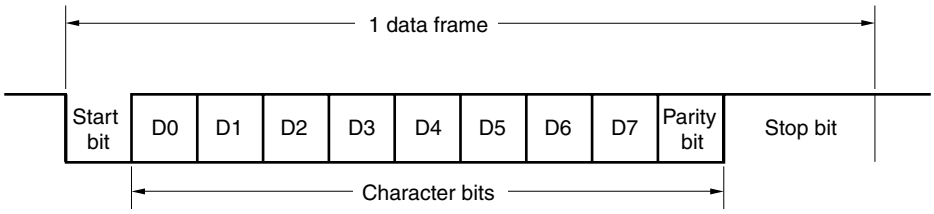
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

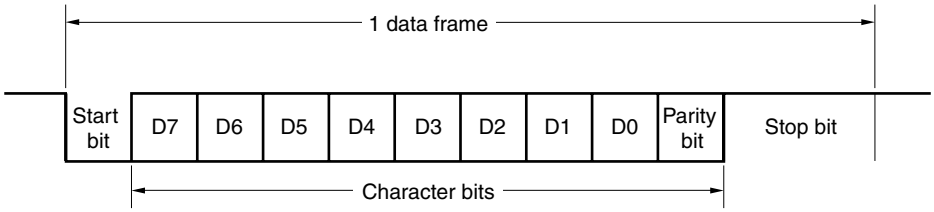
Figures 14-20 and 14-21 show the format and waveform example of the normal transmit/receive data.

Figure 14-21. Format of Normal UART Transmit/Receive Data

1. LSB-first transmission/reception



2. MSB-first transmission/reception



One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6n (ASIM6n).

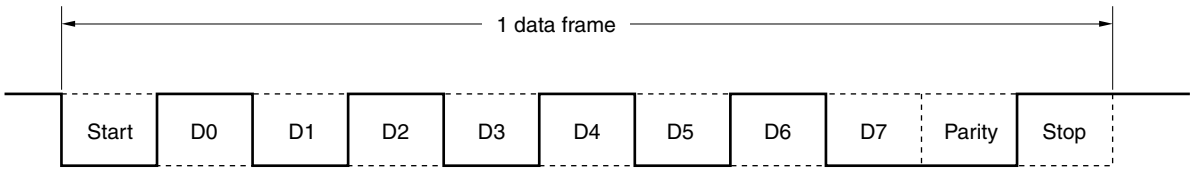
Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6n (ASICL6n).

Whether the TxD6n pins outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6n.

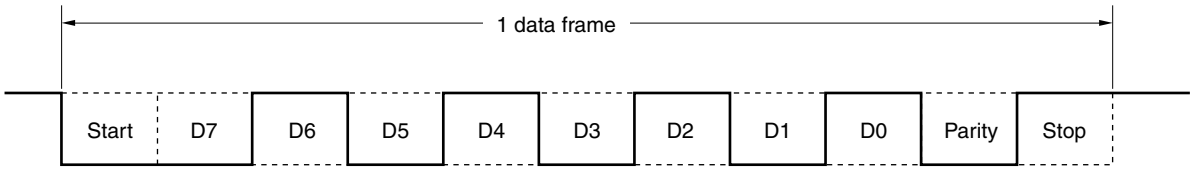
Remark n = 0, 1

Figure 14-22. Example of Normal UART Transmit/Receive Data Waveform

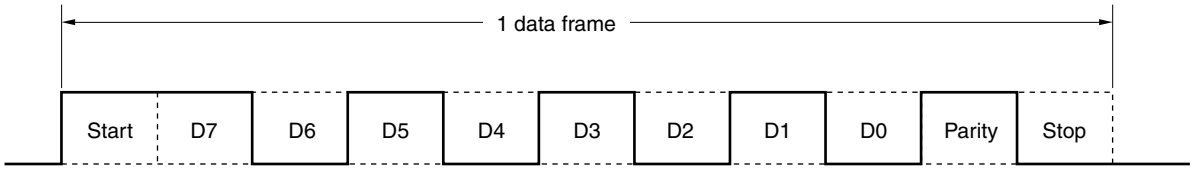
1. Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



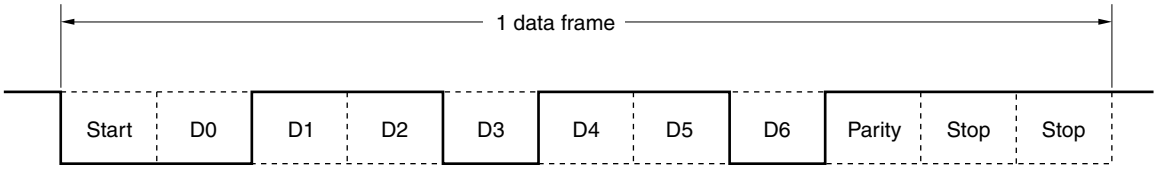
2. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



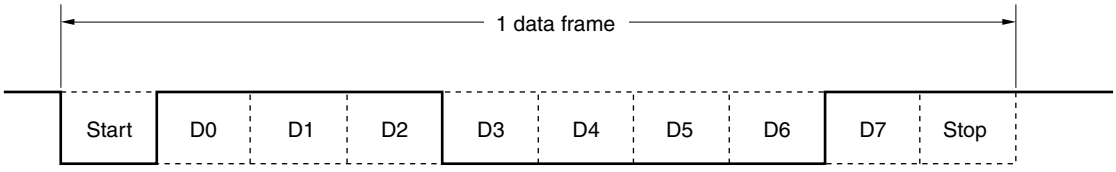
3. Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6n pin inverted output



4. Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H



5. Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Communication data: 87H



Remark n = 0, 1

(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 PS61n and PS60n 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.

Remark n = 0, 1

(c) Normal transmission

When bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and bit 6 (TXE6n) of ASIM6n is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6n (TXB6n). The start bit, parity bit, and stop bit are automatically appended to the data.

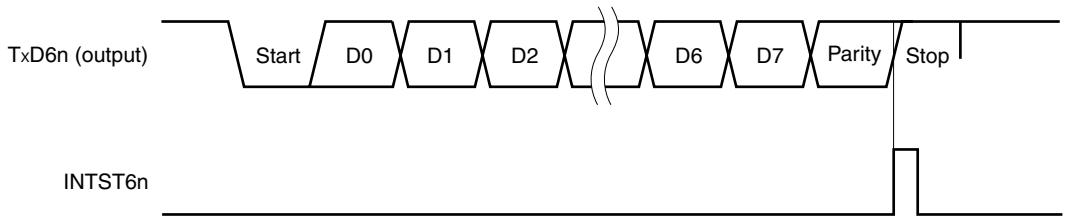
When transmission is started, the data in TXB6n is transferred to transmit shift register 6n (TXS6n). After that, the data is sequentially output from TXS6n to the TxD6n pins. When transmission is completed, the parity and stop bits set by ASIM6n are appended and a transmission completion interrupt request (INTST6n) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6n.

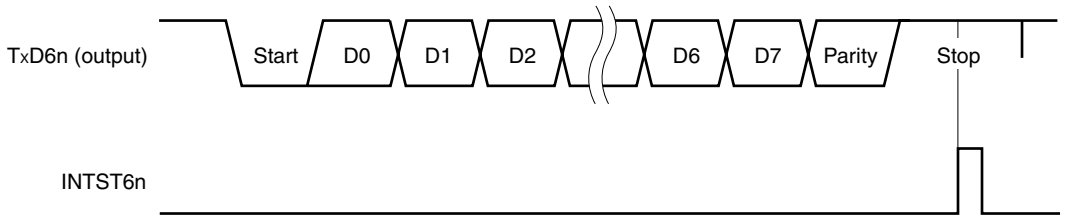
Figure 14-23 shows the timing of the transmission completion interrupt request (INTST6n). This interrupt occurs as soon as the last stop bit has been output.

Figure 14-23. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



Remark n = 0, 1

(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6n (TXB6n) as soon as transmit shift register 6 (TXS6n) has started its shift operation. Consequently, even while the INTST6n interrupt is being serviced after transmission of one data frame, data can be continuously transmitted and an efficient communication rate can be realized. In addition, the TXB6n register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6n) of asynchronous serial interface transmission status register 6n (ASIF6n) when the transmission completion interrupt has occurred.
To transmit data continuously, be sure to reference the ASIF6n register to check the transmission status and whether the TXB6n register can be written, and then write the data.

Cautions 1. The TXBF6n and TXSF6n flags of the ASIF6n register change from "10" to "11", and to "01" during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6n and TXSF6n flags for judgment. Read only the TXBF6n flag when executing continuous transmission.

2. When the device is used in LIN communication operation, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6n (ASIF6n) is 00H before writing transmit data to transmit buffer register 6n (TXB6n).

TXBF6n	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6n register. Be sure to check that the TXBF6n flag is "0". If so, write the next transmit data (second byte) to the TXB6n register. If data is written to the TXB6n register while the TXBF6n flag is "1", the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6n flag.

TXSF6n	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

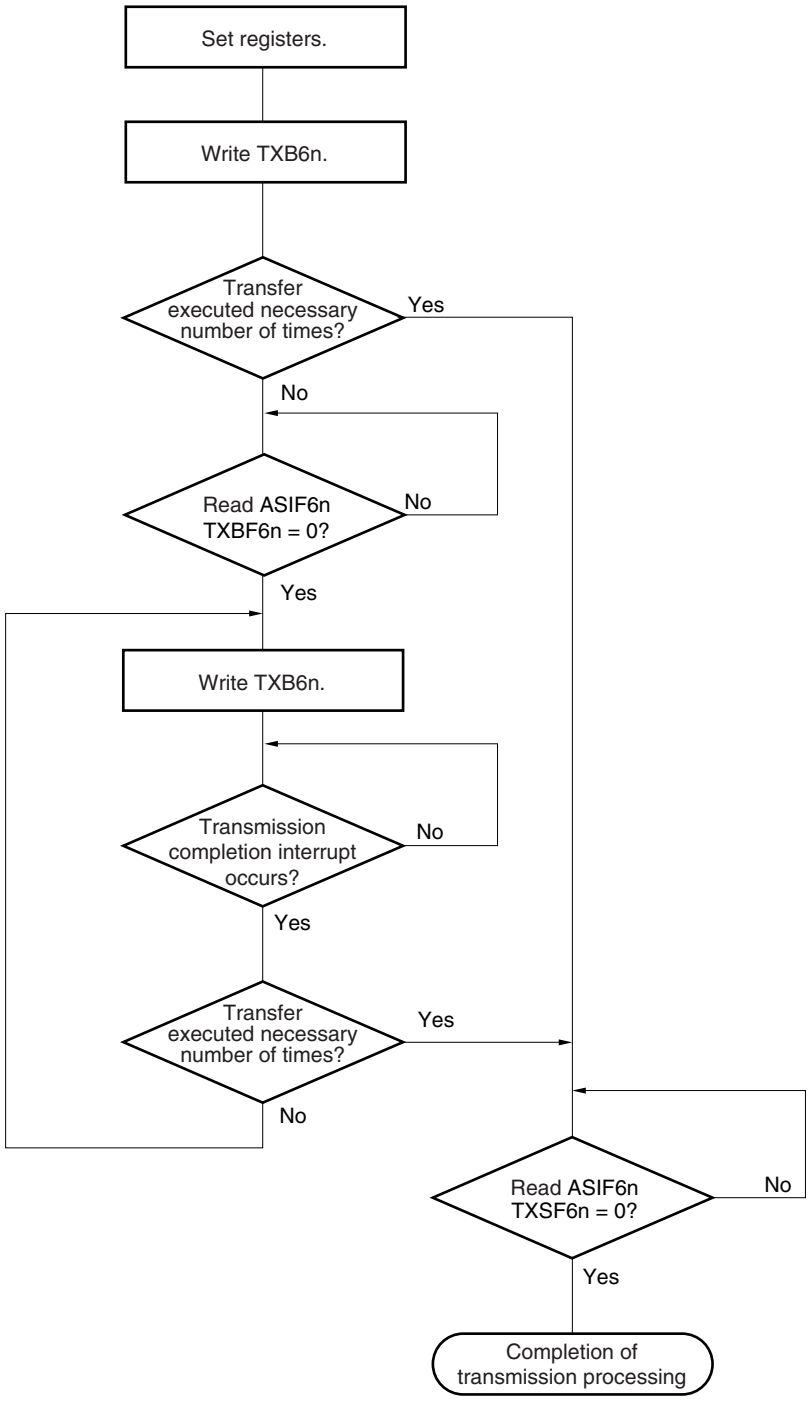
Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6n flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6n flag is "1", the transmit data cannot be guaranteed.

2. During continuous transmission, an overrun error may occur, which means that the next transmission was completed before execution of INTST6n interrupt servicing after transmission of one data frame. An overrun error can be detected by developing a program that can count the number of transmit data and by referencing the TXSF6n flag.

Remark n = 0, 1

Figure 14-24 shows an example of the continuous transmission processing flow.

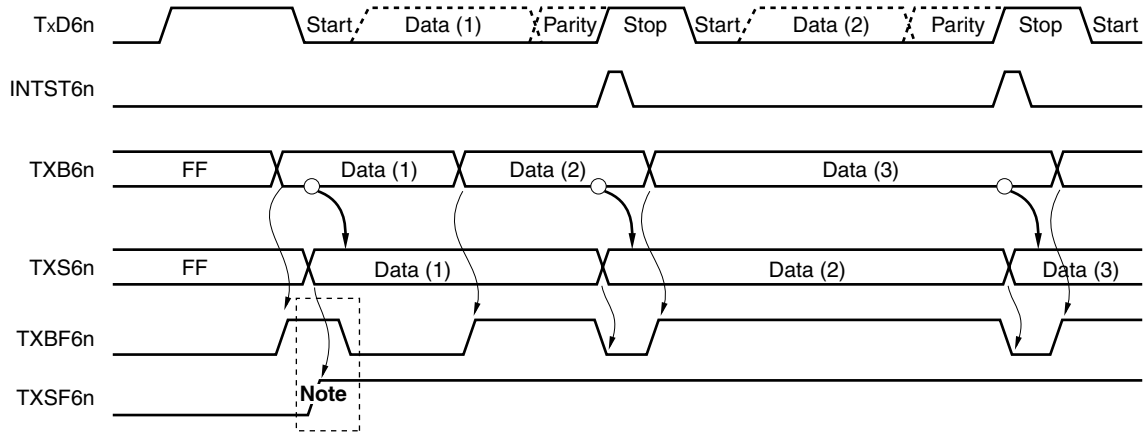
Figure 14-24. Example of Continuous Transmission Processing Flow



Remark TXB6n: Transmit buffer register 6n
ASIF6n: Asynchronous serial interface transmission status register 6n
TXBF6n: Bit 1 of ASIF6n (transmit buffer data flag)
TXSF6n: Bit 0 of ASIF6n (transmit shift register data flag)
n = 0, 1

Figure 14-25 shows the timing of starting continuous transmission, and Figure 14-26 shows the timing of ending continuous transmission.

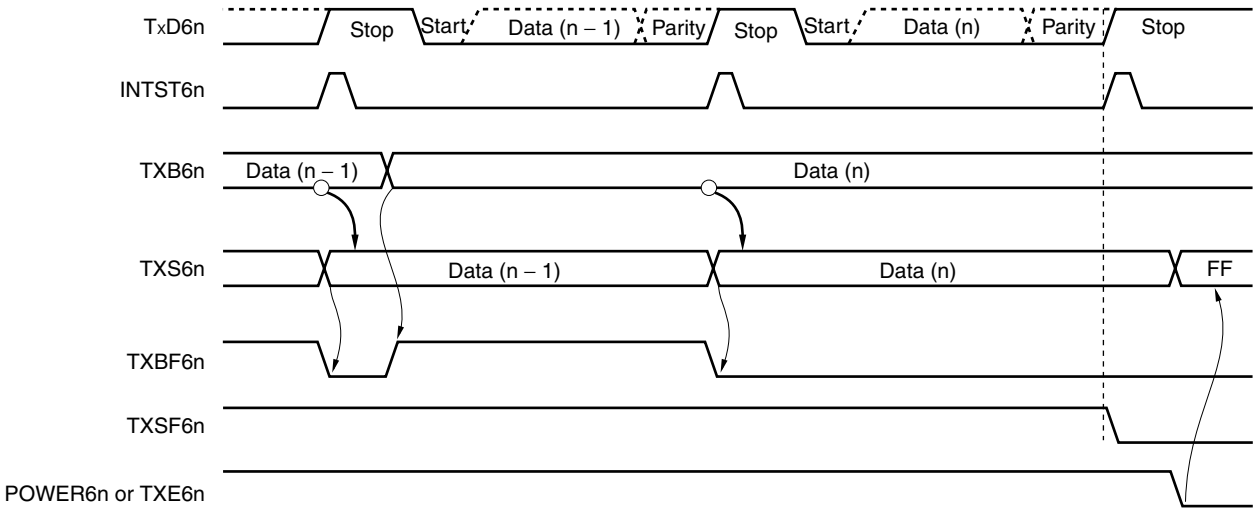
Figure 14-25. Timing of Starting Continuous Transmission



Note When ASIF6n is read, there is a period in which TXBF6n and TXSF6n = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6n bit.

Remark TxD6n: TxD6n pins (output)
INTST6n: Interrupt request signal
TXB6n: Transmit buffer register 6n
TXS6n: Transmit shift register 6n
ASIF6n: Asynchronous serial interface transmission status register 6n
TXBF6n: Bit 1 of ASIF6n
TXSF6n: Bit 0 of ASIF6n
n = 0, 1

Figure 14-26. Timing of Ending Continuous Transmission



Remark TxD6n: TxD6n pins (output)
INTST6n: Interrupt request signal
TXB6n: Transmit buffer register 6n
TXS6n: Transmit shift register 6n
ASIF6n: Asynchronous serial interface transmission status register 6n
TXBF6n: Bit 1 of ASIF6n
TXSF6n: Bit 0 of ASIF6n
POWER6n: Bit 7 of asynchronous serial interface operation mode register (ASIM6n)
TXE6n: Bit 6 of asynchronous serial interface operation mode register (ASIM6n)
n = 0, 1

(e) Normal reception

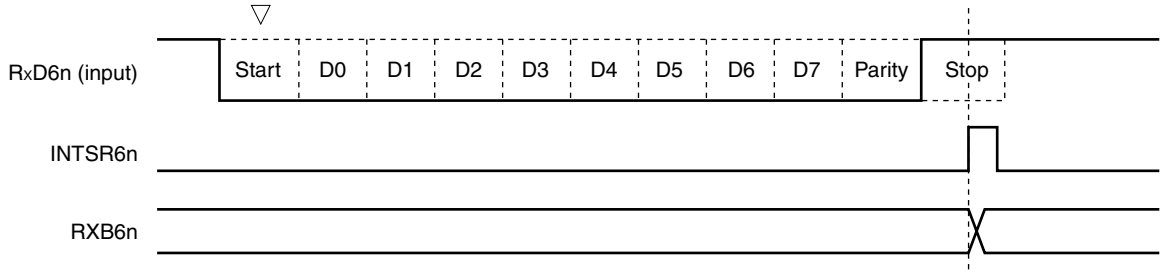
Reception is enabled and the RxD6n pins input is sampled when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and then bit 5 (RXE6n) of ASIM6n is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6n pins input is detected. When the set value of baud rate generator control register 6n (BRGC6n) has been counted, the RxD6n pins input is sampled again (▽ in Figure 14-27). If the RxD6n pins are low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register (RXS6n) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6n) is generated and the data of RXS6n is written to receive buffer register 6n (RXB6n). If an overrun error (OVE6n) occurs, however, the receive data is not written to RXB6n.

Even if a parity error (PE6n) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSR6n/INTSRE6n) is generated on completion of reception.

Figure 14-27. Reception Completion Interrupt Request Timing



Cautions 1. If a reception error occurs, read ASIS6n and then RXB6n to clear the error flag. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 6n (ASIS6n) before reading RXB6n.

Remark n = 0, 1

(f) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6n (ASIS6n) is set as a result of data reception, a reception error interrupt request (INTSR6n/INTSRE6n) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6n in the reception error interrupt servicing (INTSR6n/INTSRE6n) (see **Figure 14-9, 14-10**).

The contents of ASIS6n are cleared to 0 when ASIS6n is read.

Table 14-3. Cause of Reception Error

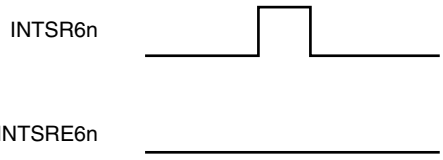
Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6n (RXB6n).

The error interrupt can be separated into reception completion interrupt (INTSR6n) and error interrupt (INTSRE6n) by clearing bit 0 (ISRM6n) of asynchronous serial interface operation mode register 6n (ASIM6n) to 0.

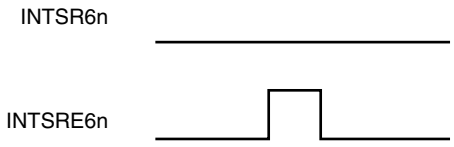
Figure 14-28. Reception Error Interrupt

1. If ISRM6n is cleared to 0 (reception completion interrupt (INTSR6n) and error interrupt (INTSRE6n) are separated)

(a) No error during reception

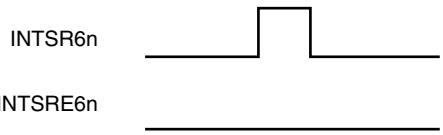


(b) Error during reception

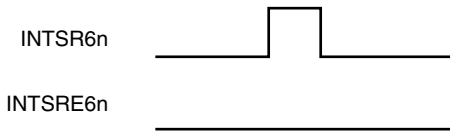


2. If ISRM6n is set to 1 (error interrupt is included in INTSR6n)

(a) No error during reception



(b) Error during reception



Remark n = 0, 1

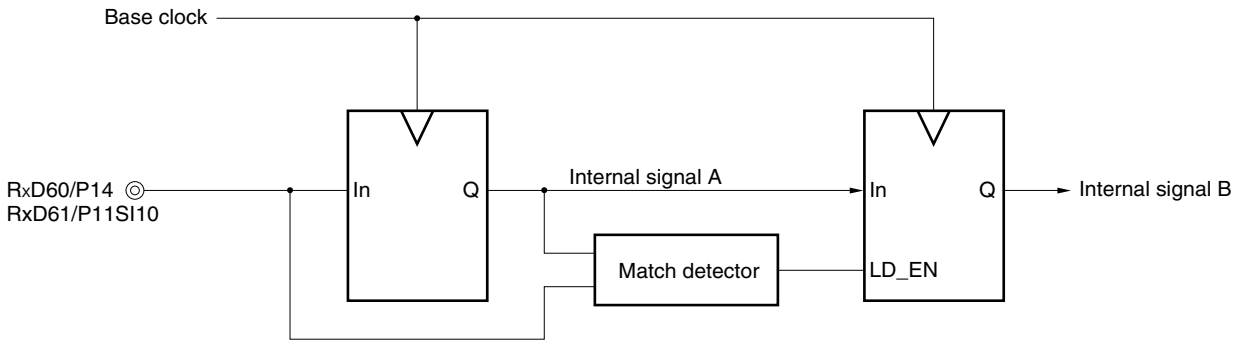
(g) Noise filter of receive data

The RXD6n signal's is sampled with the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 14-29, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 14-29. Noise Filter Circuit



(h) SBF transmission

When the device is used in LIN communication operation, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, see **Figure 14-1 LIN Transmission Operation**.

When bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and bit 6 (TXE6n) of ASIM6n is then set to 1, transmission is enabled.

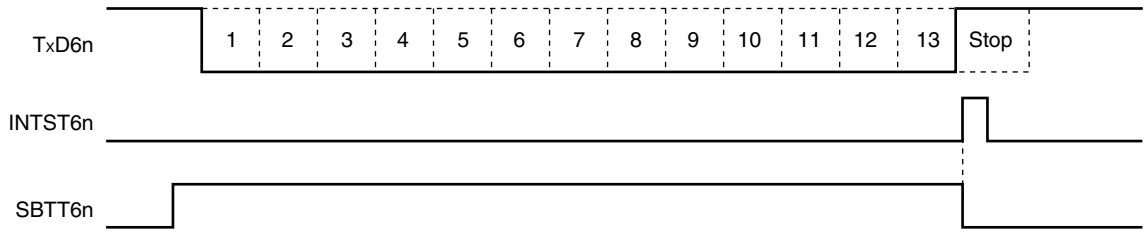
SBF transmission can be started by setting bit 5 (SBTT6n) of asynchronous serial interface control register 6n (ASICL6n) to 1.

Thereafter, a low level of bits 13 to 20 (set by bits 4 to 2 (SBL62n to SBL60n) of ASICL6n) is output. Following the end of SBF transmission, the transmission completion interrupt request (INTST6n) is generated and SBTT6n 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 6n (TXB6n), or until SBTT6n is set to 1.

Remark n = 0, 1

Figure 14-30. SBF Transmission



Remark TxD6n: TxD6n pins (output)
INTST6n: Transmission completion interrupt request
SBTT6n: Bit 5 of asynchronous serial interface control register 6n (ASICL6n)
n = 0, 1

(i) SBF reception

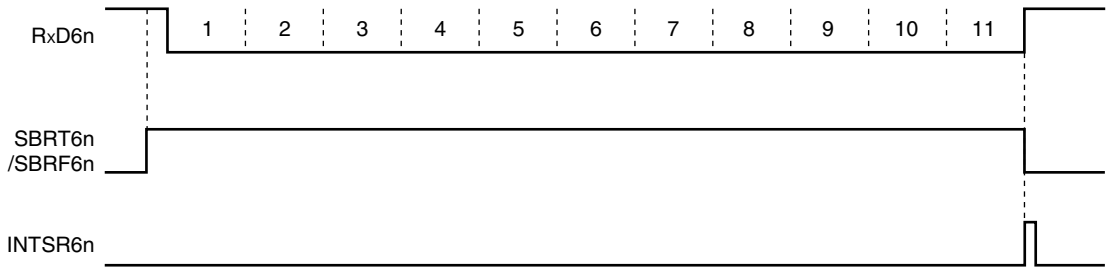
When the device is incorporated in LIN, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, see **Figure 14-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and then bit 5 (RXE6n) of ASIM6n is set to 1. SBF reception is enabled when bit 6 (SBRT6n) of asynchronous serial interface control register 6n (ASICL6n) is set to 1. In the SBF reception enabled status, the RxD6n pins are sampled and the start bit is detected in the same manner as the normal reception enable status.

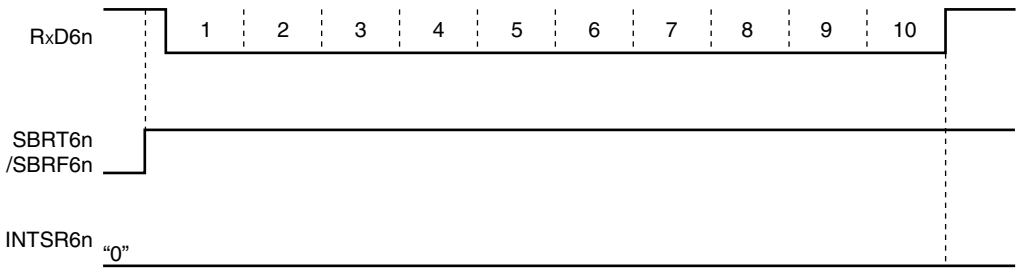
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6n (RXS6n) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6n) is generated as normal processing. At this time, the SBRF6n and SBRT6n bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6n, PE6n, and FE6n (bits 0 to 2 of asynchronous serial interface reception error status register 6n (ASIS6n)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6n (RXS6n) and receive buffer register 6n (RXB6n) 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 SBRF6n and SBRT6n bits are not cleared.

Figure 14-31. 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 RxD6n: RxD6n pins (input)
SBRT6n: Bit 6 of asynchronous serial interface control register 6n (ASICL6n)
SBRF6n: Bit 7 of ASICL6n
INTSR6n: Reception completion interrupt request
n = 0, 1

14.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 8-bit programmable counter, and generates a serial clock for transmission/reception of UART60 and UART61.

Separate 8-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

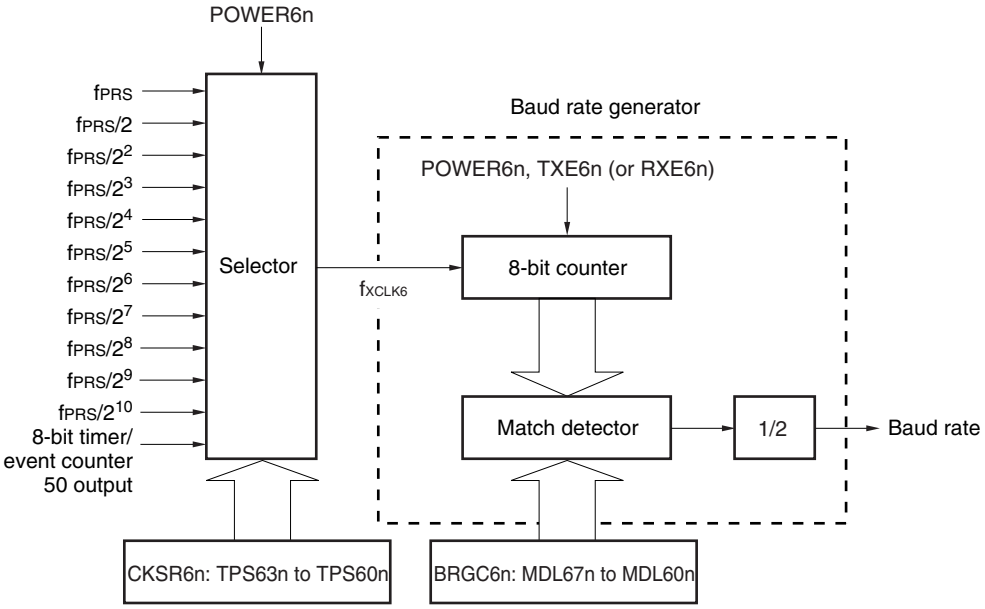
- Base clock
  The clock selected by bits 3 to 0 (TPS63n to TPS60n) of clock selection register 6n (CKSR6n) is supplied to each module when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 1. This clock is called the base clock and its frequency is called f_{XCLK6}. The base clock is fixed to low level when POWER6n = 0.

- Transmission counter
  This counter stops operation, cleared to 0, when bit 7 (POWER6n) or bit 6 (TXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 0.
  It starts counting when POWER6n = 1 and TXE6n = 1.
  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6n (TXB6n).
  If data are continuously transmitted, the counter is cleared to 0 again when one frame of data has been completely transmitted. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until POWER6n or TXE6n is cleared to 0.

- Reception counter
  This counter stops operation, cleared to 0, when bit 7 (POWER6n) or bit 5 (RXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 0.
  It starts counting when the start bit has been detected.
  The counter stops operation after one frame has been received, until the next start bit is detected.

Remark n = 0, 1

Figure 14-32. Configuration of Baud Rate Generator



Remark POWER6n: Bit 7 of asynchronous serial interface operation mode register 6n (ASIM6n)
TXE6n: Bit 6 of ASIM6n
RXE6n: Bit 5 of ASIM6n
CKSR6n: Clock selection register 6n
BRGC6n: Baud rate generator control register 6n
n = 0, 1

(2) Generation of serial clock

A serial clock can be generated by using clock selection register 6n (CKSR6n) and baud rate generator control register 6n (BRGC6n).

Select the clock to be input to the 8-bit counter by using bits 3 to 0 (TPS63n to TPS60n) of CKSR6n.

Bits 7 to 0 (MDL67n to MDL60n) of BRGC6n can be used to select the division value of the 8-bit counter.

(a) Baud rate

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 TPS63n to TPS60n bits of CKSR6n register
k: Value set by MDL67n to MDL60n bits of BRGC6n register (k = 4, 5, 6, ..., 255)

(b) 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 MDL67n to MDL60n 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 14-4. Set Data of Baud Rate Generator

Baud Rate [bps]	f_{PRS} = 5.0 MHz				f_{PRS} = 10.0 MHz				f_{PRS} = 20.0 MHz			
	TPS63n, TPS60n	k	Calculated Value	ERR [%]	TPS63n, TPS60n	k	Calculated Value	ERR [%]	TPS63n, TPS60n	k	Calculated Value	ERR [%]
300	7H	65	301	0.16	8H	65	301	0.16	9H	65	301	0.16
600	6H	65	601	0.16	7H	65	601	0.16	8H	65	601	0.16
1200	5H	65	1202	0.16	6H	65	1202	0.16	7H	65	1202	0.16
2400	4H	65	2404	0.16	5H	65	2404	0.16	6H	65	2404	0.16
4800	3H	65	4808	0.16	4H	65	4808	0.16	5H	65	4808	0.16
9600	2H	65	9615	0.16	3H	65	9615	0.16	4H	65	9615	0.16
19200	1H	65	19231	0.16	2H	65	19231	0.16	3H	65	19231	0.16
24000	3H	13	24038	0.16	4H	13	24038	0.16	5H	13	24038	0.16
31250	4H	5	31250	0	5H	5	31250	0	6H	5	31250	0
38400	0H	65	38462	0.16	1H	65	38462	0.16	2H	65	38462	0.16
48000	2H	13	48077	0.16	3H	13	48077	0.16	4H	13	48077	0.16
76800	0H	33	75758	−1.36	0H	65	76923	0.16	1H	65	76923	0.16
115200	1H	11	113636	−1.36	0H	43	116279	0.94	0H	87	114943	−0.22
153600	1H	8	156250	1.73	0H	33	151515	−1.36	1H	33	151515	−1.36
312500	0H	8	312500	0	1H	8	312500	0	2H	8	312500	0
625000	0H	4	625000	0	1H	4	625000	0	2H	4	625000	0

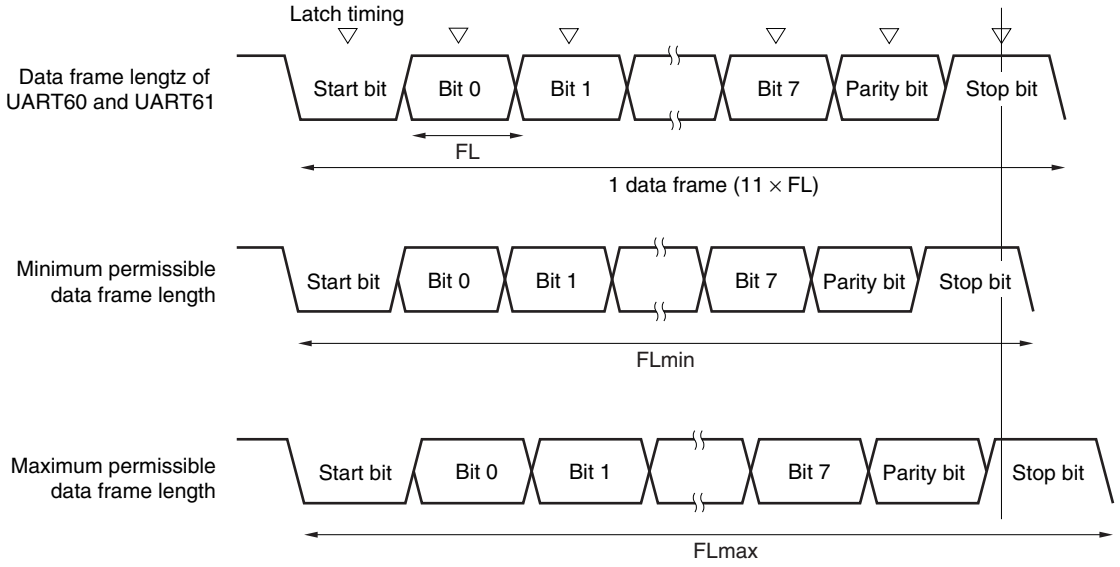
Remark TPS63n to TPS60n: Bits 3 to 0 of clock selection register 6n (CKSR6n) (setting of base clock (f_{XCLK6}))
k: Value set by MDL67n to MDL60n bits of baud rate generator control register 6n (BRGC6n) (k = 4, 5, 6, ..., 255)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error
n = 0, 1

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 14-33. Permissible Baud Rate Range During Reception



As shown in Figure 14-33, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6n (BRGC6n) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART60 and UART61
k: Set value of BRGC6n
FL: 1-bit data length
Margin of latch timing: 2 clocks

Remark n = 0, 1

Minimum permissible data frame length: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

The permissible baud rate error between UART60 and UART61 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 14-5. 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.

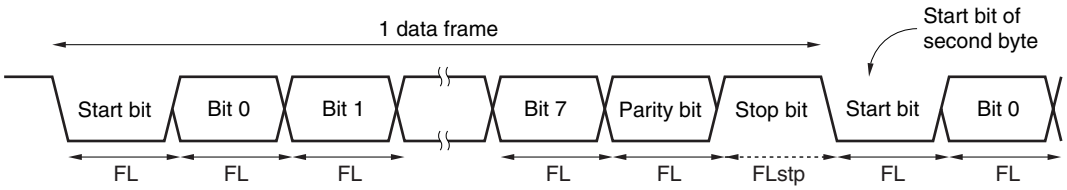
2. k: Set value of BRGC6n

3. n = 0, 1

(5) Data frame length during continuous transmission

When data is continuously transmitted, the data frame length from a stop bit to the next start bit is extended by two clocks of base clock from the normal value. However, the result of communication is not affected because the timing is initialized on the reception side when the start bit is detected.

Figure 14-34. 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 15 SERIAL INTERFACES CSI10 AND CSI11

The 78K0/FE2 incorporate serial interfaces CSI10 and CSI11.

15.1 Functions of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 have the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial communication is not performed and can enable a reduction in the power consumption.

For details, see **15.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to communicate 8-bit data using three lines: a serial clock line ($\overline{\text{SCK1n}}$) and two serial data lines (SI1n and SO1n).

The processing time of data communication can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

For details, see **15.4.2 3-wire serial I/O mode**.

Remark n = 0, 1

15.2 Configuration of Serial Interfaces CSI10 and CSI11

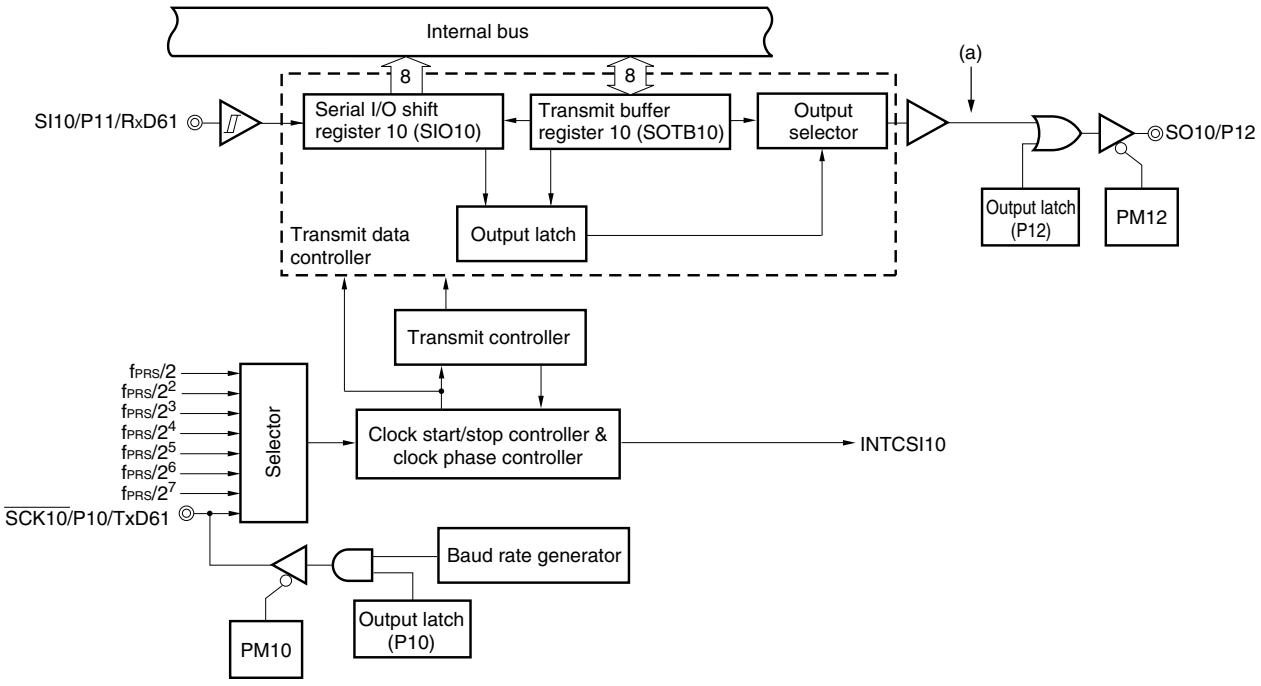
Serial interfaces CSI10 and CSI11 include the following hardware.

Table 15-1. Configuration of Serial Interfaces CSI10 and CSI11

Item	Configuration
Controller	Transmit controller Clock start/stop controller & clock phase controller
Registers	Transmit buffer register 1n (SOTB1n) Serial I/O shift register 1n (SIO1n)
Control registers	Serial operation mode register 1n (CSIM1n) Serial clock selection register 1n (CSIC1n) Port mode register 1 (PM1) or port mode register 7 (PM7), Port mode register 0 (PM0) Port register 1 (P1) or port register 7 (P7), port register 0 (P0)

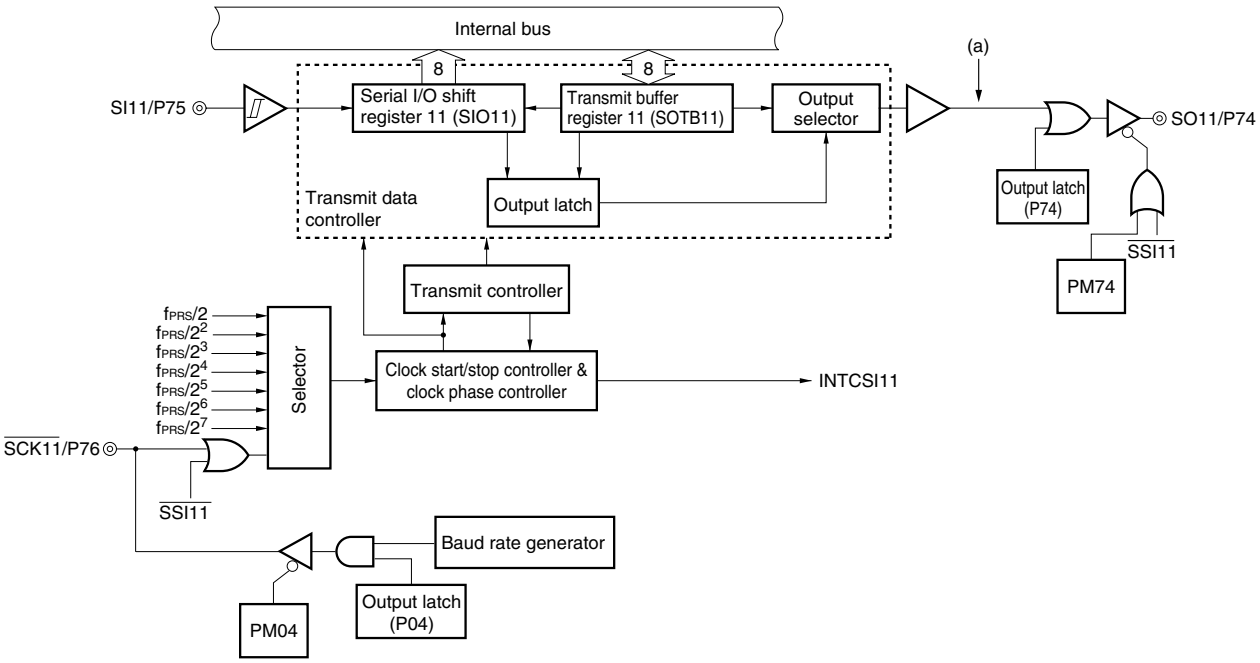
Remark n = 0, 1

Figure 15-1. Block Diagram of Serial Interface CSI10



Remark (a): SO10 output

Figure 15-2. Block Diagram of Serial Interface CSI11



Remark (a): SO11 output

(1) Transmit buffer register 1n (SOTB1n)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB1n when bit 7 (CSIE1n) and bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1.

The data written to SOTB1n is converted from parallel data into serial data by serial I/O shift register 1n, and output to the serial output pin (SO1n).

SOTB1n can be written or read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SOTB1n when CSOT1n = 1 (during serial communication).

2. In the slave mode, transmission/reception is started when data is written to SOTB11 with a low level input to the $\overline{\text{SSI11}}$ pin. For details of the transmission/reception operation, see 15.4.2 (2) Communication operation.

(2) Serial I/O shift register 1n (SIO1n)

This is an 8-bit register that converts data from parallel data into serial data and vice versa.

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

Reception is started by reading data from SIO1n if bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

During reception, the data is read from the serial input pin (SI1n) to SIO1n.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SIO1n when CSOT1n = 1 (during serial communication).

2. In the slave mode, reception is started when data is read from SIO11 with a low level input to the $\overline{\text{SSI11}}$ pin. For details of the reception operation, see 15.4.2 (2) Communication operation.

Remark n = 0, 1

15.3 Registers Controlling Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 are controlled by the following four registers.

- Serial operation mode register 1n (CSIM1n)
- Serial clock selection register 1n (CSIC1n)
- Port mode register 1 (PM1) or port mode register 7 (PM7), port mode register 0 (PM0)
- Port register 1 (P1) or port register 7 (P7), port mode register 0(P0)

(1) Serial operation mode register 1n (CSIM1n)

CSIM1n is used to select the operation mode and enable or disable operation.

CSIM1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 15-3. Format of Serial Operation Mode Register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 2] and asynchronously resets the internal circuit[Note 3].
1	Enables operation

TRMD10[Note 4]	Transmit/receive mode control
0[Note 5]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10[Note 6]	First bit specification
0	MSB
1	LSB

CSOT10	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. To use P10/$\overline{\text{SCK10}}$/TxD61 and P12/SO10 as general-purpose ports, set CSIM10 in the default status (00H).

3. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

4. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

5. The SO10 output (see **(a)** in **Figure 15-1**) is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

6. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

Caution Be sure to clear bit 5 to 0.

Figure 15-4. 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 P74/SO11, P76/$\overline{\text{SCK11}}$, and P05/$\overline{\text{SSI11}}$/TI001 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 (see **(a)** in **Figure 15-2**) 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 1n (CSIC1n)

This register specifies the timing of the data transmission/reception and sets the serial clock.

CSIC1n can be set by a 1-bit or 8-bit memory manipulation instruction.

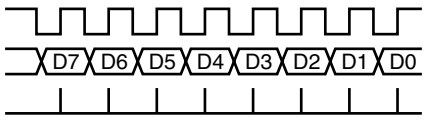
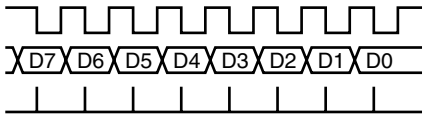
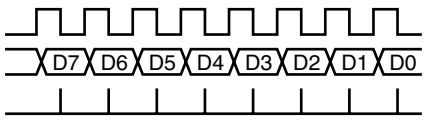
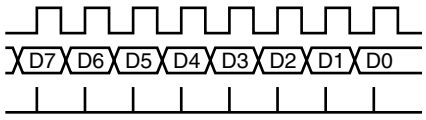
Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 15-5. Format of Serial Clock Selection Register 10 (CSIC10)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	Specification of data transmission/reception timing	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	CSI10 serial clock selection					Mode
				f_{PRS} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	$f_{PRS}/2$	2 MHz	2.5 MHz	5 MHz	10 MHz	Master mode
0	0	1	$f_{PRS}/2^2$	1 MHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	500 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	250 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	125 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	62.5 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	31.25 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	External clock input to $\overline{SCK10}$					Slave mode

Cautions 1. Do not write to CSIC10 while CSIE10 = 1 (operation enabled).

2. To use P10/$\overline{SCK10}$/TxD61 and P12/SO10 as general-purpose ports, set CSIC10 in the default status (00H).

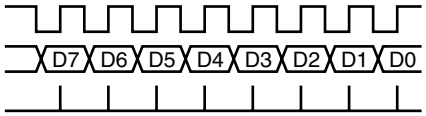
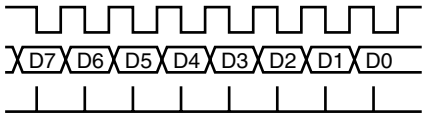
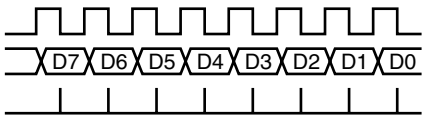
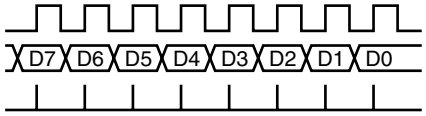
3. The phase type of the data clock is type 1 after reset.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 15-6. 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} = 4 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	$f_{PRS}/2$	2 MHz	2.5 MHz	5 MHz	10 MHz	Master mode
0	0	1	$f_{PRS}/2^2$	1 MHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	500 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	250 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	125 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	62.5 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	31.25 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	External clock input to $\overline{SCK11}$					Slave mode

Cautions 1. Do not write to CSIC11 while CSIE11 = 1 (operation enabled).

2. To use P74/SO11 and P76/$\overline{SCK11}$ 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, 1 and 7 (PM0, PM1, PM7)

These registers set port 0, 1 and 7 input/output in 1-bit units.

When using P10/$\overline{\text{SCK10}}$ and P76/$\overline{\text{SCK11}}$ as the clock output pins of the serial interface, clear PM10 and PM76, and the output latches of P10 and P76 to 1.

When using P12/SO10 and P74/SO11 as the data output pins of the serial interface, clear PM12, PM74, P12 and P74 to 0.

When using P10/$\overline{\text{SCK10}}$/TxD61 and P76/$\overline{\text{SCK11}}$ as the clock input pins of the serial interface, P11/SI10/RxD61 and P75/SI11 as the data input pins, and P05/$\overline{\text{SSI11}}$/TI001 as the chip select input pin, set PM10, PM76, PM11, PM75 and PM05 to 1. At this time, the output latches of P10, P76, P11, P75 and P05 may be 0 or 1.

PM0, PM1 and PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 15-7. 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	PM06	PM05	1	1	1	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0, 1, 5, 6)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

Figure 15-8. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

Figure 15-9. Format of Port Mode Register 7 (PM7)

Address: FF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 6)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

15.4 Operation of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

15.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P10/$\overline{\text{SCK10}}$/TxD61, P11/SI10/RxD61, P12/SO10, P74/SO11, P75/SI11, and P76/$\overline{\text{SCK11}}$ 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 1n (CSIM1n).

To set the operation stop mode, clear bit 7 (CSIE1n) of CSIM1n to 0.

(a) Serial operation mode register 1n (CSIM1n)

CSIM1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CSIM1n to 00H.

Remark n = 0, 1

- Serial operation mode register 10 (CSIM10)

Address: FF80H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. To use P10/$\overline{\text{SCK10}}$/TxD61 and P12/SO10 as general-purpose ports, set CSIM10 in the default status (00H).

2. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

- Serial operation mode register 11 (CSIM11)

Address: FF88H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11	DIR11	0	0	0	CSOT11

CSIE11	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. To use P74/SO11, P76/$\overline{\text{SCK11}}$, and P05/$\overline{\text{SSI11}}$/TI001 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 ($\overline{SCK1n}$), serial output (SO1n), and serial input (SI1n) lines.

(1) Registers used

- Serial operation mode register 1n (CSIM1n)
- Serial clock selection register 1n (CSIC1n)
- Port mode register 1 (PM1) or port mode register 7 (PM7)
- Port register 1 (P1) or port register 7 (P7)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC1n register (see **Figures 15-5** and **15-6**).
<2> Set bits 0 and 4 to 6 (CSOT1n, DIR1n, SSE11 (serial interface CSI11 only), and TRMD1n) of the CSIM1n register (see **Figures 15-3** and **15-4**).
<3> Set bit 7 (CSIE1n) of the CSIM1n register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 1n (SOTB1n). → Data transmission/reception is started.
Read data from serial I/O shift register 1n (SIO1n). → Data reception is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

Remark n = 0, 1

The relationship between the register settings and pins is shown below.

Table 15-2. Relationship Between Register Settings and Pins (1/2)

(a) Serial interface CSI10

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 Operation	Pin Function		
									SI10/RxD61/ P11	SO10/P12	$\overline{\text{SCK10}}$/TxD61/ P10
0	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	Stop	RxD61/ P11	P12	TxD61/ P10Note 2
1	0	1	×	×Note 1	×Note 1	1	×	Slave reception Note 3	SI10	P12	$\overline{\text{SCK10}}$ (input)Note 3
1	1	×Note 1	×Note 1	0	0	1	×	Slave transmission Note 3	RxD61/ P11	SO10	$\overline{\text{SCK10}}$ (input)Note 3
1	1	1	×	0	0	1	×	Slave transmission/ reception Note 3	SI10	SO10	$\overline{\text{SCK10}}$ (input)Note 3
1	0	1	×	×Note 1	×Note 1	0	1	Master reception	SI10	P12	$\overline{\text{SCK10}}$ (output)
1	1	×Note 1	×Note 1	0	0	0	1	Master transmission	RxD61/ P11	SO10	$\overline{\text{SCK10}}$ (output)
1	1	1	×	0	0	0	1	Master transmission/ reception	SI10	SO10	$\overline{\text{SCK10}}$ (output)

Notes 1. Can be set as port function.

2. To use P10/$\overline{\text{SCK10}}$/TxD61 as port pins, clear CKP10 to 0.

3. To use the slave mode, set CKS102, CKS101, and CKS100 to 1, 1, 1.

Remark ×: don't care
CSIE10: Bit 7 of serial operation mode register 10 (CSIM10)
TRMD10: Bit 6 of CSIM10
CKP10: Bit 4 of serial clock selection register 10 (CSIC10)
CKS102, CKS101, CKS100: Bits 2 to 0 of CSIC10
PM1×: Port mode register
P1×: Port output latch

Table 15-2. Relationship Between Register Settings and Pins (2/2)

(b) Serial interface CSI11

CSIE11	TRMD11	SSE11	PM75	P75	PM74	P74	PM76	P76	PM05	P05	CSI11 Operation	Pin Function			
												SI11/ P75	SO11/ P74	$\overline{\text{SCK11}}$/ P76	$\overline{\text{SSI11}}$/ TI001/P05
0	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	Stop	P75	P74	P76Note 2	TI001/ P05
1	0	0	1	×	×Note 1	×Note 1	1	×	×Note 1	×Note 1	Slave reception Note 3	SI11	P74	$\overline{\text{SCK11}}$ (input) Note 3	TI001/ P05
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	×Note 1	×Note 1	0	0	1	×	×Note 1	×Note 1	Slave transmission Note 3	P75	SO11	$\overline{\text{SCK11}}$ (input) Note 3	TI001/ P05
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	1	×	0	0	1	×	×Note 1	×Note 1	Slave transmission/ reception Note 3	SI11	SO11	$\overline{\text{SCK11}}$ (input) Note 3	TI001/ P05
		1							1	×					$\overline{\text{SSI11}}$
1	0	0	1	×	×Note 1	×Note 1	0	1	×Note 1	×Note 1	Master reception	SI11	P74	$\overline{\text{SCK11}}$ (output)	TI001/ P05
1	1	0	×Note 1	×Note 1	0	0	0	1	×Note 1	×Note 1	Master transmission	P75	SO11	$\overline{\text{SCK11}}$ (output)	TI001/ P05
1	1	0	1	×	0	0	0	1	×Note 1	×Note 1	Master transmission/ reception	SI11	SO11	$\overline{\text{SCK11}}$ (output)	TI001/ P05

Notes 1. Can be set as port function.

2. To use P76/$\overline{\text{SCK11}}$ as port pins, clear CKP11 to 0.

3. To use the slave mode, set CKS112, CKS111, and CKS110 to 1, 1, 1.

Remark ×: don't care
CSIE11: Bit 7 of serial operation mode register 11 (CSIM11)
TRMD11: Bit 6 of CSIM11
CKP11: Bit 4 of serial clock selection register 11 (CSIC11)
CKS112, CKS111, CKS110: Bits 2 to 0 of CSIC11
PM7×: Port mode register 7×
P7×: Port 7× output latch
PM05: Port mode register 05
P05: Port 05 output latch

(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1. Transmission/reception is started when a value is written to transmit buffer register 1n (SOTB1n). In addition, data can be received when bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

Reception is started when data is read from serial I/O shift register 1n (SIO1n).

However, communication is performed as follows if bit 5 (SSE11) of CSIM11 is 1 when serial interface CSI11 is in the slave mode.

<1> Low level input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception is started when SOTB11 is written, or reception is started when SIO11 is read.

<2> High level input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception or reception is held, therefore, even if SOTB11 is written or SIO11 is read, transmission/reception or reception will not be started.

<3> Data is written to SOTB11 or data is read from SIO11 while a high level is input to the $\overline{\text{SSI11}}$ pin, then a low level is input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception or reception is started.

<4> A high level is input to the $\overline{\text{SSI11}}$ pin during transmission/reception or reception
→ Transmission/reception or reception is suspended.

After communication has been started, bit 0 (CSOT1n) of CSIM1n is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF1n) is set, and CSOT1n is cleared to 0. Then the next communication is enabled.

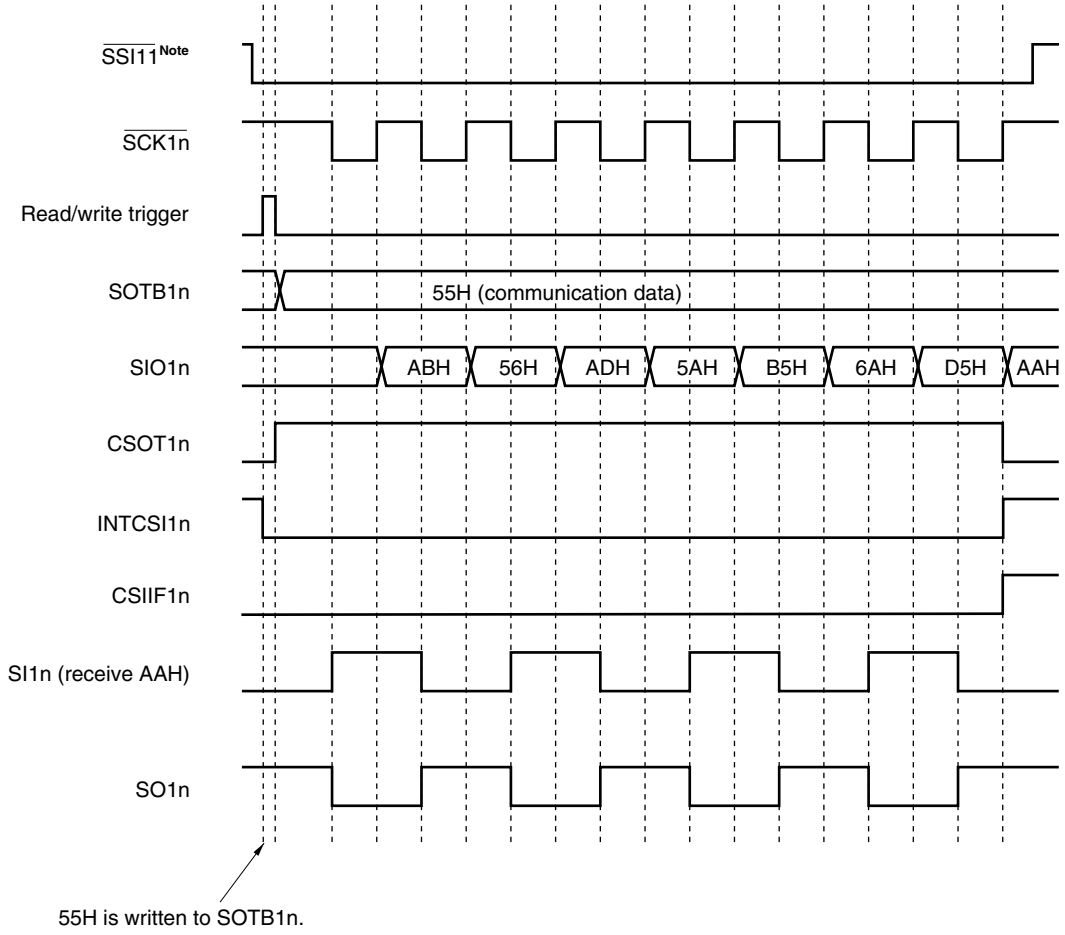
Cautions 1. Do not access the control register and data register when CSOT1n = 1 (during serial communication).

2. When using serial interface CSI11, wait for the duration of at least one clock before the clock operation is started to change the level of the $\overline{\text{SSI11}}$ pin in the slave mode; otherwise, malfunctioning may occur.

Remark n = 0, 1

Figure 15-10. Timing in 3-Wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 0, SSE11 = 1[Note])

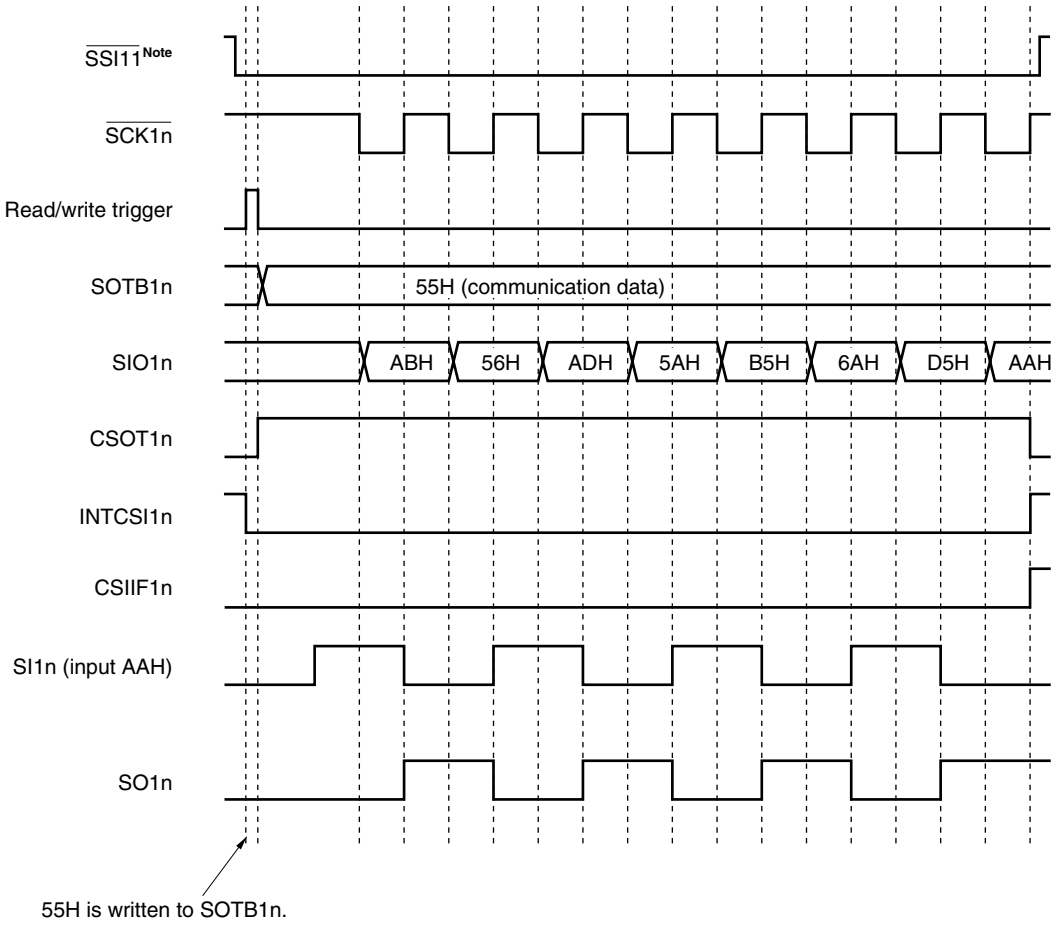


Note The SSE11 flag and $\overline{\text{SSI11}}$ pin are available only for serial interface CSI11, and are used in the slave mode.

Remark n = 0, 1

Figure 15-10. Timing in 3-Wire Serial I/O Mode (2/2)

(2) Transmission/reception timing (Type 2; TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 1, SSE11 = 1[Note])

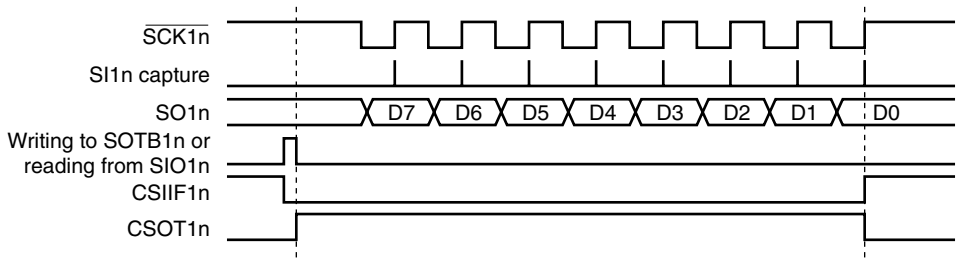


Note The SSE11 flag and $\overline{\text{SSI11}}$ pin are available only for serial interface CSI11, and are used in the slave mode.

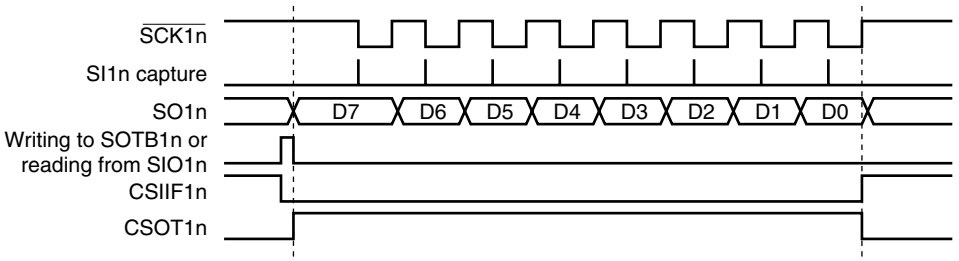
Remark n = 0, 1

Figure 15-11. Timing of Clock/Data Phase

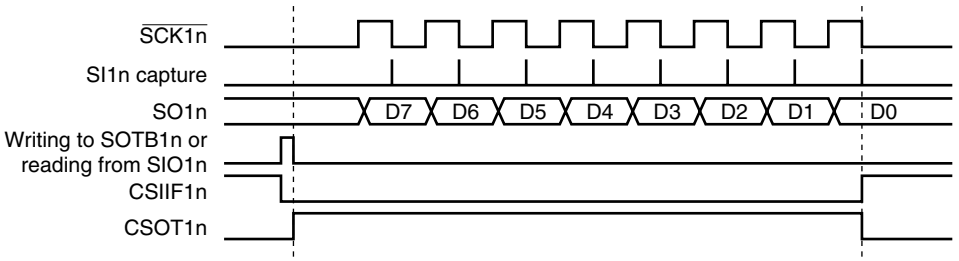
(a) Type 1; CKP1n = 0, DAP1n = 0, DIR1n = 0



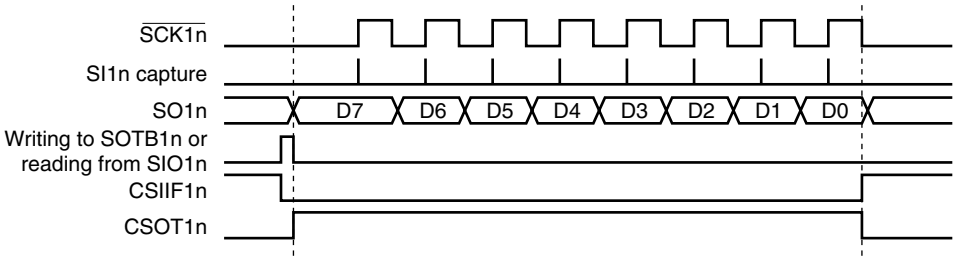
(b) Type 2; CKP1n = 0, DAP1n = 1, DIR1n = 0



(c) Type 3; CKP1n = 1, DAP1n = 0, DIR1n = 0



(d) Type 4; CKP1n = 1, DAP1n = 1, DIR1n = 0



Remarks 1. The above figure illustrates a communication operation where data is transmitted with the MSB first.

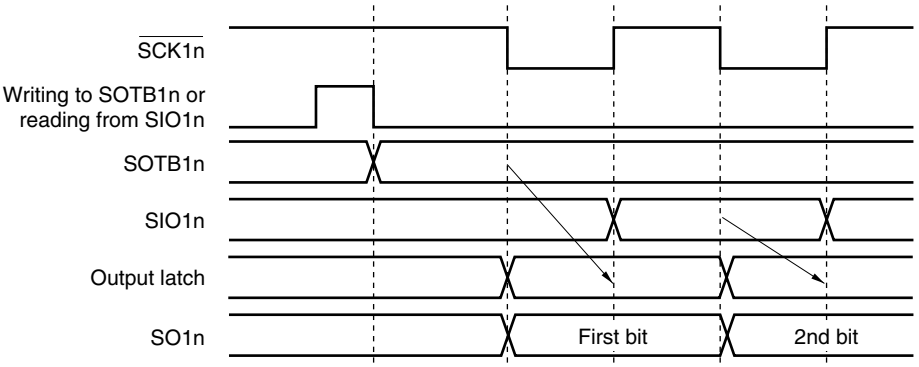
2. n = 0, 1

(3) Timing of output to SO1n pin (first bit)

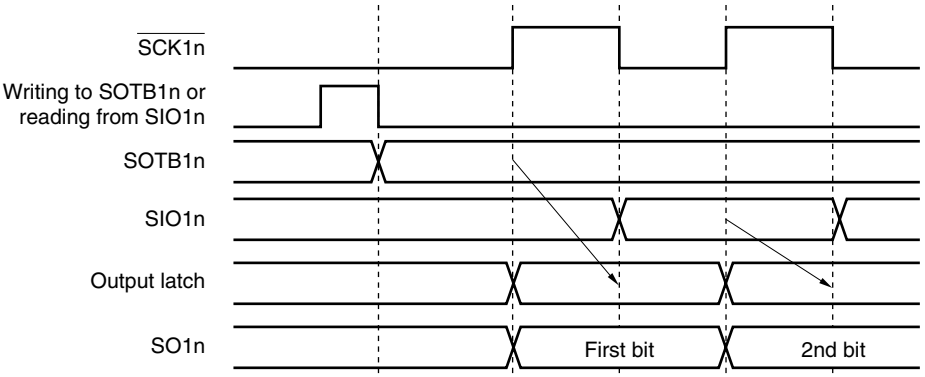
When communication is started, the value of transmit buffer register 1n (SOTB1n) is output from the SO1n pin. The output operation of the first bit at this time is described below.

Figure 15-12. Output Operation of First Bit (1/2)

(a) Type 1: CKP1n = 0, DAP1n = 0



(b) Type 3: CKP1n = 1, DAP1n = 0



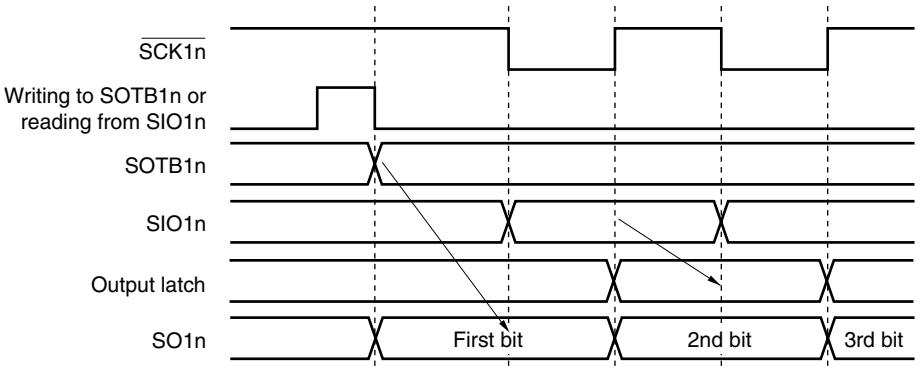
The first bit is directly latched by the SOTB1n register to the output latch at the falling (or rising) edge of $\overline{\text{SCK1n}}$, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next rising (or falling) edge of $\overline{\text{SCK1n}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin.

The second and subsequent bits are latched by the SIO1n register to the output latch at the next falling (or rising) edge of $\overline{\text{SCK1n}}$, and the data is output from the SO1n pin.

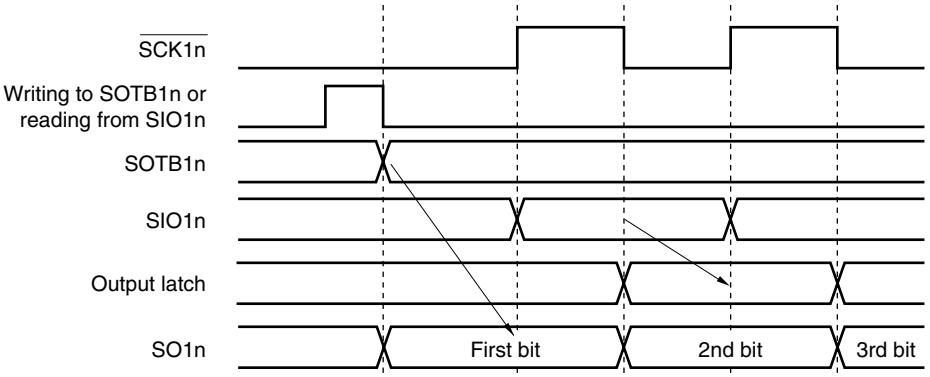
Remark n = 0, 1

Figure 15-12. Output Operation of First Bit (2/2)

(c) Type 2: CKP1n = 0, DAP1n = 1



(d) Type 4: CKP1n = 1, DAP1n = 1



The first bit is directly latched by the SOTB1n register at the falling edge of the write signal of the SOTB1n register or the read signal of the SIO1n register, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next falling (or rising) edge of $\overline{\text{SCK1n}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin.
The second and subsequent bits are latched by the SIO1n register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK1n}}$, and the data is output from the SO1n pin.

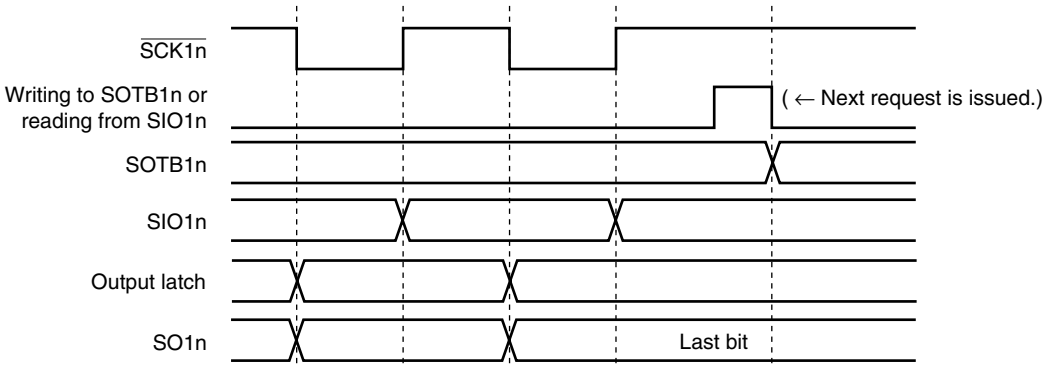
Remark n = 0, 1

(4) Output value of SO1n pin (last bit)

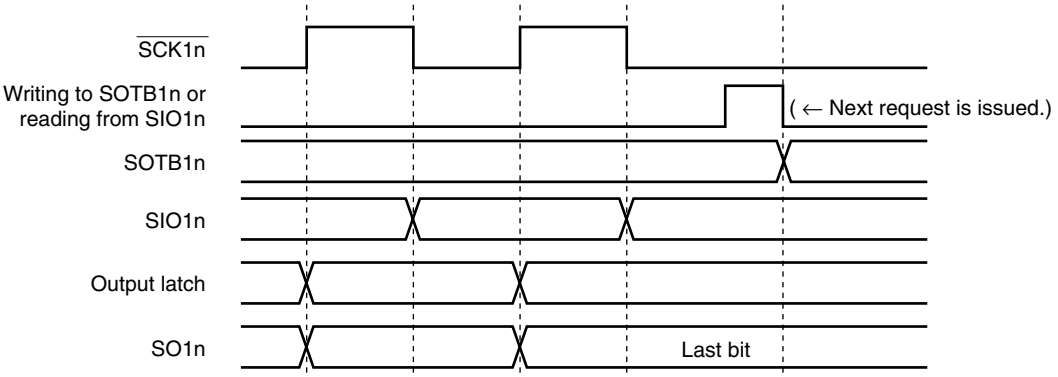
After communication has been completed, the SO1n pin holds the output value of the last bit.

Figure 15-13. Output Value of SO1n Pin (Last Bit) (1/2)

(a) Type 1: CKP1n = 0, DAP1n = 0



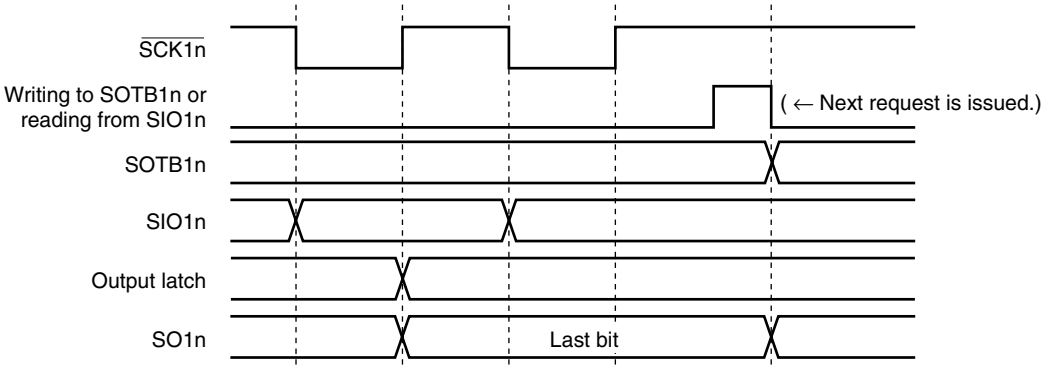
(b) Type 3: CKP1n = 1, DAP1n = 0



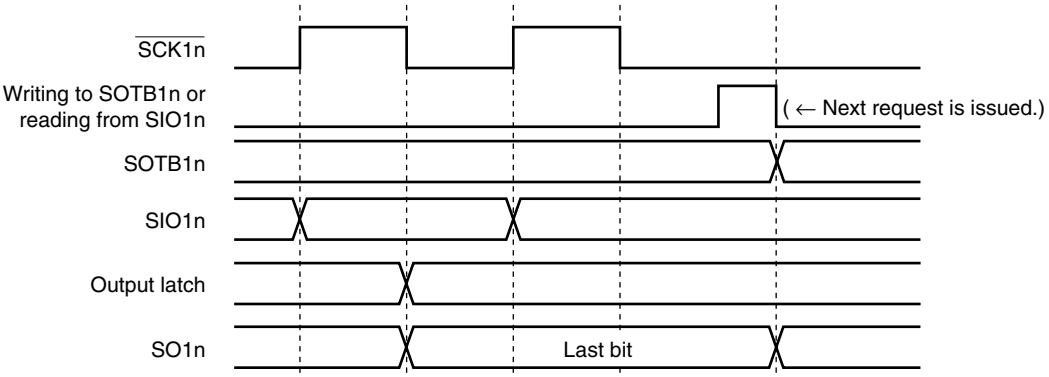
Remark n = 0, 1

Figure 15-13. Output Value of SO1n Pin (Last Bit) (2/2)

(c) Type 2: CKP1n = 0, DAP1n = 1



(d) Type 4: CKP1n = 1, DAP1n = 1



Remark n = 0, 1

(5) SO1n output (see (a) in Figures 15-1 and 15-2)

The status of the SO1n output is as follows if bit 7 (CSIE1n) of serial operation mode register 1n (CSIM1n) is cleared to 0.

Table 15-3. SO1n Output Status

TRMD1n	DAP1n	DIR1n	SO1n Output[Note 1]
TRMD1n = 0[Note]	–	–	Outputs low level[Note 2]
TRMD1n = 1	DAP1n = 0	–	Value of SO1n latch (low-level output)
	DAP1n = 1	DIR1n = 0	Value of bit 7 of SOTB1n
		DIR1n = 1	Value of bit 0 of SOTB1n

Notes 1. The actual output of the SO10/P12 or SO11/P74 pin is determined according to PM12 and P12 or PM74 and P74, as well as the SO1n output.

2. Status after reset

Caution If a value is written to TRMD1n, DAP1n, and DIR1n, the output value of SO1n changes.

Remark n = 0, 1

CHAPTER 16 CAN CONTROLLER

16.1 Outline Description

This product features an on-chip 1-channel CAN (Controller Area Network) controller that complies with CAN protocol as standardized in ISO 11898.

16.1.1 Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max. (CAN clock input ≥ 8 MHz)
- 16 message buffers/1 channel
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel

16.1.2 Overview of functions

Table 16-1 presents an overview of the CAN controller functions.

Table 16-1. Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input ≥ 8 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	- 16 message buffers/1 channel - Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	- Unique ID can be set to each message buffer. - Mask setting of four patterns is possible for each channel. - A receive completion interrupt is generated each time a message is received and stored in a message buffer. - Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function). - Receive history list function
Message transmission	- Unique ID can be set to each message buffer. - Transmit completion interrupt for each message buffer - Message buffer number 0 to 7 specified as the transmit message buffer can be used for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as “ABT”)). - Transmission history list function
Remote frame processing	Remote frame processing by transmit message buffer
Time stamp function	- The time stamp function can be set for a message reception when a 16-bit timer is used in combination. Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected.).
Diagnostic function	- Readable error counters - “Valid protocol operation flag” for verification of bus connections - Receive-only mode - Single-shot mode - CAN protocol error type decoding - Self-test mode
Forced release from bus-off state	- Forced release from bus-off (by ignoring timing constraint) possible by software. - No automatic release from bus-off (software must re-enable).
Power save mode	- CAN sleep mode (can be woken up by CAN bus) - CAN stop mode (cannot be woken up by CAN bus)

16.1.3 Configuration

The CAN controller is composed of the following four blocks.

(1) NPB interface

This functional block provides an NPB (NEC peripheral I/O bus) interface and means of transmitting and receiving signals between the CAN module and the host CPU.

(2) MAC (Memory Access Controller)

This functional block controls access to the CAN protocol layer and to the CAN RAM within the CAN module.

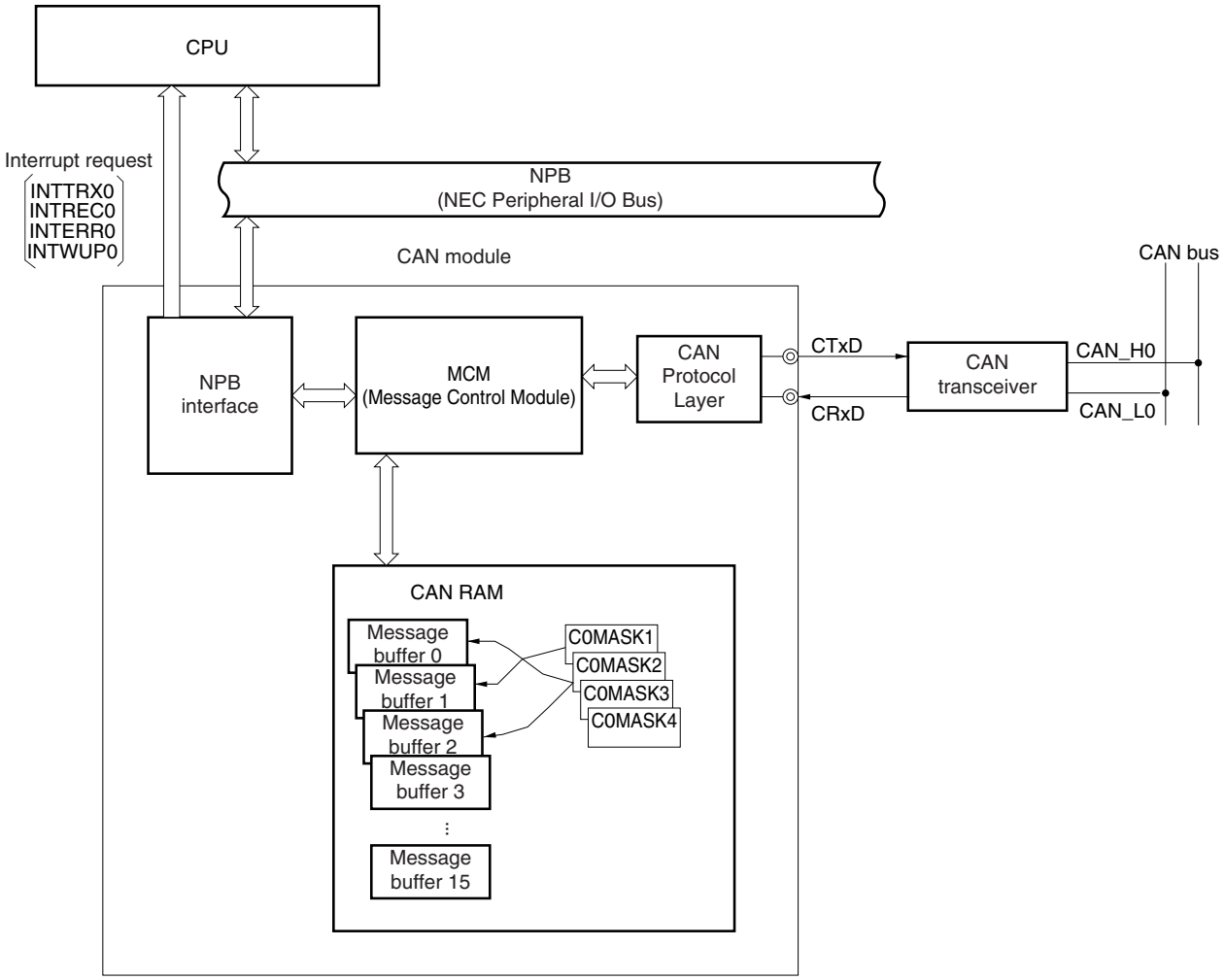
(3) CAN protocol layer

This functional block is involved in the operation of the CAN protocol and its related settings.

(4) CAN RAM

This is the CAN memory functional block, which is used to store message IDs, message data, etc.

Figure 16-1. Block Diagram of CAN Module



16.2 CAN Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed by ISO 11898. For details, refer to the ISO 11898 specifications.

The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link and medium access control. The composition of these layers is illustrated below.

Figure 16-2. Composition of Layers

Higher

Lower

Data link layer[Note]	• Logical link control (LLC)	• Acceptance filtering • Overload report • Recovery management
	• Medium access control (MAC)	• Data capsuled/not capsuled • Frame coding (stuffing/not stuffing) • Medium access management • Error detection • Error report • Acknowledgement • Seriated/not seriated
Physical layer		Prescription of signal level and bit description

Note CAN controller specification

16.2.1 Frame format

(1) Standard format frame

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2048 messages.

(2) Extended format frame

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers which increase the number of messages that can be handled to 2048 x 218 messages.
- Extended format frame is set when “recessive level” (CMOS level equals “1”) is set for both the SRR and IDE bits in the arbitration field.

16.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 16-2. Frame Types

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

(1) Bus value

The bus values are divided into dominant and recessive.

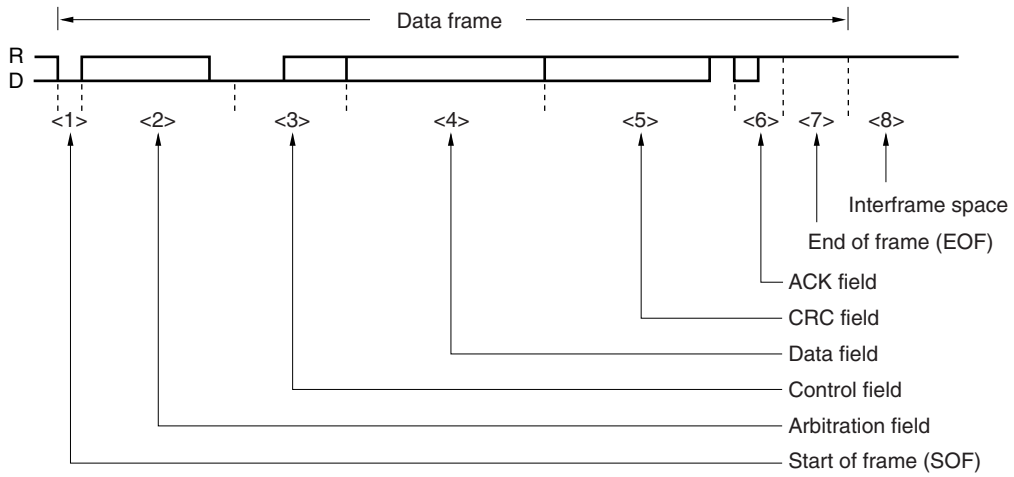
- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

16.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 16-3. Data Frame

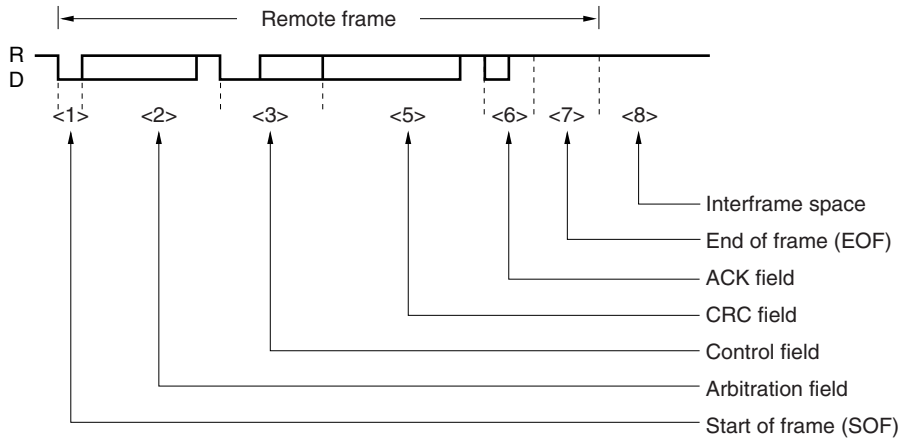


Remark D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 16-4. Remote Frame



Remarks 1. The data field is not transferred even if the control field's data length code is not "0000B".

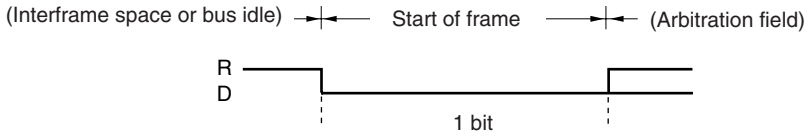
2. D: Dominant = 0
R: Recessive = 1

(3) Description of fields

<1> Start of frame (SOF)

The start of frame field is located at the start of a data frame or remote frame.

Figure 16-5. Start of Frame (SOF)



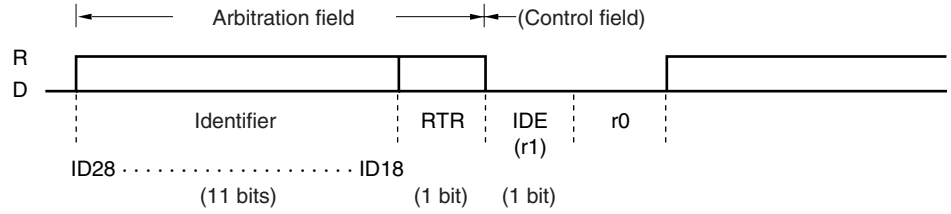
Remark D: Dominant = 0
R: Recessive = 1

- If dominant level is detected in the bus idle state, a hard-synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If dominant level is sampled at the sample point following such a hard-synchronization, the bit is assigned to be a SOF. If recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a disturbance only. No error frame is generated in such case.

<2> Arbitration field

The arbitration field is used to set the priority, data frame/remote frame, and frame format.

Figure 16-6. Arbitration Field (in Standard Format Mode)



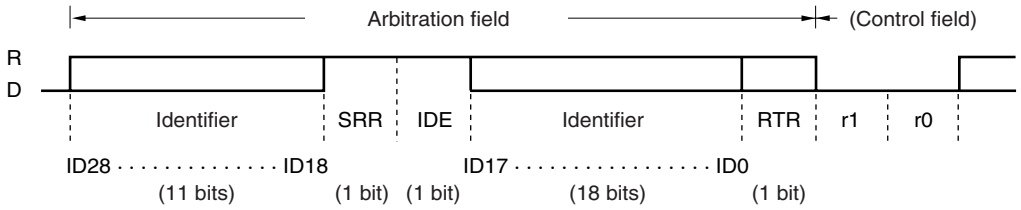
Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Figure 16-7. Arbitration Field (in Extended Format Mode)



Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Table 16-3. RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

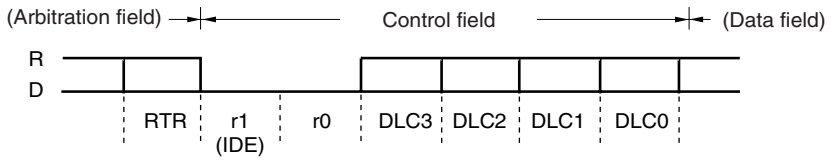
Table 16-4. Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

Frame Format	SRR Bit	IDE Bit	Number. of Bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

<3> Control field

The control field sets “N” as the number of data bytes in the data field (N = 0 to 8).

Figure 16-8. Control Field



Remark D: Dominant = 0
R: Recessive = 1

In a standard format frame, the control field’s IDE bit is the same as the r1 bit.

Table 16-5. Data Length Setting

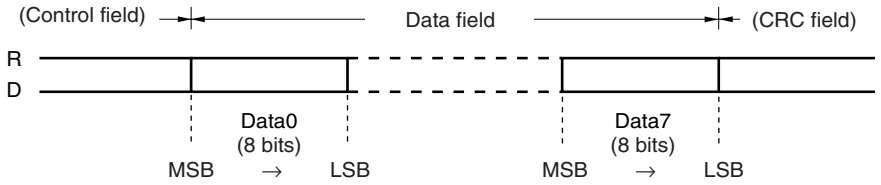
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

Caution In the remote frame, there is no data field even if the data length code is not 0000B.

<4> Data field

The data field contains the amount of data (byte units) set by the control field. Up to 8 units of data can be set.

Figure 16-9. Data Field

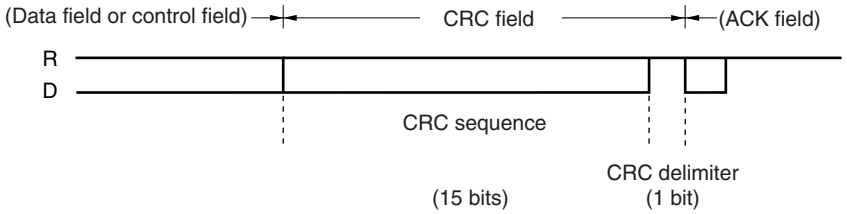


Remark D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 16-10. CRC Field



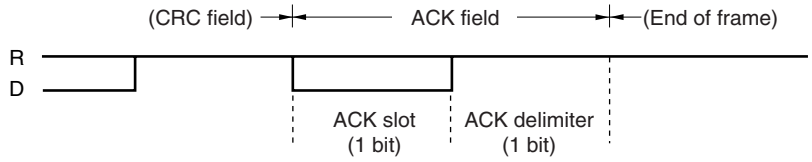
Remark D: Dominant = 0
R: Recessive = 1

- The polynomial P(X) used to generate the 15-bit CRC sequence is expressed as follows.
  P(X) = X15 + X14 + X10 + X8 + X7 + X4 + X3 + 1
- Transmitting node: Transmits the CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, and data field.
- Receiving node: Compares the CRC sequence calculated using data bits that exclude the stuffing bits in the receive data with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node issues an error frame.

<6> ACK field

The ACK field is used to acknowledge normal reception.

Figure 16-11. ACK Field



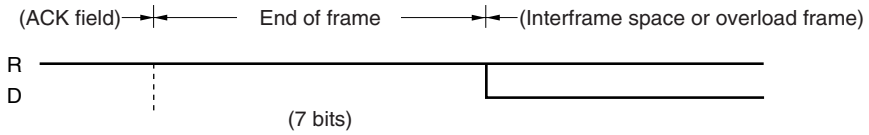
Remark D: Dominant = 0
R: Recessive = 1

- If no CRC error is detected, the receiving node sets the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

<7> End of frame (EOF)

The end of frame field indicates the end of data frame/remote frame.

Figure 16-12. End of Frame (EOF)



Remark D: Dominant = 0
R: Recessive = 1

<8> Interframe space

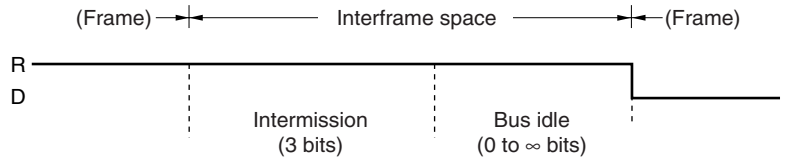
The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- The bus state differs depending on the error status.

(a) Error active node

The interframe space consists of a 3-bit intermission field and a bus idle field.

Figure 16-13. Interframe Space (Error Active Node)



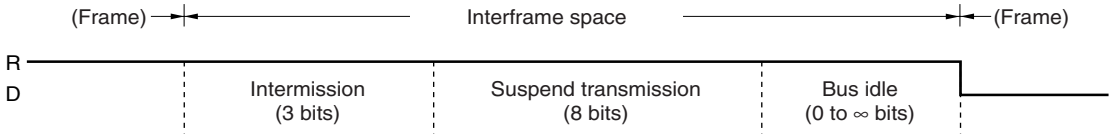
Remarks 1. Bus idle: State in which the bus is not used by any node.

2. D: Dominant = 0
R: Recessive = 1

(b) Error passive node

The interframe space consists of an intermission field, a suspend transmission field, and a bus idle field.

Figure 16-14. Interframe Space (Error Passive Node)



Remarks 1. Bus idle: State in which the bus is not used by any node.
Suspend transmission: Sequence of 8 recessive-level bits transmitted from the node in the error passive status.

2. D: Dominant = 0
R: Recessive = 1

Usually, the intermission field is 3 bits. If the transmitting node detects a dominant level at the third bit of the intermission field, however, it executes transmission.

- Operation in error status

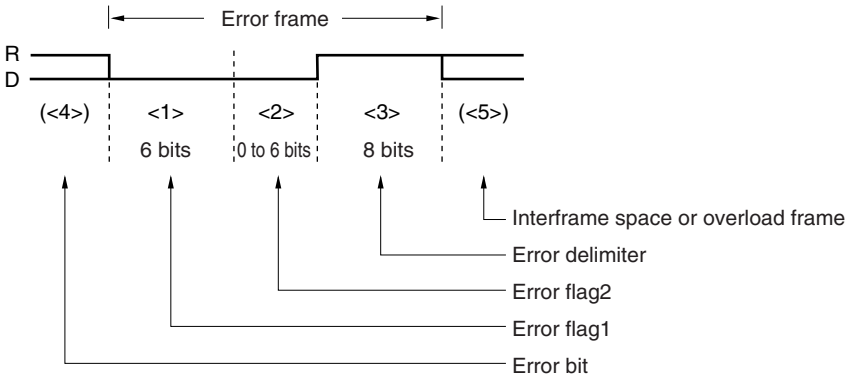
Table 16-6. Operation in Error Status

Error Status	Operation
Error active	A node in this status can transmit immediately after a 3-bit intermission.
Error passive	A node in this status can transmit 8 bits after the intermission.

16.2.4 Error frame

An error frame is output by a node that has detected an error.

Figure 16-15. Error Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 16-7. Definition Error Frame Fields

No.	Name	Bit Count	Definition
<1>	Error flag1	6	Error active node: Outputs 6 dominant-level bits consecutively. Error passive node: Outputs 6 recessive-level bits consecutively. If another node outputs a dominant level while one node is outputting a passive error flag, the passive error flag is not cleared until the same level is detected 6 bits in a row.
<2>	Error flag2	0 to 6	Nodes receiving error flag 1 detect bit stuff errors and issues this error flag.
<3>	Error delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Error bit	–	The bit at which the error was detected. The error flag is output from the bit next to the error bit. In the case of a CRC error, this bit is output following the ACK delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.5

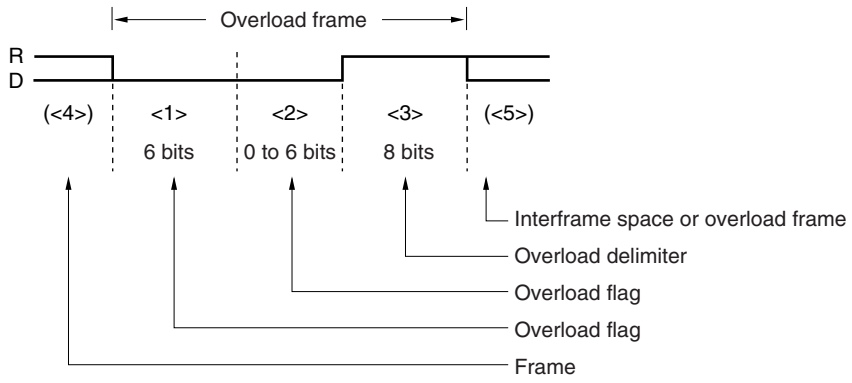
16.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation[Note]
- If a dominant level is detected at the first two bits during intermission
- If a dominant level is detected at the last bit (7th bit) of the end of frame or at the last bit (8th bit) of the error delimiter/overload delimiter

Note The CAN is internally fast enough to process all received frames not generating overload frames.

Figure 16-16. Overload Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 16-8. Definition of Overload Frame Fields

No	Name	Bit Count	Definition
<1>	Overload flag	6	Outputs 6 dominant-level bits consecutively.
<2>	Overload flag from other node	0 to 6	The node that received an overload flag in the interframe space outputs an overload flag.
<3>	Overload delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Frame	–	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

16.3 Functions

16.3.1 Determining bus priority

(1) When a node starts transmission:

- During bus idle, the node that output data first transmits the data.

(2) When more than one node starts transmission:

- The node that outputs the dominant level for the longest consecutively from the first bit of the arbitration field acquires the bus priority (if a dominant level and a recessive level are simultaneously transmitted, the dominant level is taken as the bus value).
- The transmitting node compares its output arbitration field and the data level on the bus.

Table 16-9. Determining Bus Priority

Level match	Continuous transmission
Level mismatch	Continuous transmission

(3) Priority of data frame and remote frame

- When a data frame and a remote frame are on the bus, the data frame has priority because its RTR bit, the last bit in the arbitration field, carries a dominant level.

Remark If the extended-format data frame and the standard-format remote frame conflict on the bus (if ID28 to ID18 of both of them are the same), the standard-format remote frames takes priority.

16.3.2 Bit stuffing

Bit stuffing is used to establish synchronization by appending 1-bit inverted data if the same level continues for 5 bits, in order to prevent a burst error.

Table 16-10. Bit Stuffing

Transmission	During the transmission of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, 1 inverted-level bit of data is inserted before the following bit.
Reception	During the reception of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, reception is continued after deleting the next bit.

16.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

16.3.4 Multi cast

Although there is one transmitting node, two or more nodes can receive the same data at the same time because the same identifier can be set to two or more nodes.

16.3.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function puts the CAN controller in waiting mode to achieve low power consumption.

The controller is woken up from the CAN sleep mode by bus operation but it is not woken up from the CAN stop mode by bus operation (the CAN stop mode is controlled by CPU access).

16.3.6 Error control function

(1) Error types

Table 16-11. Error Types

Type	Description of Error		Detection State	
	Detection Method	Detection Condition	Transmission/ Reception	Field/Frame
Bit error	Comparison of output level and level on the bus	Mismatch of levels	Transmitting/ receiving node	Bit that outputting data on the bus at the start of frame to end of frame, error frame and overload frame.
Stuff error	Check the receive data at the stuff bit	6 consecutive bits of the same output level	Receiving node	Start of frame to CRC sequence
CRC error	Comparison of the CRC sequence generated from the receive data and the received CRC sequence	Mismatch of CRC	Receiving node	CRC field
Form error	Field/frame check of the fixed format	Detection of fixed format violation	Receiving node	CRC delimiter ACK field End of frame Error frame Overload frame
ACK error	Check of the ACK slot by the transmitting node	Detection of recessive level in ACK slot	Transmitting node	ACK slot

(2) Output timing of error frame

Table 16-12. Output Timing of Error Frame

Type	Output Timing
Bit error, stuff error, form error, ACK error	Error frame output is started at the timing of the bit following the detected error.
CRC error	Error frame output is started at the timing of the bit following the ACK delimiter.

(3) Processing in case of error

The transmission node re-transmits the data frame or remote frame after the error frame (However, it does not re-transmit the frame in the single-shot mode.).

(4) Error state

(a) Types of error states

The following three types of error states are defined by the CAN specification.

- Error active
- Error passive
- Bus-off

These types of error states are classified by the values of the TEC7 to TEC0 bits (transmission error counter bits) and the REC6 to REC0 bits (reception error counter bits) of the CAN error counter register (C0ERC) as shown in Table 16-13.
The present error state is indicated by the CAN module information register (C0INFO).
When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the C0INFO register is set to 1. In this case, the bus state must be tested because it is considered that the bus has a serious fault. An error counter value of 128 or more indicates an error passive state and the TECS1 or RECS1 bit of the C0INFO register is set to 1.

- If the value of the transmission error counter is greater than or equal to 256 (actually, the transmission error counter does not indicate a value greater than or equal to 256), the bus-off state is reached and the BOFF bit of the C0INFO register is set to 1.
- If only one node is active on the bus at startup (i.e., a particular case such as when the bus is connected only to the local station), ACK is not returned even if data is transmitted. Consequently, re-transmission of the error frame and data is repeated. In the error passive state, however, the transmission error counter is not incremented and the bus-off state is not reached.

Table 16-13. Types of Error States

Type	Operation	Value of Error Counter	Indication of C0INFO Register	Operation specific to Given Error State
Error active	Transmission	0-95	TECS1, TECS0 = 00	- Outputs an active error flag (6 consecutive dominant-level bits) on detection of the error.
	Reception	0-95	RECS1, RECS0 = 00	
	Transmission	96-127	TECS1, TECS0 = 01	
	Reception	96-127	RECS1, RECS0 = 01	
Error passive	Transmission	128-255	TECS1, TECS0 = 11	- Outputs a passive error flag (6 consecutive recessive-level bits) on detection of the error. - Transmits 8 recessive-level bits, in between transmissions, following an intermission (suspend transmission).
	Reception	128 or more	RECS1, RECS0 = 11	
Bus-off	Transmission	256 or more (not indicated)[Note]	BOFF = 1, TECS1, TECS0 = 11	- Communication is not possible. Messages are not stored when receiving frames, however, the following operations of <1>, <2>, and <3> are done. <1> TSOUT toggles. <2> REC is incremented/decremented. <3> VALID bit is set. - If the CAN module is entered to the initialization mode and then transition request to any operation mode is made, and when 11 consecutive recessive-level bits are detected 128 times, the error counter is reset to 0 and the error active state can be restored.

Note The value of the transmission error counter (TEC) is invalid when the BOFF bit is set to 1. If an error that increments the value of the transmission error counter by +8 while the counter value is in a range of 248 to 255, the counter is not incremented and the bus-off state is assumed.

(b) Error counter

The error counter counts up when an error has occurred, and counts down upon successful transmission and reception. The error counter is updated immediately after error detection.

Table 16-14. Error Counter

State	Transmission Error Counter (TEC7 to TEC0)	Reception Error Counter (REC6 to REC0)
Receiving node detects an error (except bit error in the active error flag or overload flag).	No change	+1 (when REPS bit = 0)
Receiving node detects dominant level following error flag of error frame.	No change	+8 (when REPS bit = 0)
Transmitting node transmits an error flag. [As exceptions, the error counter does not change in the following cases.] <1> ACK error is detected in error passive state and dominant level is not detected while the passive error flag is being output. <2> A stuff error is detected in an arbitration field that transmitted a recessive level as a stuff bit, but a dominant level is detected.	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active transmitting node)	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active receiving node)	No change	+8 (when REPS bit = 0)
When the node detects 14 consecutive dominant-level bits from the beginning of the active error flag or overload flag, and then subsequently detects 8 consecutive dominant-level bits. When the node detects 8 consecutive dominant levels after a passive error flag	+8 (during transmission)	+8 (during reception, when REPS bit = 0)
When the transmitting node has completed transmission without error (±0 if error counter = 0)	–1	No change
When the receiving node has completed reception without error	No change	- –1 (1 ≤ REC6 to REC0 ≤ 127, when REPS bit = 0) - ±0 (REC6 to REC0 = 0, when REPS bit = 0) - Value of 119 to 255 is set (when REPS bit = 1)

(c) Occurrence of bit error in intermission

An overload frame is generated.

Caution If an error occurs, the error flag output (active or passive) is controlled according to the contents of the transmission error counter and reception error counter before the error occurred. The value of the error counter is incremented after the error flag has been output.

(5) Recovery from bus-off state

When the CAN module is in the bus-off state, the CAN module permanently sets its output signals (CTxD) to recessive level.

The CAN module recovers from the bus-off state in the following bus-off recovery sequence.

<1> A request to enter the CAN initialization mode

<2> A request to enter a CAN operation mode

(a) Recovery operation through normal recovery sequence

(b) Forced recovery operation that skips recovery sequence

(a) Recovery operation from bus-off state through normal recovery sequence

The CAN module first issues a request to enter the initialization mode (refer to timing <1> in Figure 16-17). This request will be immediately acknowledged, and the OPMODE bits of the C0CTRL register are cleared to 000B. Processing such as analyzing the fault that has caused the bus-off state, re-defining the CAN module and message buffer using application software, or stopping the operation of the CAN module can be performed by clearing the GOM bit to 0.

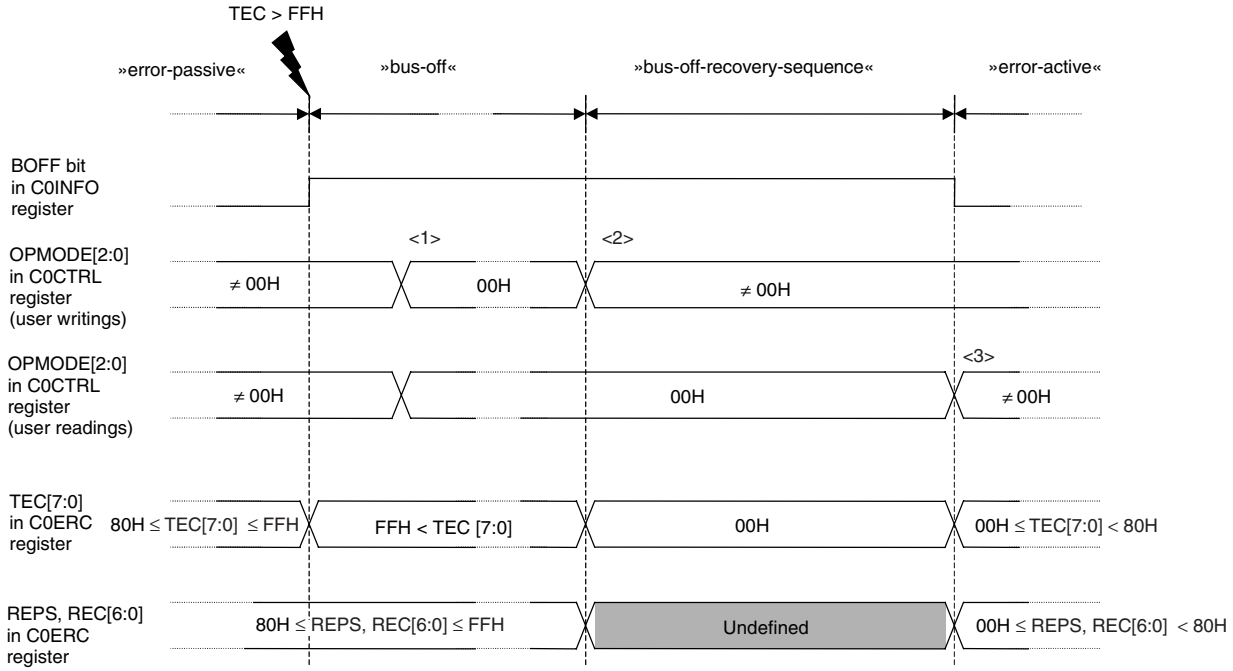
Next, the user requests to change the mode from the initialization mode to an operation mode (refer to timing <2> in Figure 16-17). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the CAN protocol ISO 11898, and it is necessary to detect 11 consecutive recessive-level bits 128 times. At this time, the request to change the mode to an operation mode is held pending until the recovery conditions are satisfied. When the recovery conditions are satisfied (refer to timing <3> in Figure 16-17), the CAN module can enter the operation mode it has requested. Until the CAN module enters this operation mode, it stays in the initialization mode. Completion to be requested operation mode can be confirmed by reading the OPMODE bits of the C0CTRL register.

During the bus-off period and bus-off recovery sequence, the BOFF bit of the C0INFO register stays set (to 1). In the bus-off recovery sequence, the reception error counter (REC[6:0]) counts the number of times 11 consecutive recessive-level bits have been detected on the bus. Therefore, the recovery state can be checked by reading REC[6:0].

Cautions 1 When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.
Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

2. In the bus-off recovery sequence, REC[6:0] counts up (+1) each time 11 consecutive recessive-level bits have been detected. Even during the bus-off period, the CAN module can enter the CAN sleep mode or CAN stop mode. To start the bus-off recovery sequence, it is necessary to transit to the initialization mode once. However, when the CAN module is in either CAN sleep mode or CAN stop mode, transition request to the initialization mode is not accepted, thus you have to release the CAN sleep mode first. In this case, as soon as the CAN sleep mode is released, the bus-off recovery sequence starts and no transition to initialization mode is necessary. If the can module detects a dominant edge on the CAN bus while in sleep mode even during bus-off, the sleep mode will be left and the bus-off recovery sequence will start.

Figure 16-17. Recovery Operation from Bus-off State through Normal Recovery Sequence



(b) Forced recovery operation that skips bus-off recovery sequence

The CAN module can be forcibly released from the bus-off state, regardless of the bus state, by skipping the bus-off recovery sequence. Here is the procedure.

First, the CAN module requests to enter the initialization mode. For the operation and points to be noted at this time, refer to (a) Recovery operation from bus-off state through normal recovery sequence.

Next, the module requests to enter an operation mode. At the same time, the CCERC bit of the C0CTRL register must be set to 1.

As a result, the bus-off recovery sequence defined by the CAN protocol ISO 11898 is skipped, and the module immediately enters the operation mode. In this case, the module is connected to the CAN bus after it has monitored 11 consecutive recessive-level bits. For details, refer to the processing in Figure 16-56.

Caution This function is not defined by the CAN protocol ISO 11898. When using this function, thoroughly evaluate its effect on the network system.

(6) Initializing CAN module error counter register (C0ERC) in initialization mode

If it is necessary to initialize the CAN module error counter register (C0ERC) and CAN module information register (C0INFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the C0CTRL register in the initialization mode. When initialization has been completed, the CCERC bit is automatically cleared to 0.

Cautions 1. This function is enabled only in the initialization mode. Even if the CCERC bit is set to 1 in a CAN operation mode, the C0ERC and C0INFO registers are not initialized.

2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.

16.3.7 Baud rate control function

(1) Prescaler

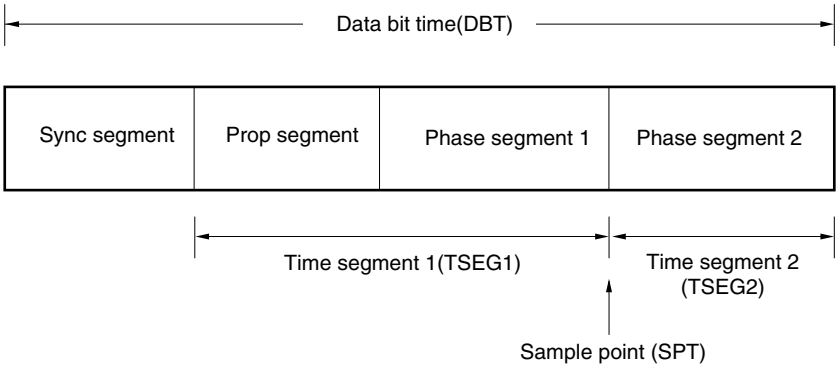
The CAN controller has a prescaler that divides the clock (f_{CAN}) supplied to CAN. This prescaler generates a CAN protocol layer basic clock (f_{TQ}) derived from the CAN module system clock (f_{CANMOD}), and divided by 1 to 256 (refer to **16.6 (12) CAN Bit Rate Prescaler Register (C0BRP)**).

(2) Data bit time (8-25 time quanta)

One data bit time is defined as shown in Figure 16-18.

The CAN controller sets time segment 1, time segment 2, and reSynchronization Jump Width (SJW) as the parameter of data bit time, as shown in Figure 16-18. Time segment 1 is equivalent to the total of the propagation (prop) segment and phase segment 1 that are defined by the CAN protocol specification. Time segment 2 is equivalent to phase segment 2.

Figure 16-18. Segment Setting

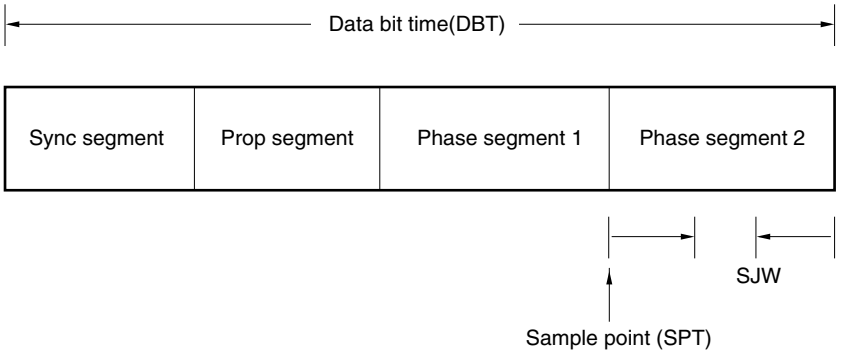


Segment Name	Settable Range	Notes on Setting to Confirm to CAN Specification
Time Segment 1 (TSEG1)	2TQ-16TQ	—
Time Segment 2 (TSEG2)	1TQ-8TQ	IPT of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length equal to phase segment 1 must be set here. This means that the length of time segment 1 minus 1TQ is the settable upper limit of time segment 2.
Resynchronization jump width(SJW)	1TQ-4TQ	The length of time segment 1 minus 1TQ or 4 TQ, whichever is smaller.

Remark IPT : Information Processing Time
TQ : Time Quanta

Reference: The CAN standard ISO 11898 specification defines the segments constituting the data bit time as shown in Figure 16-19.

Figure 16-19. Reference: Configuration of Data Bit Time Defined by CAN Specification



Segment Name	Segment Length	Description
Sync Segment (Synchronization Segment)	1	This segment starts at the edge where the level changes from recessive to dominant when hard-synchronization is established.
Prop Segment	Programmable to 1 to 8 or more	This segment absorbs the delay of the output buffer, CAN bus, and input buffer. The length of this segment is set so that ACK is returned before the start of phase segment 1. Time of prop segment ≥ (Delay of output buffer) + 2 x (Delay of CAN bus) + (Delay of input buffer)
Phase Segment 1	Programmable to 1 to 8	This segment compensates for an error of data bit time. The longer this segment, the wider the permissible range but the slower the communication speed.
Phase Segment 2	Phase Segment 1 or IPT, whichever greater	
SJW	Programmable from 1TQ to length of segment 1 or 4TQ, whichever is smaller	This width sets the upper limit of expansion or contraction of the phase segment during resynchronization.

Remark IPT : Information Processing Time
TQ : Time Quanta

(3) Synchronizing data bit

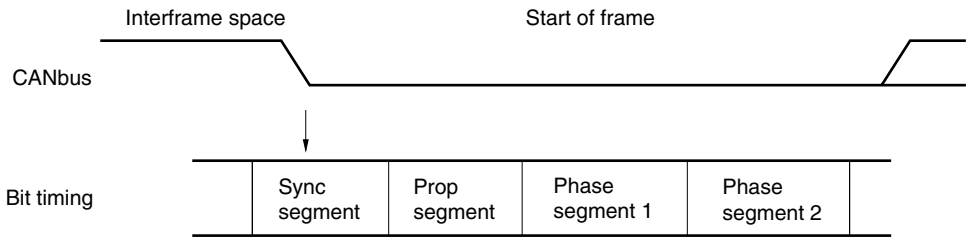
- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

(a) Hard-synchronization

This synchronization is established when the receiving node detects the start of frame in the interframe space.

- When a falling edge is detected on the bus, that TQ means the sync segment and the next segment is the prop segment. In this case, synchronization is established regardless of SJW.

Figure 16-20. Hard-synchronization at Recognition of Dominant Level during Bus Idle

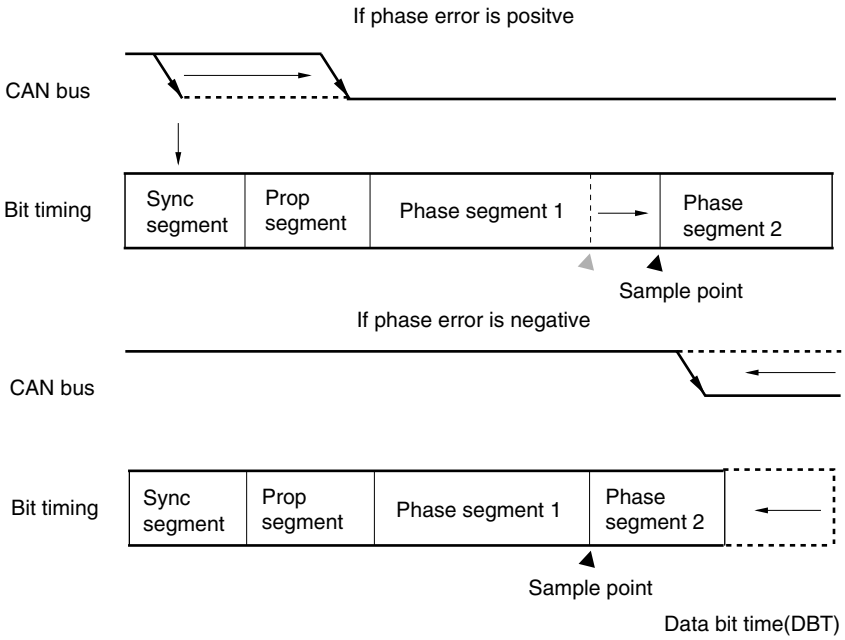


(b) Resynchronization

Synchronization is established again if a level change is detected on the bus during reception (only if a recessive level was sampled previously).

- The phase error of the edge is given by the relative position of the detected edge and sync segment.
  <Sign of phase error>
  0: If the edge is within the sync segment
  Positive: If the edge is before the sample point (phase error)
  Negative: If the edge is after the sample point (phase error)
  If phase error is positive: Phase segment 1 is longer by specified SJW.
  If phase error is negative: Phase segment 2 is shorter by specified SJW.
- The sample point of the data of the receiving node moves relatively due to the “discrepancy” in baud rate between the transmitting node and receiving node.

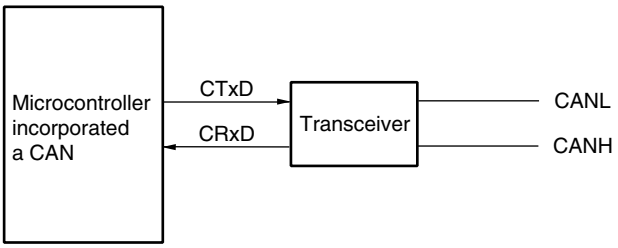
Figure 16-21. Resynchronization



16.4 Connection with Target System

The microcontroller incorporated a CAN has to be connected to the CAN bus using an external transceiver.

Figure 16-22. Connection to CAN Bus



16.5 Internal Registers of CAN Controller

16.5.1 CAN controller configuration

Table 16-15. List of CAN Controller Registers

Item	Register Name
CAN global registers	CAN global control register (C0GMCTRL)
	CAN global clock selection register (C0GMCS)
	CAN global automatic block transmission control register (C0GMABT)
	CAN global automatic block transmission delay register (C0GMABTD)
CAN module registers	CAN module mask 1 register (C0MASK1L, C0MASK1H)
	CAN module mask 2 register (C0MASK2L, C0MASK2H)
	CAN module mask3 register (C0MASK3L, C0MASK3H)
	CAN module mask 4 registers (C0MASK4L, C0MASK4H)
	CAN module control register (C0CTRL)
	CAN module last error code register (C0LEC)
	CAN module information register (C0INFO)
	CAN module error counter register (C0ERC)
	CAN module interrupt enable register (C0IE)
	CAN module interrupt status register (C0INTS)
	CAN module bit rate prescaler register (C0BRP)
	CAN module bit rate register (C0BTR)
	CAN module last in-pointer register (C0LIPT)
	CAN module receive history list register (C0RGPT)
	CAN module last out-pointer register (C0LOPT)
	CAN module transmit history list register (C0TGPT)
	CAN module time stamp register (C0TS)
Message buffer registers	CAN message data byte 01 register m (C0MDATA01m)
	CAN message data byte 0 register m (C0MDATA0m)
	CAN message data byte 1 register m (C0MDATA1m)
	CAN message data byte 23 register m (C0MDATA23m)
	CAN message data byte 2 register m (C0MDATA2m)
	CAN message data byte 3 Register m (C0MDATA3m)
	CAN message data byte 45 Register m (C0MDATA45m)
	CAN message data byte 4 Register m (C0MDATA4m)
	CAN message data byte 5 Register m (C0MDATA5m)
	CAN message data byte 67 Register m (C0MDATA67m)
	CAN message data byte 6 register m (C0MDATA6m)
	CAN message data byte 7 register m (C0MDATA7m)
	CAN message data length register m (C0MDLCm)
	CAN message configuration register m (C0MCONFm)
	CAN message ID register m (C0MIDLm, C0MIDHm)
	CAN message control register m (C0MCTRLm)

Remark m = 0 to 15

16.5.2 Register access type

Table 16-16. Register Access Types (1/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA00H	CAN0 message data byte 01 register 00	C0MDATA0100	R/W			√	Undefined
FA00H	CAN0 message data byte 0 register 00	C0MDATA000			√		Undefined
FA01H	CAN0 message data byte 1 register 00	C0MDATA100			√		Undefined
FA02H	CAN0 message data byte 23 register 00	C0MDATA2300				√	Undefined
FA02H	CAN0 message data byte 2 register 00	C0MDATA200			√		Undefined
FA03H	CAN0 message data byte 3 register 00	C0MDATA300			√		Undefined
FA04H	CAN0 message data byte 45 register 00	C0MDATA4500				√	Undefined
FA04H	CAN0 message data byte 4 register 00	C0MDATA400			√		Undefined
FA05H	CAN0 message data byte 5 register 00	C0MDATA500			√		Undefined
FA06H	CAN0 message data byte 67 register 00	C0MDATA6700				√	Undefined
FA06H	CAN0 message data byte 6 register 00	C0MDATA600			√		Undefined
FA07H	CAN0 message data byte 7 register 00	C0MDATA700			√		Undefined
FA08H	CAN0 message data length code register 00	C0MDLC00			√		0000xxxxB
FA09H	CAN0 message configuration register 00	C0MCONF00			√		Undefined
FA0AH	CAN0 message ID register 00	C0MIDL00				√	Undefined
FA0CH		C0MIDH00				√	Undefined
FA0EH	CAN0 message control register 00	C0MCTRL00				√	00x00000 000xx000B
FA10H	CAN0 message data byte 01 register 01	C0MDATA0101				√	Undefined
FA10H	CAN0 message data byte 0 register 01	C0MDATA001			√		Undefined
FA11H	CAN0 message data byte 1 register 01	C0MDATA101			√		Undefined
FA12H	CAN0 message data byte 23 register 01	C0MDATA2301				√	Undefined
FA12H	CAN0 message data byte 2 register 01	C0MDATA201			√		Undefined
FA13H	CAN0 message data byte 3 register 01	C0MDATA301			√		Undefined
FA14H	CAN0 message data byte 45 register 01	C0MDATA4501				√	Undefined
FA14H	CAN0 message data byte 4 register 01	C0MDATA401			√		Undefined
FA15H	CAN0 message data byte 5 register 01	C0MDATA501			√		Undefined
FA16H	CAN0 message data byte 67 register 01	C0MDATA6701				√	Undefined
FA16H	CAN0 message data byte 6 register 01	C0MDATA601			√		Undefined
FA17H	CAN0 message data byte 7 register 01	C0MDATA701			√		Undefined
FA18H	CAN0 message data length code register 01	C0MDLC01			√		0000xxxxB
FA19H	CAN0 message configuration register 01	C0MCONF01			√		Undefined
FA1AH	CAN0 message ID register 01	C0MIDL01				√	Undefined
FA1CH		C0MIDH01				√	Undefined
FA1EH	CAN0 message control register 01	C0MCTRL01				√	00x00000 000xx000B

Table 16-16. Register Access Types (2/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA20H	CAN0 message data byte 01 register 02	C0MDATA0102	R/W			√	Undefined
FA20H	CAN0 message data byte 0 register 02	C0MDATA002			√		Undefined
FA21H	CAN0 message data byte 1 register 02	C0MDATA102			√		Undefined
FA22H	CAN0 message data byte 23 register 02	C0MDATA2302				√	Undefined
FA22H	CAN0 message data byte 2 register 02	C0MDATA202			√		Undefined
FA23H	CAN0 message data byte 3 register 02	C0MDATA302			√		Undefined
FA24H	CAN0 message data byte 45 register 02	C0MDATA4502				√	Undefined
FA24H	CAN0 message data byte 4 register 02	C0MDATA402			√		Undefined
FA25H	CAN0 message data byte 5 register 02	C0MDATA502			√		Undefined
FA26H	CAN0 message data byte 67 register 02	C0MDATA6702				√	Undefined
FA26H	CAN0 message data byte 6 register 02	C0MDATA602			√		Undefined
FA27H	CAN0 message data byte 7 register 02	C0MDATA702			√		Undefined
FA28H	CAN0 message data length code register 02	C0MDLC02			√		0000xxxxB
FA29H	CAN0 message configuration register 02	C0MCONF02			√		Undefined
FA2AH	CAN0 message ID register 02	C0MIDL02				√	Undefined
FA2CH		C0MIDH02				√	Undefined
FA2EH	CAN0 message control register 02	C0MCTRL02				√	00x00000 000xx000B
FA30H	CAN0 message data byte 01 register 03	C0MDATA0103				√	Undefined
FA30H	CAN0 message data byte 0 register 03	C0MDATA003			√		Undefined
FA31H	CAN0 message data byte 1 register 03	C0MDATA103			√		Undefined
FA32H	CAN0 message data byte 23 register 03	C0MDATA2303				√	Undefined
FA32H	CAN0 message data byte 2 register 03	C0MDATA203			√		Undefined
FA33H	CAN0 message data byte 3 register 03	C0MDATA303			√		Undefined
FA34H	CAN0 message data byte 45 register 03	C0MDATA4503				√	Undefined
FA34H	CAN0 message data byte 4 register 03	C0MDATA403			√		Undefined
FA35H	CAN0 message data byte 5 register 03	C0MDATA503			√		Undefined
FA36H	CAN0 message data byte 67 register 03	C0MDATA6703				√	Undefined
FA36H	CAN0 message data byte 6 register 03	C0MDATA603			√		Undefined
FA37H	CAN0 message data byte 7 register 03	C0MDATA703			√		Undefined
FA38H	CAN0 message data length code register 03	C0MDLC03			√		0000xxxxB
FA39H	CAN0 message configuration register 03	C0MCONF03			√		Undefined
FA3AH	CAN0 message ID register 03	C0MIDL03				√	Undefined
FA3CH		C0MIDH03				√	Undefined
FA3EH	CAN0 message control register 03	C0MCTRL03				√	00x00000 000xx000B

Table 16-16. Register Access Types (3/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA40H	CAN0 message data byte 01 register 04	C0MDATA0104	R/W			√	Undefined
FA40H	CAN0 message data byte 0 register 04	C0MDATA004			√		Undefined
FA41H	CAN0 message data byte 1 register 04	C0MDATA104			√		Undefined
FA42H	CAN0 message data byte 23 register 04	C0MDATA2304				√	Undefined
FA42H	CAN0 message data byte 2 register 04	C0MDATA204			√		Undefined
FA43H	CAN0 message data byte 3 register 04	C0MDATA304			√		Undefined
FA44H	CAN0 message data byte 45 register 04	C0MDATA4504				√	Undefined
FA44H	CAN0 message data byte 4 register 04	C0MDATA404			√		Undefined
FA45H	CAN0 message data byte 5 register 04	C0MDATA504			√		Undefined
FA46H	CAN0 message data byte 67 register 04	C0MDATA6704				√	Undefined
FA46H	CAN0 message data byte 6 register 04	C0MDATA604			√		Undefined
FA47H	CAN0 message data byte 7 register 04	C0MDATA704			√		Undefined
FA48H	CAN0 message data length code register 04	C0MDLC04			√		0000xxxxB
FA49H	CAN0 message configuration register 04	C0MCONF04			√		Undefined
FA4AH	CAN0 message ID register 04	C0MIDL04				√	Undefined
FA4CH		C0MIDH04				√	Undefined
FA4EH	CAN0 message control register 04	C0MCTRL04				√	00x00000 000xx000B
FA50H	CAN0 message data byte 01 register 05	C0MDATA0105				√	Undefined
FA50H	CAN0 message data byte 0 register 05	C0MDATA005			√		Undefined
FA51H	CAN0 message data byte 1 register 05	C0MDATA105			√		Undefined
FA52H	CAN0 message data byte 23 register 05	C0MDATA2305				√	Undefined
FA52H	CAN0 message data byte 2 register 05	C0MDATA205			√		Undefined
FA53H	CAN0 message data byte 3 register 05	C0MDATA305			√		Undefined
FA54H	CAN0 message data byte 45 register 05	C0MDATA4505				√	Undefined
FA54H	CAN0 message data byte 4 register 05	C0MDATA405			√		Undefined
FA55H	CAN0 message data byte 5 register 05	C0MDATA505			√		Undefined
FA56H	CAN0 message data byte 67 register 05	C0MDATA6705				√	Undefined
FA56H	CAN0 message data byte 6 register 05	C0MDATA605			√		Undefined
FA57H	CAN0 message data byte 7 register 05	C0MDATA705			√		Undefined
FA58H	CAN0 message data length code register 05	C0MDLC05			√		0000xxxxB
FA59H	CAN0 message configuration register 05	C0MCONF05			√		Undefined
FA5AH	CAN0 message ID register 05	C0MIDL05				√	Undefined
FA5CH		C0MIDH05				√	Undefined
FA5EH	CAN0 message configuration register 05	C0MCTRL05				√	00x00000 000xx000B

Table 16-16. Register Access Types (4/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA60H	CAN0 message data byte 01 register 06	C0MDATA0106	R/W			√	Undefined
FA60H	CAN0 message data byte 0 register 06	C0MDATA006			√		Undefined
FA61H	CAN0 message data byte 1 register 06	C0MDATA106			√		Undefined
FA62H	CAN0 message data byte 23 register 06	C0MDATA2306				√	Undefined
FA62H	CAN0 message data byte 2 register 06	C0MDATA206			√		Undefined
FA63H	CAN0 message data byte 3 register 06	C0MDATA306			√		Undefined
FA64H	CAN0 message data byte 45 register 06	C0MDATA4506				√	Undefined
FA64H	CAN0 message data byte 4 register 06	C0MDATA406			√		Undefined
FA65H	CAN0 message data byte 5 register 06	C0MDATA506			√		Undefined
FA66H	CAN0 message data byte 67 register 06	C0MDATA6706				√	Undefined
FA66H	CAN0 message data byte 6 register 06	C0MDATA606			√		Undefined
FA67H	CAN0 message data byte 7 register 06	C0MDATA706			√		Undefined
FA68H	CAN0 message data length code register 06	C0MDLC06			√		0000xxxxB
FA69H	CAN0 message configuration register 06	C0MCONF06			√		Undefined
FA6AH	CAN0 message ID register 06	C0MIDL06				√	Undefined
FA6CH		C0MIDH06				√	Undefined
FA6EH	CAN0 message control register 06	C0MCTRL06				√	00x00000 000xx000B
FA70H	CAN0 message data byte 01 register 07	C0MDATA0107				√	Undefined
FA70H	CAN0 message data byte 0 register 07	C0MDATA007			√		Undefined
FA71H	CAN0 message data byte 1 register 07	C0MDATA107			√		Undefined
FA72H	CAN0 message data byte 23 register 07	C0MDATA2307				√	Undefined
FA72H	CAN0 message data byte 2 register 07	C0MDATA207			√		Undefined
FA73H	CAN0 message data byte 3 register 07	C0MDATA307			√		Undefined
FA74H	CAN0 message data byte 45 register 07	C0MDATA4507				√	Undefined
FA74H	CAN0 message data byte 4 register 07	C0MDATA407			√		Undefined
FA75H	CAN0 message data byte 5 register 07	C0MDATA507			√		Undefined
FA76H	CAN0 message data byte 67 register 07	C0MDATA6707				√	Undefined
FA76H	CAN0 message data byte 6 register 07	C0MDATA607			√		Undefined
FA77H	CAN0 message data byte 7 register 07	C0MDATA707			√		Undefined
FA78H	CAN0 message data length code register 07	C0MDLC07			√		0000xxxxB
FA79H	CAN0 message configuration register 07	C0MCONF07			√		Undefined
FA7AH	CAN0 message ID register 07	C0MIDL07				√	Undefined
FA7CH		C0MIDH07				√	Undefined
FA7EH	CAN0 message control register 07	C0MCTRL07				√	00x00000 000xx000B

Table 16-16. Register Access Types (5/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA80H	CAN0 message data byte 01 register 08	C0MDATA0108	R/W			√	Undefined
FA80H	CAN0 message data byte 0 register 08	C0MDATA008			√		Undefined
FA81H	CAN0 message data byte 1 register 08	C0MDATA108			√		Undefined
FA82H	CAN0 message data byte 23 register 08	C0MDATA2308				√	Undefined
FA82H	CAN0 message data byte 2 register 08	C0MDATA208			√		Undefined
FA83H	CAN0 message data byte 3 register 08	C0MDATA308			√		Undefined
FA84H	CAN0 message data byte 45 register 08	C0MDATA4508				√	Undefined
FA84H	CAN0 message data byte 4 register 08	C0MDATA408			√		Undefined
FA85H	CAN0 message data byte 5 register 08	C0MDATA508			√		Undefined
FA86H	CAN0 message data byte 67 register 08	C0MDATA6708				√	Undefined
FA86H	CAN0 message data byte 6 register 08	C0MDATA608			√		Undefined
FA87H	CAN0 message data byte 7 register 08	C0MDATA708			√		Undefined
FA88H	CAN0 message data length code register 08	C0MDLC08			√		0000xxxxB
FA89H	CAN0 message configuration register 08	C0MCONF08			√		Undefined
FA8AH	CAN0 message ID register 08	C0MIDL08				√	Undefined
FA8CH		C0MIDH08				√	Undefined
FA8EH	CAN0 message control register 08	C0MCTRL08				√	00x00000 000xx000B
FA90H	CAN0 message data byte 01 register 09	C0MDATA0109				√	Undefined
FA90H	CAN0 message data byte 0 register 09	C0MDATA009			√		Undefined
FA91H	CAN0 message data byte 1 register 09	C0MDATA109			√		Undefined
FA92H	CAN0 message data byte 23 register 09	C0MDATA2309				√	Undefined
FA92H	CAN0 message data byte 2 register 09	C0MDATA209			√		Undefined
FA93H	CAN0 message data byte 3 register 09	C0MDATA309			√		Undefined
FA94H	CAN0 message data byte 45 register 09	C0MDATA4509				√	Undefined
FA94H	CAN0 message data byte 4 register 09	C0MDATA409			√		Undefined
FA95H	CAN0 message data byte 5 register 09	C0MDATA509			√		Undefined
FA96H	CAN0 message data byte 67 register 09	C0MDATA6709				√	Undefined
FA96H	CAN0 message data byte 6 register 09	C0MDATA609			√		Undefined
FA97H	CAN0 message data byte 7 register 09	C0MDATA709			√		Undefined
FA98H	CAN0 message data length code register 09	C0MDLC09			√		0000xxxxB
FA99H	CAN0 message configuration register 09	C0MCONF09			√		Undefined
FA9AH	CAN0 message ID register 09	C0MIDL09				√	Undefined
FA9CH		C0MIDH09				√	Undefined
FA9EH	CAN0 message control register 09	C0MCTRL09				√	00x00000 000xx000B

Table 16-16. Register Access Types (6/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FAA0H	CAN0 message data byte 01 register 10	C0MDATA0110	R/W			√	Undefined
FAA0H	CAN0 message data byte 0 register 10	C0MDATA010			√		Undefined
FAA1H	CAN0 message data byte 1 register 10	C0MDATA110			√		Undefined
FAA2H	CAN0 message data byte 23 register 10	C0MDATA2310				√	Undefined
FAA2H	CAN0 message data byte 2 register 10	C0MDATA210			√		Undefined
FAA3H	CAN0 message data byte 3 register 10	C0MDATA310			√		Undefined
FAA4H	CAN0 message data byte 45 register 10	C0MDATA4510				√	Undefined
FAA4H	CAN0 message data byte 4 register 10	C0MDATA410			√		Undefined
FAA5H	CAN0 message data byte 5 register 10	C0MDATA510			√		Undefined
FAA6H	CAN0 message data byte 67 register 10	C0MDATA6710				√	Undefined
FAA6H	CAN0 message data byte 6 register 10	C0MDATA610			√		Undefined
FAA7H	CAN0 message data byte 7 register 10	C0MDATA710			√		Undefined
FAA8H	CAN0 message data length code register 10	C0MDLC10			√		0000xxxxB
FAA9H	CAN0 message configuration register 10	C0MCONF10			√		Undefined
FAAAH	CAN0 message ID register 10	C0MIDL10				√	Undefined
FAACH		C0MIDH10				√	Undefined
FAAEH	CAN0 message control register 10	C0MCTRL10				√	00x00000 000xx000B
FAB0H	CAN0 message data byte 01 register 11	C0MDATA0111				√	Undefined
FAB0H	CAN0 message data byte 0 register 11	C0MDATA011			√		Undefined
FAB1H	CAN0 message data byte 1 register 11	C0MDATA111			√		Undefined
FAB2H	CAN0 message data byte 23 register 11	C0MDATA2311				√	Undefined
FAB2H	CAN0 message data byte 2 register 11	C0MDATA211			√		Undefined
FAB3H	CAN0 message data byte 3 register 11	C0MDATA311			√		Undefined
FAB4H	CAN0 message data byte 45 register 11	C0MDATA4511				√	Undefined
FAB4H	CAN0 message data byte 4 register 11	C0MDATA411			√		Undefined
FAB5H	CAN0 message data byte 51 register 11	C0MDATA511			√		Undefined
FAB6H	CAN0 message data byte 67 register 11	C0MDATA6711				√	Undefined
FAB6H	CAN0 message data byte 6 register 11	C0MDATA611			√		Undefined
FAB7H	CAN0 message data byte 71 register 11	C0MDATA711			√		Undefined
FAB8H	CAN0 message data length code register 11	C0MDLC11			√		0000xxxxB
FAB9H	CAN0 message configuration register 11	C0MCONF11			√		Undefined
FABAH	CAN0 message ID register 11	C0MIDL11				√	Undefined
FABCH		C0MIDH11				√	Undefined
FABEH	CAN0 message control register 11	C0MCTRL11				√	00x00000 000xx000B

Table 16-16. Register Access Types (7/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FAC0H	CAN0 message data byte 01 register 12	C0MDATA0112	R/W			√	Undefined
FAC0H	CAN0 message data byte 0 register 12	C0MDATA012			√		Undefined
FAC1H	CAN0 message data byte 1 register 12	C0MDATA112			√		Undefined
FAC2H	CAN0 message data byte 23 register 12	C0MDATA2312				√	Undefined
FAC2H	CAN0 message data byte 2 register 12	C0MDATA212			√		Undefined
FAC3H	CAN0 message data byte 3 register 12	C0MDATA312			√		Undefined
FAC4H	CAN0 message data byte 45 register 12	C0MDATA4512				√	Undefined
FAC4H	CAN0 message data byte 4 register 12	C0MDATA412			√		Undefined
FAC5H	CAN0 message data byte 5 register 12	C0MDATA512			√		Undefined
FAC6H	CAN0 message data byte 67 register 12	C0MDATA6712				√	Undefined
FAC6H	CAN0 message data byte 6 register 12	C0MDATA612			√		Undefined
FAC7H	CAN0 message data byte 7 register 12	C0MDATA712			√		Undefined
FAC8H	CAN0 message data length code register 12	C0MDLC12			√		0000xxxxB
FAC9H	CAN0 message configuration register 12	C0MCONF12			√		Undefined
FACAH	CAN0 message ID register 12	C0MIDL12				√	Undefined
FACCH		C0MIDH12				√	Undefined
FACEH	CAN0 message control register 12	C0MCTRL12				√	00x00000 000xx000B
FAD0H	CAN0 message data byte 01 register 13	C0MDATA0113				√	Undefined
FAD0H	CAN0 message data byte 0 register 13	C0MDATA013			√		Undefined
FAD1H	CAN0 message data byte 1 register 13	C0MDATA113			√		Undefined
FAD2H	CAN0 message data byte 23 register 13	C0MDATA2313				√	Undefined
FAD2H	CAN0 message data byte 2 register 13	C0MDATA213			√		Undefined
FAD3H	CAN0 message data byte 3 register 13	C0MDATA313			√		Undefined
FAD4H	CAN0 message data byte 45 register 13	C0MDATA4513				√	Undefined
FAD4H	CAN0 message data byte 4 register 13	C0MDATA413			√		Undefined
FAD5H	CAN0 message data byte 5 register 13	C0MDATA513			√		Undefined
FAD6H	CAN0 message data byte 67 register 13	C0MDATA6713				√	Undefined
FAD6H	CAN0 message data byte 6 register 13	C0MDATA613			√		Undefined
FAD7H	CAN0 message data byte 7 register 13	C0MDATA713			√		Undefined
FAD8H	CAN0 message data length code register 13	C0MDLC13			√		0000xxxxB
FAD9H	CAN0 message configuration register 13	C0MCONF13			√		Undefined
FADAH	CAN0 message ID register 13	C0MIDL13				√	Undefined
FADCH		C0MIDH13				√	Undefined
FADEH	CAN0 message control register 13	C0MCTRL13				√	00x00000 000xx000B

Table 16-16. Register Access Types (8/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FAE0H	CAN0 message data byte 01 register 14	C0MDATA0114	R/W			√	Undefined
FAE0H	CAN0 message data byte 0 register 14	C0MDATA014			√		Undefined
FAE1H	CAN0 message data byte 1 register 14	C0MDATA114			√		Undefined
FAE2H	CAN0 message data byte 23 register 14	C0MDATA2314				√	Undefined
FAE2H	CAN0 message data byte 2 register 14	C0MDATA214			√		Undefined
FAE3H	CAN0 message data byte 3 register 14	C0MDATA314			√		Undefined
FAE4H	CAN0 message data byte 45 register 14	C0MDATA4514				√	Undefined
FAE4H	CAN0 message data byte 4 register 14	C0MDATA414			√		Undefined
FAE5H	CAN0 message data byte 5 register 14	C0MDATA514			√		Undefined
FAE6H	CAN0 message data byte 67 register 14	C0MDATA6714				√	Undefined
FAE6H	CAN0 message data byte 6 register 14	C0MDATA614			√		Undefined
FAE7H	CAN0 message data byte 7 register 14	C0MDATA714			√		Undefined
FAE8H	CAN0 message data length code register 14	C0MDLC14			√		0000xxxxB
FAE9H	CAN0 message configuration register 14	C0MCONF14			√		Undefined
FAEAH	CAN0 message ID register 14	C0MIDL14				√	Undefined
FAECH		C0MIDH14				√	Undefined
FAEEH	CAN0 message control register 14	C0MCTRL14				√	00x00000 000xx000B
FAF0H	CAN0 message data byte 01 register 15	C0MDATA0115				√	Undefined
FAF0H	CAN0 message data byte 0 register 15	C0MDATA015			√		Undefined
FAF1H	CAN0 message data byte 1 register 15	C0MDATA115			√		Undefined
FAF2H	CAN0 message data byte 23 register 15	C0MDATA2315				√	Undefined
FAF2H	CAN0 message data byte 2 register 15	C0MDATA215			√		Undefined
FAF3H	CAN0 message data byte 3 register 15	C0MDATA315			√		Undefined
FAF4H	CAN0 message data byte 45 register 15	C0MDATA4515				√	Undefined
FAF4H	CAN0 message data byte 4 register 15	C0MDATA415			√		Undefined
FAF5H	CAN0 message data byte 5 register 15	C0MDATA515			√		Undefined
FAF6H	CAN0 message data byte 67 register 15	C0MDATA6715				√	Undefined
FAF6H	CAN0 message data byte 6 register 15	C0MDATA615			√		Undefined
FAF7H	CAN0 message data byte 7 register 15	C0MDATA715			√		Undefined
FAF8H	CAN0 message data length code register 15	C0MDLC15			√		0000xxxx
FAF9H	CAN0 message configuration register 15	C0MCONF15			√		Undefined
FAFAH	CAN0 message ID register 15	C0MIDL15				√	Undefined
FAFCH		C0MIDH15				√	Undefined
FAFEH	CAN0 message control register 15	C0MCTRL15				√	00x00000 000xx000B

Table 16-16. Register Access Types (9/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FF60H	CAN0 module receive history list register	C0RGPT	R/W	-	–	√	xx02H
FF62H	CAN0 module transmit history list register	C0TGPT	R/W	–	–	√	xx02H
FF64H	CAN0 global control register	C0GMCTRL	R/W	–	–	√	0000H
FF66H	CAN0 global automatic block transmission control register	C0GMABT	R/W	–	–	√	0000H
FF68H	CAN0 module last out-pointer register	C0LOPT	R	–	√	–	Undefined
FF6EH	CAN0 global clock select register	C0GMCS	R/W	–	√	–	0FH
FF6FH	CAN0 global automatic block transmission delay setting register	C0GMABTD	R/W	–	√	–	00H
FF70H	CAN0 module mask 1 register	C0MASK1L	R/W	–	–	√	Undefined
FF72H		C0MASK1H					
FF74H	CAN0 module mask 2 register	C0MASK2L	R/W	–	–	√	Undefined
FF76H		C0MASK2H					
FF78H	CAN0 module mask 3 register	C0MASK3L	R/W	–	–	√	Undefined
FF7AH		C0MASK3H					
FF7CH	CAN0 module mask 4 register	C0MASK4L	R/W	–	–	√	Undefined
FF7EH		C0MASK4H					
FF8AH	CAN0 module time stamp register	C0TS	R/W	–	–	√	0000H
FF90H	CAN0 module control register	C0CTRL	R/W	–	–	√	0000H
FF92H	CAN0 module last error information register	C0LEC	R/W	–	√	–	00H
FF93H	CAN0 module information register	C0INFO	R	–	√	–	00H
FF94H	CAN0 module error counter register	C0ERC	R	–	–	√	0000H
FF96H	CAN0 module interrupt enable register	C0IE	R/W	–	–	√	0000H
FF98H	CAN0 module interrupt status register	C0INTS	R/W	–	–	√	0000H
FF9CH	CAN0 module bit rate register	C0BTR	R/W	–	–	√	370FH
FF9EH	CAN0 module bit rate prescaler register	C0BRP	R/W	–	√	–	FFH
FF9FH	CAN0 module last in-pointer register	C0LIPT	R	–	√	–	Undefined

16.5.3 Register bit configuration

Table 16-17. Bit Configuration of CAN Global Registers

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
FF64H	C0GMCTRL(W)	0	0	0	0	0	0	0	Clear GOM
FF65H		0	0	0	0	0	0	Set EFSD	Set GOM
FF64H	C0GMCTRL(R)	0	0	0	0	0	0	EFSD	GOM
FF65H		MBON	0	0	0	0	0	0	0
FF66H	C0GMABT(W)	0	0	0	0	0	0	0	Clear ABTTRG
FF67H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
FF66H	C0GMABT(R)	0	0	0	0	0	0	ABTCLR	ABTTRG
FF67H		0	0	0	0	0	0	0	0
FF6EH	C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0
FF6FH	C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remark (R) When read
(W) When write

Table 16-18. Bit Configuration of CAN Module Registers (1/2)

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
FF60H	C0RGPT(W)	0	0	0	0	0	0	0	Clear ROVF
FF61H		0	0	0	0	0	0	0	0
FF60H	C0RGPT(R)	0	0	0	0	0	0	RHPM	ROVF
FF61H		RGPT[7:0]							
FF62H	C0LOPT	LOPT[7:0]							
FF64H	C0TGPT(W)	0	0	0	0	0	0	0	Clear TOVF
FF65H		0	0	0	0	0	0	0	0
FF64H	C0TGPT(R)	0	0	0	0	0	0	THPM	TOVF
FF65H		TGPT[7:0]							
FF70H	C0MASK1L	CM1ID [7:0]							
FF71H		CM1ID [15:8]							
FF72H	C0MASK1H	CM1ID [23:16]							
FF73H		0	0	0	CM1ID [28:24]				
FF74H	C0MASK2L	CM2ID [7:0]							
FF75H		CM2ID [15:8]							
FF76H	C0MASK2H	CM2ID [23:16]							
FF77H		0	0	0	CM2ID [28:24]				
FF78H	C0MASK3L	CM3ID [7:0]							
FF79H		CM3ID [15:8]							
FF7AH	C0MASK3H	CM3ID [23:16]							
FF7BH		0	0	0	CM3ID [28:24]				
FF7CH	C0MASK4L	CM4ID [7:0]							
FF7DH		CM4ID [15:8]							
FF7EH	C0MASK4H	CM4ID [23:16]							
FF7FH		0	0	0	CM4ID [28:24]				
FF8AH	C0TS(W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN
FF8BH		0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
FF8AH	C0TS(R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
FF8BH		0	0	0	0	0	0	0	0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remark (R) When read
(W) When write

Table 16-18. Bit Configuration of CAN Module Registers (2/2)

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
FF90H	C0CTRL(W)	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE 1	Clear PSMODE 0	Clear OPMODE 2	Clear OPMODE 1	Clear OPMODE 0
FF91H		Set CCERC	Set AL	0	Set PSMODE 1	Set PSMODE 0	Set OPMODE 2	Set OPMODE 1	Set OPMODE 0
FF90H	C0CTRL(R)	CCERC	AL	VALID	PS MODE1	PS MODE0	OP MODE2	OP MODE1	OP MODE0
FF91H		0	0	0	0	0	0	RSTAT	TSTAT
FF92H	C0LEC(W)	0	0	0	0	0	0	0	0
FF92H	C0LEC(R)	0	0	0	0	0	LEC2	LEC1	LEC0
FF93H	C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
FF94H	C0ERC	TEC[7:0]							
FF95H		REC[7:0]							
FF96H	C0IE(W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
FF97H		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
FF96H	C0IE(R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
FF97H		0	0	0	0	0	0	0	0
FF98H	C0INTS(W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
FF99H		0	0	0	0	0	0	0	0
FF98H	C0INTS(R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
FF99H		0	0	0	0	0	0	0	0
FF9CH	C0BTR	0	0	0	0	TSEG1[3:0]			
FF9DH		0	0	SJW[1:0]		0	TSEG2[2:0]		
FF9EH	C0BRP	TQPRS[7:0]							
FF9FH	C0LIPT	LIPT[7:0]							

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remark (R) When read
(W) When write

Table 16-19. Bit Configuration of Message Buffer Registers

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
FAx0H	C0MDATA01m	Message data (byte 0)							
FAx1H		Message data (byte 1)							
FAx0H	C0MDATA0m	Message data (byte 0)							
FAx1H	C0MDATA1m	Message data (byte 1)							
FAx2H	C0MDATA23m	Message data (byte 2)							
FAx3H		Message data (byte 3)							
FAx2H	C0MDATA2m	Message data (byte 2)							
FAx3H	C0MDATA3m	Message data (byte 3)							
FAx4H	C0MDATA45m	Message data (byte 4)							
FAx5H		Message data (byte 5)							
FAx4H	C0MDATA4m	Message data (byte 4)							
FAx5H	C0MDATA5m	Message data (byte 5)							
FAx6H	C0MDATA67m	Message data (byte 6)							
FAx7H		Message data (byte 7)							
FAx6H	C0MDATA6m	Message data (byte 6)							
FAx7H	C0MDATA7m	Message data (byte 7)							
FAx8H	C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0
FAx9H	C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
FAxAH	C0MIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
FAxBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
FAxCH	C0MIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
FAxDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
FAxEH	C0MCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
FAxFH		0	0	0	0	Set IE	0	Set TRQ	Set RDY
FAxEH	C0MCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
FAxFH		0	0	MUC	0	0	0	0	0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remarks 1. (R) When read
(W) When write

2. m = 0 to 15

16.6 Bit Set/Clear Function

The CAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written directly. Do not write any values directly via bit manipulation, read/modify/write, or direct writing of target values.

- CAN global control register (C0GMCTRL)
- CAN global automatic block transmission control register (C0GMABT)
- CAN module control register (C0CTRL)
- CAN module interrupt enable register (C0IE)
- CAN module interrupt status register (C0INTS)
- CAN module receive history list register (C0RGPT)
- CAN module transmit history list register (C0TGPT)
- CAN module time stamp register (C0TS)
- CAN message control register (C0MCTRLm)

Remark m = 0 to 15

All the 16 bits in the above registers can be read via the usual method. Use the procedure described in figure 16-23 below to set or clear the lower 8 bits in these registers.

Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (refer to the 16-bit data after a write operation in **Figure 16-24**). **Figure 16-23** shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 16-23. Example of Bit Setting/Clearing Operations

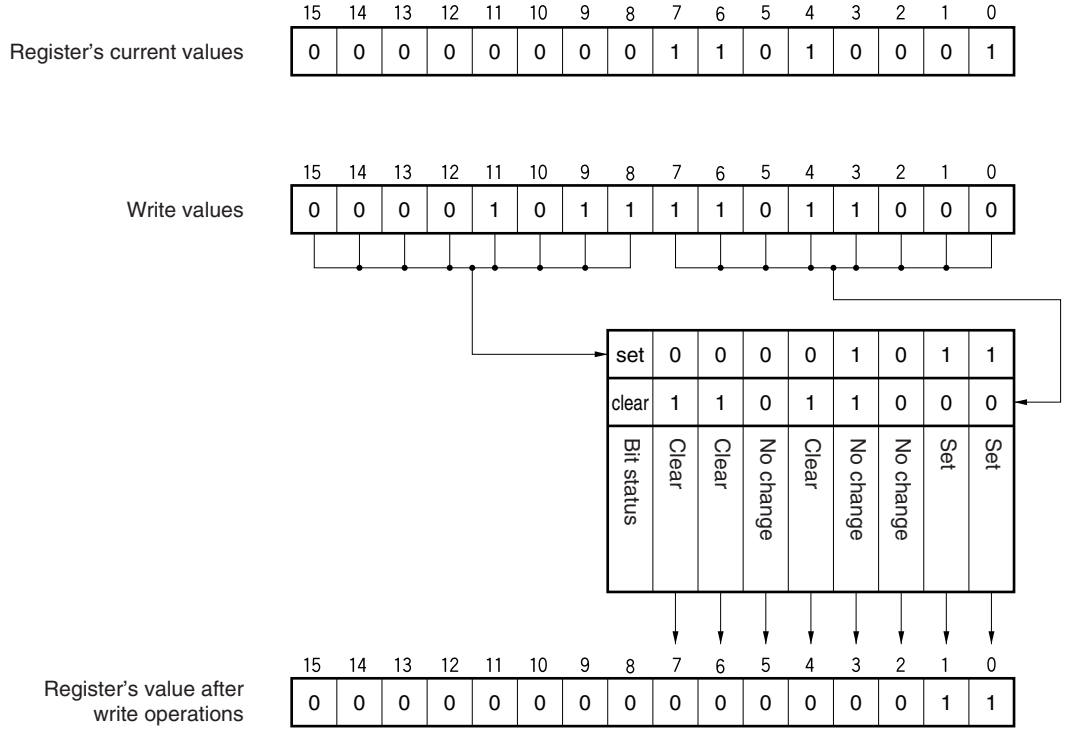


Figure 16-24. 16-Bit Data during Write Operation

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

set n	clear n	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

Remark n = 0 to 7

16.7 Control Registers

Remark m = 0 to 15

(1) CAN global control register (C0GMCTRL)

The C0GMCTRL register is used to control the operation of the CAN module.

After reset: 0000H R/W Address: FF64H, FF65H

(a) Read

	15	14	13	12	11	10	9	8
C0GMCTRL	MBON	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EFSD	GOM

(b) Write

	15	14	13	12	11	10	9	8
C0GMCTRL	0	0	0	0	0	0	Set EFSD	Set GOM
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear GOM

(a) Read

MBON	Bit Enabling Access to Message Buffer Register, Transmit/Receive History List Registers
0	Write access and read access to the message buffer register and the transmit/receive history list registers is disabled.
1	Write access and read access to the message buffer register and the transmit/receive history list registers is enabled.

Cautions 1. While the MBON bit is cleared (to 0), software access to the message buffers (C0MDATA0m, C0MDATA1m, C0MDATA01m, C0MDATA2m, C0MDATA3m, C0MDATA23m, C0MDATA4m, C0MDATA5m, C0MDATA45m, C0MDATA6m, C0MDATA7m, C0MDATA67m, C0MDLCm, C0MCONFm, C0MIDLm, C0MIDHm, and C0MCTRLm), or registers related to transmit history or receive history (C0LOPT, C0TGPT, C0LIPT, and C0RGPT) is disabled.

2. This bit is read-only. Even if 1 is written to MBON while it is 0, the value of MBON does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

Remark MBON bit is cleared (to 0) when the CAN module enters CAN sleep mode/CAN stop mode or GOM bit is cleared (to 0).

MBON bit is set (to 1) when the CAN sleep mode/the CAN stop mode is released or GOM bit is set (to 1).

EFSD	Bit Enabling Forced Shut Down
0	Forced shut down by GOM = 0 disabled.
1	Forced shut down by GOM = 0 enabled.

Caution To request forced shutdown, the GOM bit must be cleared to 0 in a subsequent, immediately following write access after the EFSD bit has been set to 1. If access to another register (including reading the C0GMCTRL register) is executed without clearing the GOM bit immediately after the EFSD bit has been set to 1, the EFSD bit is forcibly cleared to 0, and the forced shutdown request is invalid.

GOM	Global Operation Mode Bit
0	CAN module is disabled from operating.
1	CAN module is enabled to operate.

Caution The GOM bit can be cleared only in the initialization mode or immediately after EFSD bit is set (to 1).

(b) Write

Set EFSD	EFSD Bit Setting
0	No change in ESFD bit .
1	EFSD bit set to 1.

Set GOM	Clear GOM	GOM Bit Setting
0	1	GOM bit cleared to 0.
1	0	GOM bit set to 1.
Other than above		No change in GOM bit.

Caution Set GOM bit and ESFD bit always separately.

(2) CAN global clock selection register (C0GMCS)

The C0GMCS register is used to select the CAN module system clock.

After reset: 0FH R/W Address: FF6EH

	7	6	5	4	3	2	1	0
C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN Module System Clock (f_{CANMOD})
0	0	0	0	f_{CAN}/1
0	0	0	1	f_{CAN}/2
0	0	1	0	f_{CAN}/3
0	0	1	1	f_{CAN}/4
0	1	0	0	f_{CAN}/5
0	1	0	1	f_{CAN}/6
0	1	1	0	f_{CAN}/7
0	1	1	1	f_{CAN}/8
1	0	0	0	f_{CAN}/9
1	0	0	1	f_{CAN}/10
1	0	1	0	f_{CAN}/11
1	0	1	1	f_{CAN}/12
1	1	0	0	f_{CAN}/13
1	1	0	1	f_{CAN}/14
1	1	1	0	f_{CAN}/15
1	1	1	1	f_{CAN}/16 (Default value)

Remark f_{CAN} = Clock supplied to CAN

(3) CAN global automatic block transmission control register (C0GMABT)

The C0GMABT register is used to control the automatic block transmission (ABT) operation.

After reset: 0000H R/W Address: FF66H, FF67H

(a) Read

	15	14	13	12	11	10	9	8
C0GMABT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ABTCLR	ABTTRG

(b) Write

	15	14	13	12	11	10	9	8
C0GMABT	0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ABTTRG

Caution Before changing the normal operation mode with ABT to the initialization mode, be sure to set the C0GMABT register to the default value (0000H).

(a) Read

ABTCLR	Automatic Block Transmission Engine Clear Status Bit
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

Remarks 1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared (0).
The operation is not guaranteed if the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1.

2. When the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 as soon as the requested clearing processing is complete.

ABTTRG	Automatic Block Transmission Status Bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

Caution Do not set the ABTTRG bit (ABTTRG = 1) in the initialization mode. If the ABTTRG bit is set in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.

(b) Write

Set ABTCLR	Automatic Block Transmission Engine Clear Request Bit
0	The automatic block transmission engine is in idle state or under operation.
1	Request to clear the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the ABTTRG bit to 1.

Set ABTTRG	Clear ABTTRG	Automatic Block Transmission Start Bit
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change in ABTTRG bit.

(4) CAN global automatic block transmission delay setting register (C0GMABTD)

The C0GMABTD register is used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in the normal operation mode with ABT.

After reset: 00H R/W Address: FF6FH

	7	6	5	4	3	2	1	0
C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	Data frame interval during automatic block transmission (unit: Data bit time (DBT))
0	0	0	0	0 DBT (default value)
0	0	0	1	2^5 DBT
0	0	1	0	2^6 DBT
0	0	1	1	2^7 DBT
0	1	0	0	2^8 DBT
0	1	0	1	2^9 DBT
0	1	1	0	2^{10} DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
Other than above				Setting prohibited

Cautions 1. Do not change the contents of the C0GMABTD register while the ABTTRG bit is set to 1.

2. The timing at which the ABT message is actually transmitted onto the CAN bus differs depending on the status of transmission from the other station or how a request to transmit a message other than an ABT message (message buffers 8 to 15) is made.

(5) CAN module mask control register (C0MASKaL, C0MASKaH) (a = 1, 2, 3, or 4)

The C0MASKaL and C0MASKaH registers are used to extend the number of receivable messages into the same message buffer by masking part of the ID comparison of a message and invalidating the ID of the masked part.

- CAN Module Mask 1 Register (C0MASK1L, C0MASK1H)

After reset: Undefined R/W Address: C0MASK1L FF70H, FF71H
C0MASK1H FF72H, FF73H

	15	14	13	12	11	10	9	8
C0MASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK1H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 2 Register (C0MASK2L, C0MASK2H)

After reset: Undefined R/W Address: C0MASK2L FF74H, FF75H
C0MASK2H FF76H, FF77H

	15	14	13	12	11	10	9	8
C0MASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK2H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 3 Register (C0MASK3L, C0MASK3H)

After reset: Undefined R/W Address: C0MASK3L FF78H, FF79H
C0MASK3H FF7AH, FF7BH

	15	14	13	12	11	10	9	8
C0MASK3L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK3H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 4 Register (C0MASK4L, C0MASK4H)

After reset: Undefined R/W Address: C0MASK4L FF7CH, FF7DH
C0MASK4H FF7EH, FF7FH

	15	14	13	12	11	10	9	8
C0MASK4L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK4H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMID28-CMID0	Sets Mask Pattern of ID Bit.
0	The ID bits of the message buffer set by the CMID28 to CMID0 bits are compared with the ID bits of the received message frame.
1	The ID bits of the message buffer set by the CMID28 to CMID0 bits are not compared with the ID bits of the received message frame (they are masked).

Remark Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

(6) CAN module control register (C0CTRL)

The C0CTRL register is used to control the operation mode of the CAN module.

After reset: 0000H R/W Address: FF90H, FF91H

(a) Read

	15	14	13	12	11	10	9	8
C0CTRL	0	0	0	0	0	0	RSTAT	TSTAT
	7	6	5	4	3	2	1	0
	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0

(b) Write

	15	14	13	12	11	10	9	8
C0CTRL	Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
	7	6	5	4	3	2	1	0
	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0

(a) Read

RSTAT	Reception Status Bit
0	Reception is stopped.
1	Reception is in progress.

Remark - The RSTAT bit is set to 1 under the following conditions (timing).

- The SOF bit of a receive frame is detected
- On occurrence of arbitration loss during a transmit frame

- The RSTAT bit is cleared to 0 under the following conditions (timing)
  - When a recessive level is detected at the second bit of the interframe space
  - On transition to the initialization mode at the first bit of the interframe space

TSTAT	Transmission Status Bit
0	Transmission is stopped.
1	Transmission is in progress.

Remark - The TSTAT bit is set to 1 under the following conditions (timing).

- The SOF bit of a transmit frame is detected
- The first bit of an error flag is detected during a transmit frame

- The TSTAT bit is cleared to 0 under the following conditions (timing).

- During transition to bus-off state
- On occurrence of arbitration loss in transmit frame
- On detection of recessive level at the second bit of the interframe space
- On transition to the initialization mode at the first bit of the interframe space

CCERC	Error Counter Clear Bit
0	The C0ERC and C0INFO registers are not cleared in the initialization mode.
1	The C0ERC and C0INFO registers are cleared in the initialization mode.

Remarks 1. The CCERC bit is used to clear the C0ERC and C0INFO registers for re-initialization or forced recovery from the bus-off state. This bit can be set to 1 only in the initialization mode.

2. When the C0ERC and C0INFO registers have been cleared, the CCERC bit is also cleared to 0 automatically.

3. The CCERC bit can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.4. The CCERC bit is read-only in the CAN sleep mode or CAN stop mode.

4. The receive data may be corrupted in case of setting the CCERC bit to (1) immediately after entering the INIT mode from self-test mode.

AL	Bit to Set Operation in Case of Arbitration Loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode.

Remark The AL bit is valid only in the single-shot mode.

VALID	Valid Receive Message Frame Detection Bit
0	A valid message frame has not been received since the VALID bit was last cleared to 0.
1	A valid message frame has been received since the VALID bit was last cleared to 0.

Remarks 1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame) or transmit message buffer (remote frame).

2. Clear the VALID bit (0) before changing the initialization mode to an operation mode.

3. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal operation mode and the other in the receive-only mode, the VALID bit is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.

4. In order to clear the VALID bit, set the Clear VALID bit to 1 first and confirm that the VALID bit is cleared. If it is not cleared, perform clearing processing again.

PSMODE1	PSMODE0	Power Save Mode
0	0	No power save mode is selected.
0	1	CAN sleep mode
1	0	Setting prohibited
1	1	CAN stop mode

Cautions1. Transition to and from the CAN stop mode must be made via CAN sleep mode. A request for direct transition to and from the CAN stop mode is ignored.

2. The MBON flag of C0GMCTRL must be checked after releasing a power save mode, prior to access the message buffers again.

3. CAN Sleep mode requests are kept pending, until cancelled by software or entered on appropriate bus condition (bus idle). Software can check the actual status by reading PSMODE.

OPMODE2	OPMODE1	OPMODE0	Operation Mode
0	0	0	No operation mode is selected (CAN module is in the initialization mode).
0	0	1	Normal operation mode
0	1	0	Normal operation mode with automatic block transmission function (normal operation mode with ABT)
0	1	1	Receive-only mode
1	0	0	Single-shot mode
1	0	1	Self-test mode
Other than above			Setting prohibited

Caution Transit to initialization mode or power saving modes may take some time. Be sure to verify the success of mode change by reading the values, before proceeding.

Remark The OPMODE[2:0] bits are read-only in the CAN sleep mode or CAN stop mode.

(b)Write

Set CCERC	Clear CCERC	Setting of CCERC Bit
1	1	CCERC bit is set to 1.
Other than above	0	CCERC bit is not changed.

Set AL	Clear AL	Setting of AL Bit
0	1	AL bit is cleared to 0.
1	0	AL bit is set to 1.
Other than above		AL bit is not changed.

Clear VALID	Setting of VALID Bit
0	VALID bit is not changed.
1	VALID bit is cleared to 0.

Set PSMODE0	Clear PSMODE0	Setting of PSMODE0 Bit
0	1	PSMODE0 bit is cleared to 0.
1	0	PSMODE bit is set to 1.
Other than above		PSMODE0 bit is not changed.

Set PSMODE1	Clear PSMODE1	Setting of PSMODE1 Bit
0	1	PSMODE1 bit is cleared to 0.
1	0	PSMODE1 bit is set to 1.
Other than above		PSMODE1 bit is not changed.

Set OPMODE0	Clear OPMODE0	Setting of OPMODE0 Bit
0	1	OPMODE0 bit is cleared to 0.
1	0	OPMODE0 bit is set to 1.
Other than above		OPMODE0 bit is not changed.

Set OPMODE1	Clear OPMODE1	Setting of OPMODE1 Bit
0	1	OPMODE1 bit is cleared to 0.
1	0	OPMODE1 bit is set to 1.
Other than above		OPMODE1 bit is not changed.

Set OPMODE2	Clear OPMODE2	Setting of OPMODE2 Bit
0	1	OPMODE2 bit is cleared to 0.
1	0	OPMODE2 bit is set to 1.
Other than above		OPMODE2 bit is not changed.

(7) CAN module last error code register (C0LEC)

The C0LEC register provides the error information of the CAN protocol.

After reset: 00H R/W Address: FF92H

	7	6	5	4	3	2	1	0
C0LEC	0	0	0	0	0	LEC2	LEC1	LEC0

Remarks 1. The contents of the C0LEC register are not cleared when the CAN module changes from an operation mode to the initialization mode.

2. If an attempt is made to write a value other than 00H to the C0LEC register by software, the access is ignored.

LEC2	LEC1	LEC0	Last CAN Protocol Error Information
0	0	0	No error
0	0	1	Stuff error
0	1	0	Form error
0	1	1	ACK error
1	0	0	Bit error (The CAN module tried to transmit a recessive-level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
1	0	1	Bit error (The CAN module tried to transmit a dominant-level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
1	1	0	CRC error
1	1	1	Undefined

(8) CAN module information register (C0INFO)

The C0INFO register indicates the status of the CAN module.

After reset: 00H R Address: FF93H

	7	6	5	4	3	2	1	0
C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

BOFF	Bus-off State Bit
0	Not bus-off state (transmit error counter ≤ 255) (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter > 255) (The value of the transmit counter is 256 or more.)

TECS1	TECS0	Transmission Error Counter Status Bit
0	0	The value of the transmission error counter is less than that of the warning level (<96).
0	1	The value of the transmission error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the transmission error counter is in the range of the error passive or bus-off state (≥ 128).

RECS1	RECS0	Reception Error Counter Status Bit
0	0	The value of the reception error counter is less than that of the warning level (<96).
0	1	The value of the reception error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the reception error counter is in the error passive range (≥ 128).

(9) CAN module error counter register (C0ERC)

The C0ERC register indicates the count value of the transmission/reception error counter.

After reset: 0000H R Address: FF94H, FF95H

	15	14	13	12	11	10	9	8
C0ERC	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	Reception error passive status bit
0	Reception error counter is not error passive (<128)
1	Reception error counter is error passive range (≥128)

REC6-REC0	Reception Error Counter Bit
0-127	Number of reception errors. These bits reflect the status of the reception error counter. The number of errors is defined by the CAN protocol.

Remark REC6 to REC0 of the reception error counter are invalid in the reception error passive state (RECS [1:0] = 11B).

TEC7-TEC0	Transmission Error Counter Bit
0-255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.

Remark TEC7 to TEC0 of the transmission error counter are invalid in the bus-off state (BOFF = 1).

(10) CAN module interrupt enable register (C0IE)

The C0IE register is used to enable or disable the interrupts of the CAN module.

After reset: 0000H R/W Address: FF96H, FF97H

(a) Read

C0IE	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

(b) Write

C0IE	15	14	13	12	11	10	9	8
	0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
	7	6	5	4	3	2	1	0
	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0

(a) Read

CIE5-CIE0	CAN Module Interrupt Enable Bit
0	Output of the interrupt corresponding to interrupt status register CINTSx bit is disabled.
1	Output of the interrupt corresponding to interrupt status register CINTSx bit is enabled.

(b) Write

Set CIE5	Clear CIE5	Setting of CIE5 Bit
0	1	CIE5 bit is cleared to 0.
1	0	CIE5 bit is set to 1.
Other than above		CIE5 bit is not changed.

Set CIE4	Clear CIE4	Setting of CIE4 Bit
0	1	CIE4 bit is cleared to 0.
1	0	CIE4 bit is set to 1.
Other than above		CIE4 bit is not changed.

Set CIE3	Clear CIE3	Setting of CIE Bit
0	1	CIE3 bit is cleared to 0.
1	0	CIE3 bit is set to 1.
Other than above		CIE3 bit is not changed.

Set CIE2	Clear CIE2	Setting of CIE2 Bit
0	1	CIE2 bit is cleared to 0.
1	0	CIE2 bit is set to 1.
Other than above		CIE2 bit is not changed.

Set CIE1	Clear CIE1	Setting of CIE1 Bit
0	1	CIE1 bit is cleared to 0.
1	0	CIE1 bit is set to 1.
Other than above		CIE1 bit is not changed.

Set CIE0	Clear CIE0	Setting of CIE0 Bit
0	1	CIE0 bit is cleared to 0.
1	0	CIE0 bit is set to 1.
Other than above		CIE0 bit is not changed.

(11) CAN module interrupt status register (C0INTS)

The C0INTS register indicates the interrupt status of the CAN module.

After reset: 0000H R/W Address: FF98H, FF99H

(a) Read

C0INTS	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

(b) Write

C0INTS	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0

(a) Read

CINTS5-CINTS0	CAN Interrupt Status Bit
0	No related interrupt source event is pending.
1	A related interrupt source event is pending.

Interrupt Status Bit	Related Interrupt Source Event
CINTS5	Wakeup interrupt from CAN sleep mode[Note]
CINTS4	Arbitration loss interrupt
CINTS3	CAN protocol error interrupt
CINTS2	CAN error status interrupt
CINTS1	Interrupt on completion of reception of valid message frame to message buffer m
CINTS0	Interrupt on normal completion of transmission of message frame from message buffer m

Note The CINTS5 bit is set only when the CAN module is woken up from the CAN sleep mode by a CAN bus operation. The CINTS5 bit is not set when the CAN sleep mode has been released by software.

(b) Write

Clear CINTS5-CINTS0	Setting of CINTS5 to CINTS0 Bits
0	CINTS5 to CINTS0 bits are not changed.
1	CINTS5 to CINTS0 bits are cleared to 0.

Caution Please clear the status bit of this register with software when the confirmation of each status is necessary in the interrupt processing, because these bits are not cleared automatically.

(12) CAN module bit rate prescaler register (C0BRP)

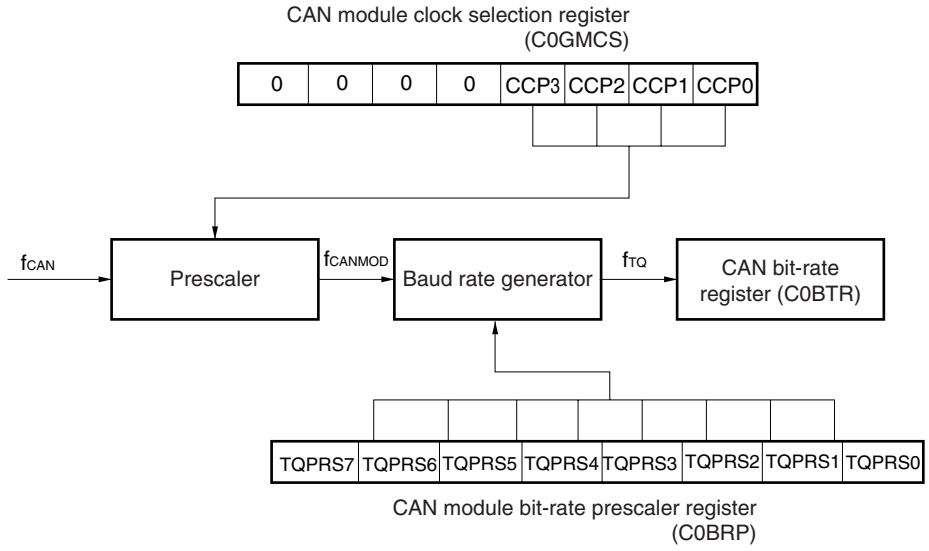
The C0BRP register is used to select the CAN protocol layer basic clock (f_{TQ}). The communication baud rate is set to the C0BTR register.

After reset: FFH R/W Address: FF9EH

	7	6	5	4	3	2	1	0
C0BRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7-TQPRS0	CAN Protocol Layer Basic System Clock (f_{TQ})
0	f_{CANMOD}/1
1	f_{CANMOD}/2
:	:
n	f_{CANMOD}/(n+1)
:	:
255	f_{CANMOD}/256 (default value)

Figure 16-25. CAN Module Clock



Caution The C0BRP register can be write-accessed only in the initialization mode.

Remark f_{CAN}: Clock supplied to CAN (f_{PRS})
f_{CANMOD}: CAN module system clock
f_{TQ}: CAN protocol layer basic system clock

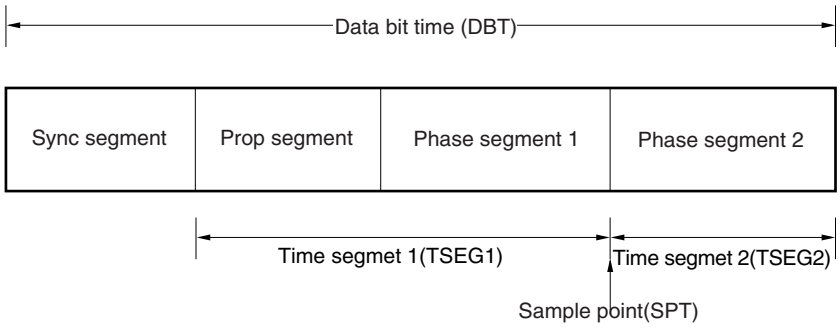
(13) CAN module bit rate register (C0BTR)

The C0BTR register is used to control the data bit time of the communication baud rate.

After reset: 370FH R/W Address: FF9CH, FF9DH

	15	14	13	12	11	10	9	8
C0BTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
	7	6	5	4	3	2	1	0
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

Figure 16-26. Data Bit Time



SJW1	SJW0	Length of Synchronization jump width
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (default value)

TSEG22	TSEG21	TSEG20	Length of time segment 2
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (default value)

TSEG13	TSEG12	TSEG11	TSEG10	Length of time segment 1
0	0	0	0	Setting prohibited
0	0	0	1	2TQ[Note]
0	0	1	0	3TQ[Note]
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (default value)

Note This setting must not be made when the C0BRP register = 00H.

Remark TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)

(14) CAN module last in-pointer register (C0LIPT)

The C0LIPT register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

After reset: Undefined R Address: FF9FH

	7	6	5	4	3	2	1	0
C0LIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7-LIPT0	Last In-Pointer Register (C0LIPT)
0 to 15	When the C0LIPT register is read, the contents of the element indexed by the last in-pointer (LIPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame was last stored.

Remark The read value of the C0LIPT register is undefined if a data frame or a remote frame has never been stored in the message buffer. If the RHPM bit of the C0RGPT register is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the C0LIPT register is undefined.

(15) CAN module receive history list register (C0RGPT)

The C0RGPT register is used to read the receive history list.

After reset: xx02H R/W Address: FF60H, FF61H

(a) Read

	15	14	13	12	11	10	9	8
C0RGPT	RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	RHPM	ROVF

(b) Write

	15	14	13	12	11	10	9	8
C0RGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ROVF

(a) Read

RGPT7-RGPT0	Receive History List Get Pointer
0 to 15	When the C0RGPT register is read, the contents of the element indexed by the receive history list get pointer (RGPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.

RHPM Note	Receive History List Pointer Match
0	The receive history list has at least one message buffer number that has not been read.
1	The receive history list has no message buffer numbers that has not been read.

Note The read value of RGPT0 to RGPT7 is invalid when RHPM = 1.

ROVF	Receive History List Overflow Bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffer in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	All the message buffer numbers that are recorded are preserved except the message buffer number recorded last Note. The RHL is fully loaded with the unread message buffer number and all RHL elements beside the last one are preserved. Message buffer number of subsequent data frame storage or remote frame assignment is always logged in the RHL element LIPT pointer –1 is pointing to. Note that the RHL will be updated, but the LIPT pointer will not be incremented. Always the position the LIPT pointer –1 is pointing to is overwritten (the receive history list does not have a vacant element).

Note If ROVF is set, RHPM is no longer cleared on message storage, but RHPM is still set, if all entries of C0RGPT are read by software.

(b) Write

Clear ROVF	Setting of ROVF Bit
0	ROVF bit is not changed.
1	ROVF bit is cleared to 0.

(16) CAN module last out-pointer register (C0LOPT)

The C0LOPT register indicates the number of the message buffer to which a data frame or a remote frame was transmitted last.

After reset: Undefined R Address: FF68H

	7	6	5	4	3	2	1	0
C0LOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7-LOPT0	Last Out-Pointer of Transmit History List (LOPT)
0 to 15	When the C0LOPT register is read, the contents of the element indexed by the last out-pointer (LOPT) of the receive history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

Remark The value read from the C0LOPT register is undefined if a data frame or remote frame has never been transmitted from a message buffer. If the THPM bit is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the C0LOPT register is undefined.

(17) CAN module transmit history list register (C0TGPT)

The C0TGPT register is used to read the transmit history list.

After reset: xx02H R/W Address: FF62H, FF63H

(a) Read

	15	14	13	12	11	10	9	8
C0TGPT	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	THPM	TOVF

(b) Write

	15	14	13	12	11	10	9	8
C0TGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear TOVF

(a) Read

TGPT7-TGPT0	Transmit History List Read Pointer
0 to 15	When the C0TGPT register is read, the contents of the element indexed by the read pointer (TGPT) of the transmit history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

THPM [Note]	Transmit History Pointer Match
0	The transmit history list has at least one message buffer number that has not been read.
1	The transmit history list has no message buffer number that has not been read.

Note The read value of TGPT0 to TGPT7 is invalid when THPM = 1.

TOVF	Transmit History List Overflow Bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element).
1	At least 7 entries have been stored since the host processor has serviced the THL last time (i.e. read CnTGPT). The first 6 entries are sequentially stored while the last entry can have been overwritten whenever a message is newly transmitted because all buffer numbers are stored at position LOPT-1 when TOVF bit is set. Thus the sequence of transmissions can not be recovered completely now.

Note If TOVF is set, THPM is no longer cleared on message transmission, but THPM is still set, if all entries of C0TGPT are read by software.

Remark Transmission from message buffer 0 to 7 is not recorded to the transmit history list in the normal operation mode with ABT.

(b) Write

Clear TOVF	Setting of TOVF Bit
0	TOVF bit is not changed.
1	TOVF bit is cleared to 0.

(18) CAN module time stamp register (C0TS)

The C0TS register is used to control the time stamp function.

After reset: 0000H R/W Address: FF8AH, FF8BH

(a) Read

	15	14	13	12	11	10	9	8
C0TS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	TSLOCK	TSSEL	TSEN

(b) Write

	15	14	13	12	11	10	9	8
C0TS	0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
	7	6	5	4	3	2	1	0
	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN

Remark The lock function of the time stamp function must not be used when the CAN module is in the normal operation mode with ABT.

(a) Read

TSLOCK	Time Stamp Lock Function Enable Bit
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0[Note].

Note The TSEN bit is automatically cleared to 0.

TSSEL	Time Stamp Capture Event Selection Bit
0	The time stamp capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

TSEN	TSOUT Signal Operation Setting Bit
0	Disable TSOUT signal toggle operation.
1	Enable TSOUT signal toggle operation.

Remark The signal TSOUT is output from the CAN macro to a timer resource, depending on implementation. Refer to documentation of device implementation for details.

(b) Write

Set TSLOCK	Clear TSLOCK	Setting of TSLOCK Bit
0	1	TSLOCK bit is cleared to 0.
1	0	TSLOCK bit is set to 1.
Other than above		TSLOCK bit is not changed.

Set TSSEL	Clear TSSEL	Setting of TSSEL Bit
0	1	TSSEL bit is cleared to 0.
1	0	TSSEL bit is set to 1.
Other than above		TSSEL bit is not changed.

Set TSEN	Clear TSEN	Setting of TSEN Bit
0	1	TSEN bit is cleared to 0.
1	0	TSEN bit is set to 1.
Other than above		TSEN bit is not changed.

(19) CAN message data byte register (C0MDATAxm)(x = 0 to 7), (C0MDATAzm) (z = 01, 23, 45, 67)

The C0MDATAxm, C0MDATAzm registers are used to store the data of a transmit/receive message.

The C0MDATAzm registers can access the C0MDATAxm registers in 16-bit units.

After reset: Undefined R/W Address: See **Table 16-16**

- C0MDATAxm Register

	7	6	5	4	3	2	1	0
C0MDATA0m	MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00

	7	6	5	4	3	2	1	0
C0MDATA1m	MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA10

	7	6	5	4	3	2	1	0
C0MDATA2m	MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20

	7	6	5	4	3	2	1	0
C0MDATA3m	MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30

	7	6	5	4	3	2	1	0
C0MDATA4m	MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40

	7	6	5	4	3	2	1	0
C0MDATA5m	MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50

	7	6	5	4	3	2	1	0
C0MDATA6m	MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60

	7	6	5	4	3	2	1	0
C0MDATA7m	MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

- C0MDATAzm Register

	15	14	13	12	11	10	9	8
C0MDATA01m	MDATA0115	MDATA0114	MDATA0113	MDATA0112	MDATA0111	MDATA0110	MDATA019	MDATA018
	7	6	5	4	3	2	1	0
	MDATA017	MDATA016	MDATA015	MDATA014	MDATA013	MDATA012	MDATA011	MDATA010

	15	14	13	12	11	10	9	8
C0MDATA23m	MDATA2315	MDATA2314	MDATA2313	MDATA2312	MDATA2311	MDATA2310	MDATA239	MDATA238
	7	6	5	4	3	2	1	0
	MDATA237	MDATA236	MDATA235	MDATA234	MDATA233	MDATA232	MDATA231	MDATA230

	15	14	13	12	11	10	9	8
C0MDATA45m	MDATA4515	MDATA4514	MDATA4513	MDATA4512	MDATA4511	MDATA4510	MDATA459	MDATA458
	7	6	5	4	3	2	1	0
	MDATA457	MDATA456	MDATA455	MDATA454	MDATA453	MDATA452	MDATA451	MDATA450

	15	14	13	12	11	10	9	8
C0MDATA67m	MDATA6715	MDATA6714	MDATA6713	MDATA6712	MDATA6711	MDATA6710	MDATA679	MDATA678
	7	6	5	4	3	2	1	0
	MDATA677	MDATA676	MDATA675	MDATA674	MDATA673	MDATA672	MDATA671	MDATA670

(20) CAN message data length register m (C0MDLCm)

The C0MDLCm register is used to set the number of bytes of the data field of a message buffer.

After reset: 0000xxxxB R/W Address: See **Table 16-16**

	7	6	5	4	3	2	1	0
C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0

MDLC3	MDLC2	MDLC1	MDLC0	Data Length Of Transmit/Receive Message
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
1	0	0	1	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set DLC value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.)[Note]
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Note The data and DLC value actually transmitted to CAN bus are as follows.

Type of Transmit Frame	Length of Transmit Data	DLC Transmitted
Data frame	Number of bytes specified by DLC (However, 8 bytes if DLC ≥ 8)	MDLC[3:0]
Remote frame	0 bytes	

Cautions 1. Be sure to set bits 7 to 4 0000B.

2. Receive data is stored in as many C0MDATAx as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. C0MDATAx in which no data is stored is undefined.

(21) CAN message configuration register (C0MCONFm)

The C0MCONFm register is used to specify the type of the message buffer and to set a mask.

After reset: Undefined R/W Address: See **Table 16-16**

	7	6	5	4	3	2	1	0
C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0

OWS	Overwrite Control Bit
0	The message buffer that has already received a data frame[Note] is not overwritten by a newly received data frame. The newly received data frame is discarded.
1	The message buffer that has already received a data frame[Note] is overwritten by a newly received data frame.

Note The “message buffer that has already received a data frame” is a receive message buffer whose DN bit has been set to 1.

Remark A remote frame is received and stored, regardless of the setting of OWS bit and DN bit. A remote frame that satisfies the other conditions (ID matches, RTR = 0, TRQ = 0) is always received and stored in the corresponding message buffer (interrupt generated, DN flag set, MDLC[3:0] bits updated, and recorded to the receive history list).

RTR	Remote Frame Request Bit[Note]
0	Transmit a data frame.
1	Transmit a remote frame.

Note The RTR bit specifies the type of message frame that is transmitted from a message buffer defined as a transmit message buffer.Even if a valid remote frame has been received, RTR of the transmit message buffer that has received the frame remains cleared to 0.Even if a remote frame whose ID matches has been received from the CAN bus with the RTR bit of the transmit message buffer set to 1 to transmit a remote frame, that remote frame is not received or stored (interrupt generated, DN flag set, MDLC[3:0] bits updated, and recorded to the receive history list).

MT2	MT1	MT0	Message Buffer Type Setting Bit
0	0	0	Transmit message buffer
0	0	1	Receive message buffer (no mask setting)
0	1	0	Receive message buffer (mask 1 set)
0	1	1	Receive message buffer (mask 2 set)
1	0	0	Receive message buffer (mask 3 set)
1	0	1	Receive message buffer (mask 4 set)
Other than above			Setting prohibited

MA0	Message Buffer Assignment Bit
0	Message buffer not used.
1	Message buffer used.

Caution Be sure to write 0 to bits 2 and 1.

(22) CAN message id register m (C0MIDLm, C0MIDHm)

The C0MIDLm and C0MIDHm registers are used to set an identifier (ID).

After reset: Undefined R/W Address: See **Table 16-16**

C0MIDLm

15	14	13	12	11	10	9	8
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
7	6	5	4	3	2	1	0
ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

C0MIDHm

15	14	13	12	11	10	9	8
IDE	0	0	ID28	ID27	ID26	ID25	ID24
7	6	5	4	3	2	1	0
ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

IDE	Format Mode Specification Bit
0	Standard format mode (ID28 to ID18: 11 bits)[Note]
1	Extended format mode (ID28 to ID0: 29 bits)

Note The ID17 to ID0 bits are not used.

ID28 to ID0	Message ID
ID28 to ID18	Standard ID value of 11 bits (when IDE = 0)
ID28 to ID0	Extended ID value of 29 bits (when IDE = 1)

Cautions 1. Be sure to write 0 to bits 14 and 13 of the C0MIDHm register.

2. Be sure to align the ID value according to the given bit positions into this registers. Note that for standard ID, the ID value must be shifted to fit into ID28 to ID11 bit positions.

(23) CAN message control register m (C0MCTRLm)

The C0MCTRLm register is used to control the operation of the message buffer.

After reset: 00x000000 00000000B R/W Address: See **Table 16-16**.

(a) Read

	15	14	13	12	11	10	9	8
C0MCTRLm	0	0	MUC	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	MOW	IE	DN	TRQ	RDY

(b) Write

	15	14	13	12	11	10	9	8
C0MCTRLm	0	0	0	0	Set IE	0	Set TRQ	Set RDY
	7	6	5	4	3	2	1	0
	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY

(a) Read

MUC[Note]	Message Buffer Data Updating Bit
0	The CAN module is not updating the message buffer (reception and storage).
1	The CAN module is updating the message buffer (reception and storage).

Note The MUC bit is undefined until the first reception and storage is performed.

MOW	Message Buffer Overwrite Status Bit
0	The message buffer is not overwritten by a newly received data frame.
1	The message buffer is overwritten by a newly received data frame.

Remark MOW bit is not set to 1 even if a remote frame is received and stored in the transmit message buffer with DN = 1.

IE	Message Buffer Interrupt Request Enable Bit
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt enabled.

DN	Message Buffer Data Updating Bit
0	A data frame or remote frame is not stored in the message buffer.
1	A data frame or remote frame is stored in the message buffer.

TRQ	Message Buffer Transmission Request Bit
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

Caution Do not set the TRQ bit and the RDY bit (1) at the same time. Set the RDY bit (1) before setting the TRQ bit.

RDY	Message Buffer Ready Bit
0	The message buffer can be written by software. The CAN module cannot write to the message buffer.
1	Writing the message buffer by software is ignored (except a write access to the RDY, TRQ, DN, and MOW bits). The CAN module can write to the message buffer.

Cautions 1. Do not clear the RDY bit (0) during message transmission. Follow the transmission abort process about clearing the RDY bit (0) for redefinition of the message buffer.

2. Clear again when RDY bit is not cleared even if this bit is cleared.

3. Be sure that RDY is cleared before writing to the other message buffer registers, by checking the status of the RDY bit.

(b) Write

Clear MOW	Setting of MOW Bit
0	MOW bit is not changed.
1	MOW bit is cleared to 0.

Set IE	Clear IE	Setting of IE Bit
0	1	IE bit is cleared to 0.
1	0	IE bit is set to 1.
Other than above		IE bit is not changed.

Caution Set IE bit and RDY bit always separately.

Clear DN	Setting of DN Bit
0	DN bit is not changed.
1	DN bit is cleared to 0.

Caution Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10.

Set TRQ	Clear TRQ	Setting of TRQ Bit
0	1	TRQ bit is cleared to 0.
1	0	TRQ bit is set to 1.
Other than above		TRQ bit is not changed.

Set RDY	Clear RDY	Setting of RDY Bit
0	1	RDY bit is cleared to 0.
1	0	RDY bit is set to 1.
Other than above		RDY bit is not changed.

Caution Set IE bit and RDY bit always separately.

16.8 CAN Controller Initialization

16.8.1 Initialization of CAN module

Before the CAN module operation is enabled, the CAN module system clock needs to be determined by setting the CCP[3:0] bits of the C0GMCS register by software. Do not change the setting of the CAN module system clock after CAN module operation is enabled.

The CAN module is enabled by setting the GOM bit of the C0GMCTRL register.

For the procedure of initializing the CAN module, refer to **16.16 Operation Of CAN Controller.**

16.8.2 Initialization of message buffer

After the CAN module is enabled, the message buffers contain undefined values. A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the CAN module from the initialization mode to one of the operation modes.

- Clear the RDY, TRQ, and DN bits of the C0MCTRLm register to 0.
- Clear the MA0 bit of the C0MCONFm register to 0.

Remark m = 0 to 15

16.8.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To redefine message buffer in initialization mode

Place the CAN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the CAN module in an operation mode.

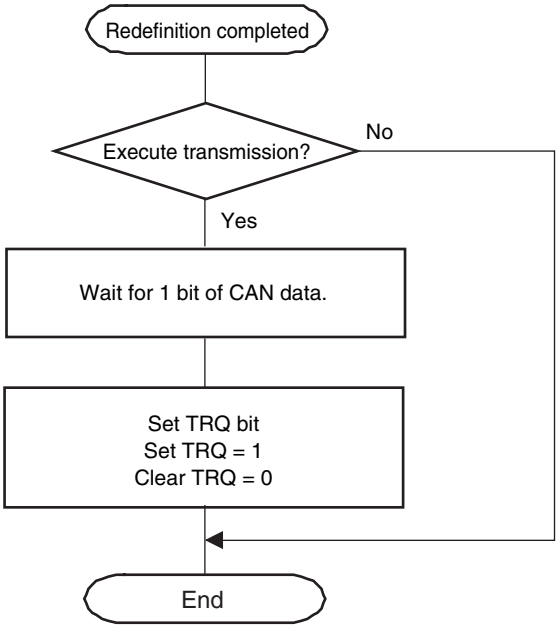
(2) To redefine message buffer during reception

Perform redefinition as shown in Figure 16-40.

(3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (refer to **16.10.4 (1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)** and **16.10.4 (2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)**. Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.

Figure 16-27. Setting Transmission Request (TRQ) to Transmit Message Buffer After Redefining



Cautions1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in Figure 16-40 is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.

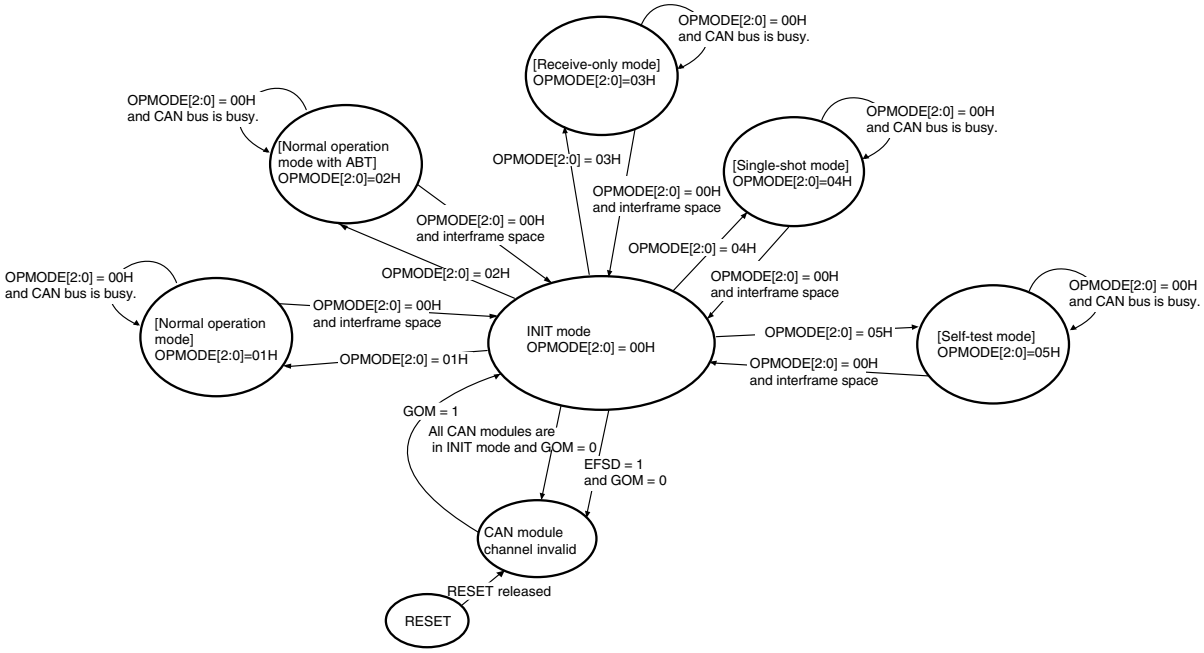
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 16-41 is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

16.8.4 Transition from initialization mode to operation mode

The CAN module can be switched to the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

Figure 16-28. Transition to Operation Modes



The transition from the initialization mode to an operation mode is controlled by the bit string OPMODE[2:0] in the C0CTRL register.

Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.

Requests for transition from the operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the CAN module enters the initialization mode at the first bit in the interframe space (the value of OPMODE[2:0] are changed to 00H). After issuing a request to change the mode to the initialization mode, read the OPMODE[2:0] bits until their value becomes 000B to confirm that the module has entered the initialization mode (refer to **Figure 16-37**).

16.8.5 Resetting error counter C0ERC of CAN module

If it is necessary to reset the CAN module error counter C0ERC and the CAN module information register C0INFO when re-initialization or forced recovery from the bus-off state is made, set the CCERC bit of the C0CTRL register to 1 in the initialization mode. When this bit is set to 1, the CAN module error counter C0ERC and the CAN module information register C0INFO are cleared to their default values.

16.9 Message Reception

16.9.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (MA0 bit of C0MCONFm register set to 1B.)
- Set as a receive message buffer
  (MT[2:0] bits of C0MCONFm register set to 001B, 010B, 011B, 100B, or 101B.)
- Ready for reception
  (RDY bit of C0MCTRLm register set to 1.)

When two or more message buffers of the CAN module receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set to store a message in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to receive and store a message (i.e., when DN = 1 indicating that a message has already been received, but rewriting is disabled because OWS = 0). In this case, the message is not actually received and stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Priority	Storing Condition If Same ID is Set	
1 (high)	Unmasked message buffer	DN = 0
		DN = 1 and OWS = 1
2	Message buffer linked to mask 1	DN = 0
		DN = 1 and OWS = 1
3	Message buffer linked to mask 2	DN = 0
		DN = 1 and OWS = 1
4	Message buffer linked to mask 3	DN = 0
		DN = 1 and OWS = 1
5(low)	Message buffer linked to mask 4	DN = 0
		DN = 1 and OWS = 1

Remark m = 0 to 15

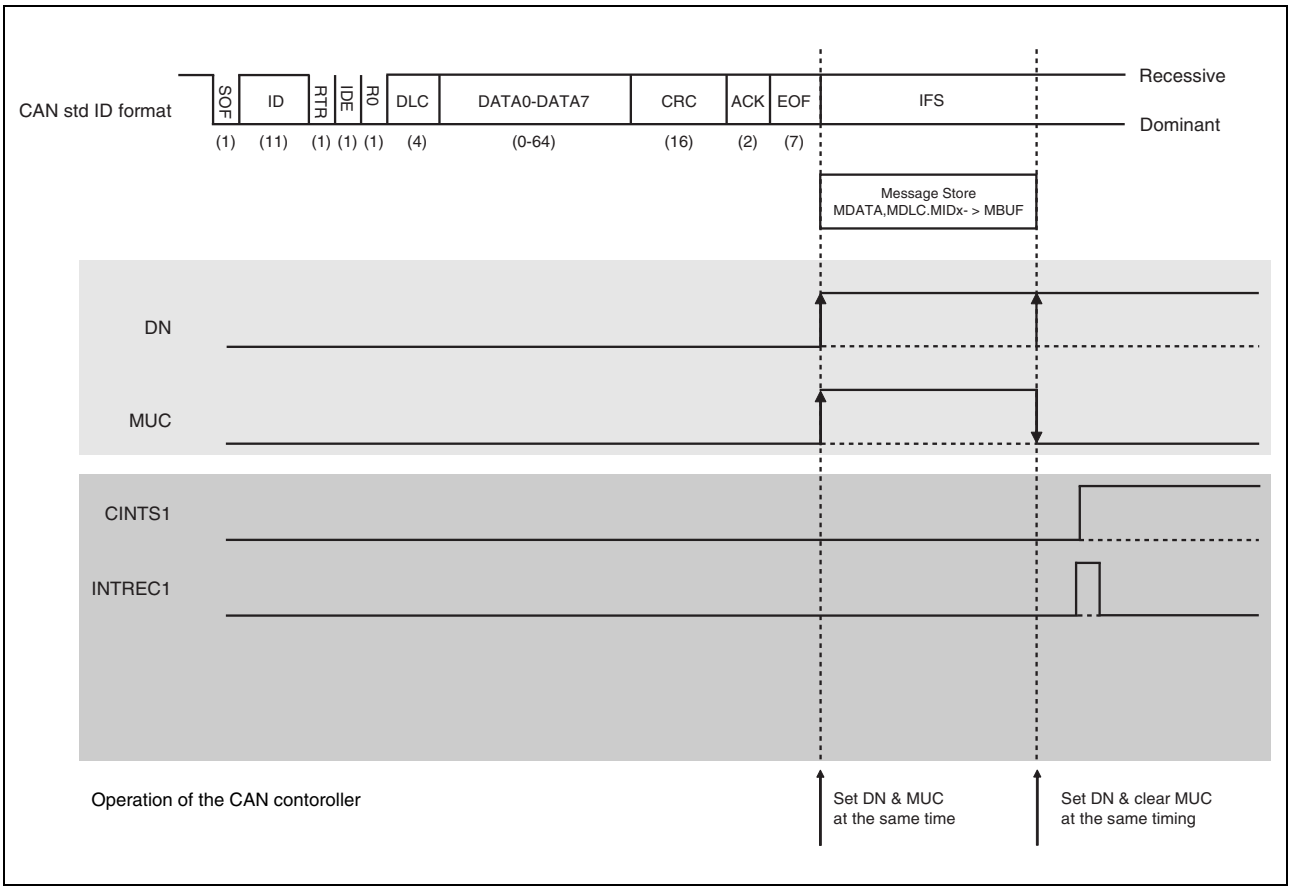
16.9.2 Receive Data Read

To keep data consistency when reading CAN message buffers, perform the data reading according to Figure 16-51 to 16-53.

During message reception, the CAN module sets DN of the C0MCTRLm register two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, the MUC bit of the C0MCTRLm register of the message buffer is set. (Refer to **Figure 16-29**.)

The receive history list is also updated just before the storage process. In addition, during storage process (MUC = 1), the RDY bit of the C0MCTRL register of the message buffer is locked to avoid the coincidental data WR by CPU. Note the storage process may be disturbed (delayed) when the CPU accesses the message buffer.

Figure 16-29. DN and MUC Bit Setting Period (for Standard ID Format)



Remark m = 0 to 15

16.9.3 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 23 messages, the last in-message pointer (LIPT) with the corresponding C0LIPT register and the receive history list get pointer (RGPT) with the corresponding C0RGPT register.

The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The C0LIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the C0LIPT register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The LIPT pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. Each time recording to the RHL has been completed, the LIPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the C0RGPT register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the C0RGPT register, the RGPT pointer is automatically incremented.

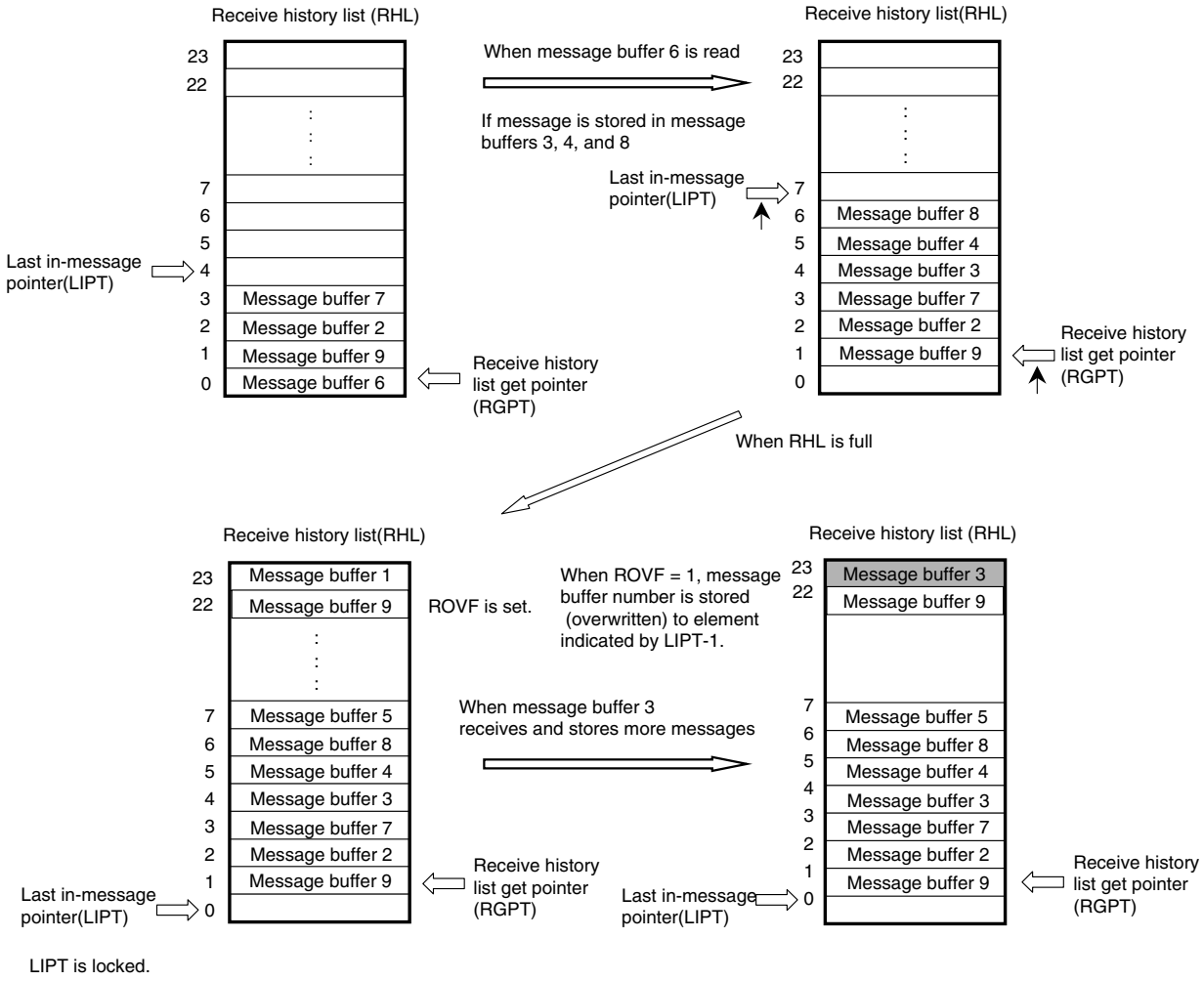
If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the C0RGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer is incremented and because its value no longer matches the value of the RGPT pointer, the RHPM bit is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the LIPT pointer is incremented and matches the value of the RGPT pointer minus 1, the ROVF bit (receive history list overflow) of the C0RGPT register is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the new message. In this case, after the ROVF bit has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However messages itself are not lost and can be located by CPU search in message buffer memory with the help of the DN-bit.

Caution If the history list is in the overflow condition (ROVF is set), reading the history list contents is still possible, until the history list is empty (indicated by RHPM flag set). Nevertheless, the history list remains in the overflow condition, until ROVF is cleared by software. If ROVF is not cleared, the RHPM flag will also not be updated (cleared) upon a message storage of newly received frame. This may lead to the situation, that RHPM indicates an empty history list, although a reception has taken place, while the history list is in the overflow state (ROVF and RHPM are set).

As long as the RHL contains 23 or less entries the sequence of occurrence is maintained. If more receptions occur without reading the RHL by the host processor, complete sequence of receptions can not be recovered.

Figure 16-30. Receive History List



16.9.4 Mask function

For any message buffer, which is used for reception, the assignment to one of four global reception masks (or no mask) can be selected.

By using the mask function, the message ID comparison can be reduced by masked bits, herewith allowing the reception of several different IDs into one buffer.

While the mask function is in effect, an identifier bit that is defined to be "1" by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.

However, this comparison is performed for any bit whose value is defined as "0" by the mask.

For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are "0" and bits ID24 and ID22 are "1", are to be stored in message buffer 14. The procedure for this example is shown below.

<1> Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

x = don't care

<2> Identifier to be configured in message buffer 14 (example)
(using CANn message ID registers L14 and H14 (C0MIDL14 and C0MIDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x

ID6	ID5	ID4	ID3	ID2	ID1	ID0
x	x	x	x	x	x	x

ID with ID27 to ID25 cleared to "0" and ID24 and ID22 set to "1" is registered (initialized) to message buffer 14.

Remark Message buffer 14 is set as a standard format identifier that is linked to mask 1 (MT[2:0] of C0MCONF14 register are set to 010B).

<3> Mask setting for CAN module 1 (mask 1) (Example)

(Using CAN1 address mask 1 registers L and H (C1MASKL1 and C1MASKH1))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1

CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1

CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
1	1	1	1	1	1	1

Not compared (masked)

0: Compared

The CMID27 to CMID24 and CMID22 bits are cleared to "0", and CMID28, CMID23, and CMID21 to CMID0 bits are set to "1".

16.9.5 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type.

Suppose, for example, the same message buffer type is set to 5 message buffers, message buffers 10 to 14, and the same ID is set to each message buffer. If the first message whose ID matches the ID of the message buffers is received, it is stored in message buffer 10. At this point, the DN bit of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

If the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and 14. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting the IE bit of the C0MCTRLm register of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. The IE bit in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and the IE bit in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing the IE bit of message buffers 0 to (k-3) and setting the IE bit of message buffer k-2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Cautions 1. MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.

2. MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.

3. MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.

4. With MBRB, "matching ID" means "matching ID after mask". Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.

5. The priority between MBRBs is mentioned in 16.9.1 Message Reception.

Remark m = 0 to 15

16.9.6 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer
  (MA0 bit of C0MCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT[2:0] bits in C0MCONFm register set to 000B)
- Ready for reception
  (RDY bit of C0MCTRLm register set to 1.)
- Set to transmit message
  (RTR bit of C0MCONFm register is cleared to 0.)
- Transmission request is not set.
  (TRQ bit of C0MCTRLm register is cleared to 1.)

Upon acceptance of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The MDLC[3:0] bit string in the C0MDLCm register stores the received DLC value.
- C0MDATA0m to C0MDATA7m in the data area are not updated (data before reception is saved).
- The DN bit of the C0MCTRLm register is set to 1.
- The CINTS1 bit of the C0INTS register is set to 1 (if the IE bit in the C0MCTRLm register of the message buffer that receives and stores the frame is set to 1).
- The reception completion interrupt (INTC0REC) is output (if the IE bit in the C0MCTRLm register of the message buffer that receives and stores the frame is set to 1 and if the CIE1 bit of the C0IE register is set to 1).
- The message buffer number is recorded to the receive history list.

Caution When a message buffer is searched for receiving and storing a remote frame, overwrite control by the OWS bit of the C0MCONFm register of the message buffer and the DN bit of the C0MCTRLm register are not affected. The setting of OWS is ignored, and DN is set in any case. If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.

Remark m = 0 to 15

16.10 Message Transmission

16.10.1 Message transmission

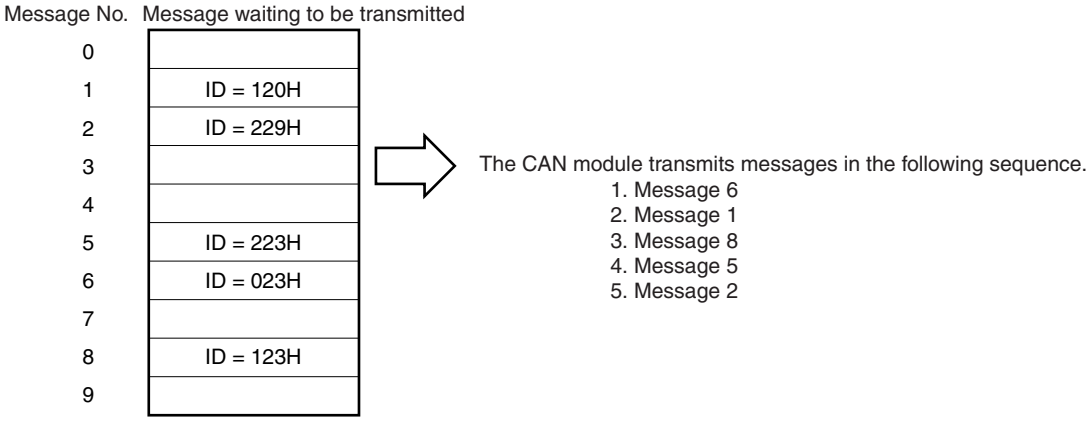
In all the operation modes, if the TRQ bit is set to 1 in a message buffer that satisfies the following conditions, the message buffer that is to transmit a message is searched.

- Used as a message buffer
  (MA0 bit of C0MCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT[2:0] bits of C0MCONFm register set to 000B.)
- Ready for transmission
  (RDY bit of C0MCTRLm register set to 1.)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs). To facilitate transmission processing by software when there are several messages awaiting transmission, the CAN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.

Transmission priority is controlled by the identifier (ID).

Figure 16-31. Message Processing Example



After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the TRQ bit set to 1 in advance) is transmitted.

If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. To solve this priority inversion effect, the software can perform a transmission abort request for the lower priority message. The highest priority is determined according to the following rules.

Priority	Conditions	Description
1(high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than message frame with the 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (RTR bit is cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (IDE bit is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If more than one transmission-pending extended ID message frame have equal values in the first 11 bits of the ID and the same frame type (equal RTR bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5(low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

Remarks 1. If automatic block transmission request bit ABTTRG is set to 1 in the normal operation mode with ABT, the TRQ bit is set to 1 only for one message buffer in the ABT message buffer group.

If the ABT mode was triggered by ABTTRG bit, one TRQ bit is set to 1 in the ABT area (buffer 0 through 7). Beyond this TRQ bit, the application can request transmissions (set TRQ to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with TRQ bit set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted at first.

Upon successful transmission of a message frame, the following operations are performed.

- The TRQ flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit CINTS0 of the C0INTS register is set to 1 (if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTC0TRX output (if the CIE0 bit of the C0IE register is set to 1 and if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).

2. When changing the contents of a transmit buffer, the RDY flag of this buffer must be cleared before updating the buffer contents. As during internal transfer actions, the RDY flag may be locked temporarily, the status of RDY must be checked by software, after changing it.

3. m = 0 to 15

16.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been were sent. The THL consists of storage elements equivalent to up to seven messages, the last out-message pointer (LOPT) with the corresponding C0LOPT register, and the transmit history list get pointer (TGPT) with the corresponding C0TGPT register.

The THL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The C0LOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the C0LOPT register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The LOPT pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the LOPT pointer. Each time recording to the THL has been completed, the LOPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The TGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the C0TGPT register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the C0TGPT register, the TGPT pointer is automatically incremented.

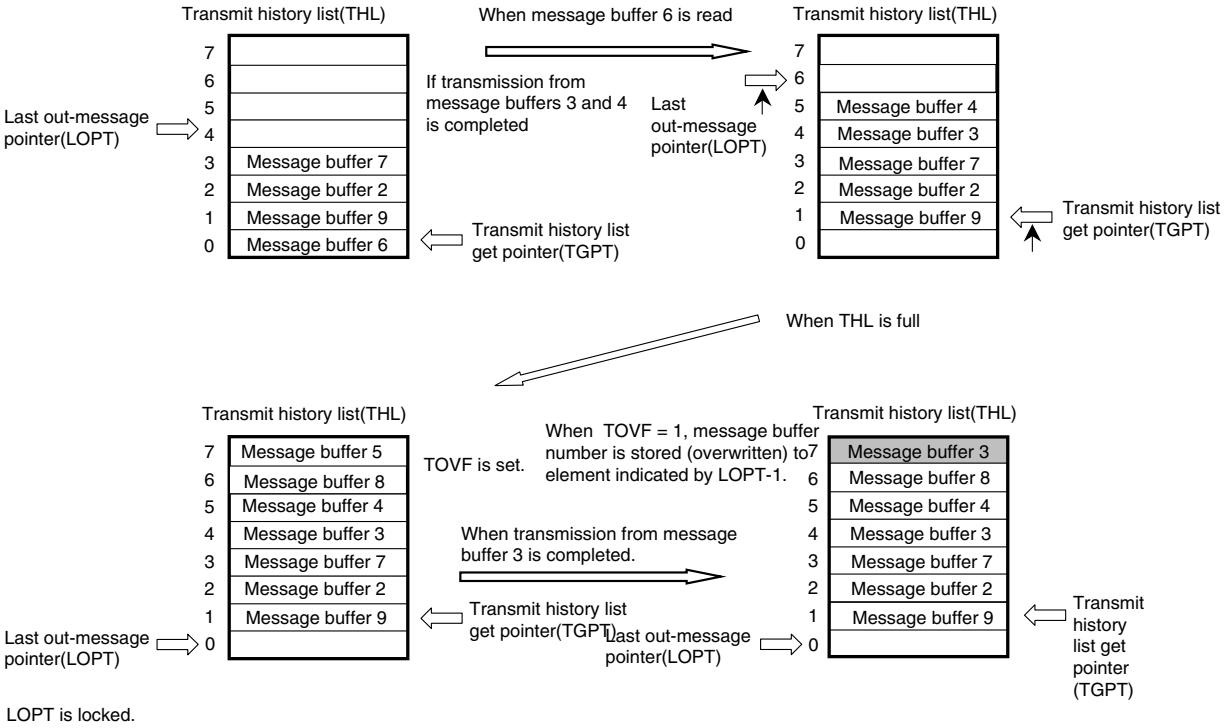
If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (transmit history list pointer match) of the C0TGPT register is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the LOPT pointer is incremented and because its value no longer matches the value of the TGPT pointer, the THPM bit is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the LOPT pointer is incremented and matches the value of the TGPT pointer minus 1, the TOVF bit (transmit history list overflow) of the C0TGPT register is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the number of the message buffer that transmitted its message afterwards. After the TOVF bit has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order. However the other transmitted messages can be found by a CPU search applied to all transmit message buffers unless the CPU has not overwritten a transmit object in one of these buffers beforehand. In total up to six transmission completions can occur without overflowing the THL.

Caution If the history list is in the overflow condition (TOVF is set), reading the history list contents is still possible, until the history list is empty (indicated by THPM flag set). Nevertheless, the history list remains in the overflow condition, until TOVF is cleared by software. If TOVF is not cleared, the THPM flag will also not be updated (cleared) upon successful transmission of a new message. This may lead to the situation, that THPM indicates an empty history list, although a successful transmission has taken place, while the history list is in the overflow state (TOVF and THPM are set).

Remark m = 0 to 15

Figure 16-32. Transmit History List



16.10.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is eight (message buffer numbers 0 to 7).

By setting OPMODE [2:0] bits of the CnCTRL register to 010B, "normal operation mode with automatic block transmission function" (hereafter referred to as ABT mode) can be selected.

To issue an ABT transmission request, define the message buffers by software first. Set the MA0 bit (1) in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting MT [2:0] bits to 000B. Be sure to set the ID for each message buffer for ABT even when the same ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the CnMIDLm and CnMIDHm registers. Set the CnMDLCm and CnMDATA0m to CnMDATA7m registers before issuing a transmission request for the ABT function.

After initialization of message buffers for ABT is finished, the RDY bit needs to be set (1). In the ABT mode, the TRQ bit does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set the ABTTRG bit to 1. Automatic block transmission is then started. When ABT is started, the TRQ bit in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 has finished, TRQ bit of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.

A delay time can be inserted by program in the interval in which the transmission request (TRQ) is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the CnGMABTD register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the CnBRP and CnBTR registers.

Among transmit objects within the ABT-area, the priority of the transmission ID is not evaluated. The data of message buffers 0 to 7 are sequentially transmitted. When transmission of the data frame from message buffer 7 has been completed, the ABTTRG bit is automatically cleared to 0 and the ABT operation is finished.

If the RDY bit of an ABT message buffer is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and the ABTTRG bit is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting the RDY and ABTTRG bits to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the ABTCLR bit to 1 while ABT mode is stopped and ABTTRG bit is cleared to 0. In this case, transmission is started from message buffer 0 if the ABTCLR bit is cleared to 0 and then the ABTTRG bit is set to 1.

An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, the IE bit of the CnMCTRLm register of each message buffer except the last message buffer needs to be cleared (0).

If a transmit message buffer other than those used by the ABT function (message buffer 8 to 15) is assigned to a transmit message buffer, the message to be transmitted next is determined by the priority of the transmission ID of the ABT message buffer whose transmission is currently held pending and the transmission ID of the message buffers other than those used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Cautions 1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0 in order to resume ABT operation at buffer No.0. If the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1, the subsequent operation is not guaranteed.

2. If the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared immediately after the processing of the clearing request is completed.

3. Do not set the ABTTRG bit in the initialization mode. If the ABTTRG bit is set in the initialization mode, the proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.

4. Do not set TRQ bit of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, the operation is not guaranteed.

5. The C0GMABTD register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of the TRQ bit for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the status of transmission from other stations and the status of the setting of the transmission request for messages other than the ABT messages (message buffer 8 to 15).

6. If a transmission request is made for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (C0GMABTD = 00H), messages other than ABT messages may be transmitted not depending on the priority of the ABT message.

7. Do not clear the RDY bit to 0 when ABTTRG = 1.

8. If a message is received from another node while normal operation mode with ABT is active, the TX-message from the ABT-area may be transmitted with delay of one frame although CnGMABTD register was set up with 00H.

Remark m = 0 to 15

16.10.4 Transmission abort process

(1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)

The user can clear the TRQ bit of the C0MCTRLm register to 0 to abort a transmission request. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the C0CTRL register and the C0TGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 16-47**).

(2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)

The user can clear the ABTTRG bit of the C0GMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the C0GMABT register = 0, clear the TRQ bit of the C0MCTRLm register to 0. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the C0CTRL register and the C0TGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 16-48**).

(3) Transmission abort process for ABT transmission in normal operation mode with automatic block transmission (ABT)

To abort ABT that is already started, clear the ABTTRG bit of the C0GMABT register to 0. In this case, the ABTTRG bit remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of the TRQ bit in the last transmitted message buffer. If the TRQ bit is set to 1 when clearing the ABTTRG bit is requested, the internal ABT pointer points to the last transmitted message buffer (for details, refer to the process in **Figure 16-49**). If the TRQ bit is cleared to 0 when clearing the ABTTRG bit is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area (for details, refer to the process in **Figure 16-50**).

Caution Be sure to abort ABT by clearing ABTTRG to 0. The operation is not guaranteed if aborting transmission is requested by clearing RDY bit.

When the normal operation mode with ABT is resumed after ABT has been aborted and ABTTRG bit is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Status of TRQ of ABT Message Buffer	Abort After Successful Transmission	Abort after erroneous transmission
Set (1)	Next message buffer in the ABT area[Note]	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area[Note]	Next message buffer in the ABT area[Note]

Note The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while ABT of message buffer 7 is in progress is regarded as completion of ABT, rather than abort, if transmission of message buffer 7 has been successfully completed, even if ABTTRG is cleared to 0. If the RDY bit in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if ABTTRG is set to 1, and ABT ends immediately.

Remark m = 0 to 15

16.10.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via the RTR bit of the C0MCONFm register. Setting (1) the RTR bit sets remote frame transmission.

Remark m = 0 to 15

16.11 Power Save Modes

16.11.1 CAN sleep mode

The CAN sleep mode can be used to set the CAN controller to standby mode in order to reduce power consumption. The CAN module can enter the CAN sleep mode from all operation modes. Release of the CAN sleep mode returns the CAN module to exactly the same operation mode from which the CAN sleep mode was entered.

In the CAN sleep mode, the CAN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering CAN sleep mode

The CPU issues a CAN sleep mode transition request by writing 01B to the PSMODE[1:0] bits of the C0CTRL register.

This transition request is only acknowledged only under the following conditions.

- The CAN module is already in one of the following operation modes
  - Normal operation mode
  - Normal operation mode with ABT
  - Receive-only mode
  - Single-shot mode
  - Self-test mode
  - CAN stop mode in all the above operation modes
- The CAN bus state is bus idle (the 4th bit in the interframe space is recessive)[Note]
- No transmission request is pending

Note If the CAN bus is fixed to dominant, the request for transition to the CAN sleep mode is held pending. Also the transition from CAN stop mode to CAN sleep mode is independent of the CAN bus state.

Remark If a sleep mode request is pending, and at the same time a message is received in a message box, the sleep mode request is not cancelled, but is executed right after message storage has been finished. This may result in AFCAN being in sleep mode, while the CPU would execute the RX interrupt routine. Therefore, the interrupt routine must check the access to the message buffers as well as reception history list registers by using the MBON flag, if sleep mode is used.

If any one of the conditions mentioned above is not met, the CAN module will operate as follows.

- If the CAN sleep mode is requested from the initialization mode, the CAN sleep mode transition request is ignored and the CAN module remains in the initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when the CAN sleep mode is requested in one of the operation modes, immediate transition to the CAN sleep mode is not possible. In this case, the CAN sleep mode transition request is held pending until the CAN bus state becomes bus idle (the 4th bit in the interframe space is recessive). In the time from the CAN sleep mode request to successful transition, the PSMODE [1:0] bits remain 00B. When the module has entered the CAN sleep mode, PSMODE [1:0] bits are set to 01B.
- If a request for transition to the initialization mode and a request for transition to the CAN sleep are made at the same time while the CAN module is in one of the operation modes, the request for the initialization mode is enabled. The CAN module enters the initialization mode at a predetermined timing. At this time, the CAN sleep mode request is not held pending and is ignored.

- Even when initialization mode and sleep mode are not requested simultaneously (i.e the first request has not been granted while the second request is made), the request for initialization has priority over the sleep mode request. The sleep mode request is cancelled when the initialization mode is requested. When a pending request for initialization mode is present, a subsequent request for Sleep mode request is cancelled right at the point in time where it was submitted.

(2) Status in CAN sleep mode

The CAN module is in one of the following states after it enters the CAN sleep mode.

- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the CAN reception pin (CRxD) remains in effect to wake up the CAN module from the CAN bus.
- To wake up the CAN module from the CPU, data can be written to PSMODE [1:0] of the CAN module control register (C0CTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for C0LIPT, C0RGPT, C0LOPT, and C0TGPT.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN Global Control register (C0GMCTRL) is cleared.
- A request for transition to the initialization mode is not acknowledged and is ignored.

(3) Releasing CAN sleep mode

The CAN sleep mode is released by the following events.

- When the CPU writes 00B to the PSMODE [1:0] bits of the C0CTRL register
- A falling edge at the CAN reception pin (CRxD) (i.e. the CAN bus level shifts from recessive to dominant)

Caution Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock to the CAN while the CAN was in sleep mode, even subsequently the CAN sleep mode will not be released and PSMODE [1:0] will continue to be 01B unless the clock to the CAN is supplied again. In addition to this, the receive message will not be received after that.

After releasing the sleep mode, the CAN module returns to the operation mode from which the CAN sleep mode was requested and the PSMODE [1:0] bits of the C0CTRL register are reset to 00B. If the CAN sleep mode is released by a change in the CAN bus state, the CINTS5 bit of the C0INTS register is set to 1, regardless of the CIE bit of the C0IE register. After the CAN module is released from the CAN sleep mode, it participates in the CAN bus again by automatically detecting 11 consecutive recessive-level bits on the CAN bus. The user application has to wait until MBON = 1, before accessing message buffers again.
When a request for transition to the initialization mode is made while the CAN module is in the CAN sleep mode, that request is ignored; the CPU has to be released from sleep mode by software first before entering the initialization mode.

Caution Be aware that the release of CAN sleep mode by CAN bus event, and thus the wake up interrupt may happen at any time, even right after requesting sleep mode, if a CAN bus event occurs.

Remark m = 0 to 15

16.11.2 CAN stop mode

The CAN stop mode can be used to set the CAN controller to standby mode to reduce power consumption. The CAN module can enter the CAN stop mode only from the CAN sleep mode. Release of the CAN stop mode puts the CAN module in the CAN sleep mode.

The CAN stop mode can only be released (entering CAN sleep mode) by writing 01B to the PSMODE [1:0] bits of the C0CTRL register and not by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

(1) Entering CAN stop mode

A CAN stop mode transition request is issued by writing 11B to the PSMODE [1:0] bits of the C0CTRL register.

A CAN stop mode request is only acknowledged when the CAN module is in the CAN sleep mode. In all other modes, the request is ignored.

Caution To set the CAN module to the CAN stop mode, the module must be in the CAN sleep mode. To confirm that the module is in the sleep mode, check that PSMODE [1:0] = 01B, and then request the CAN stop mode. If a bus change occurs at the CAN reception pin (CRxD) while this process is being performed, the CAN sleep mode is automatically released. In this case, the CAN stop mode transition request cannot be acknowledged.

(2) Status in CAN stop mode

The CAN module is in one of the following states after it enters the CAN stop mode.

- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the CAN module from the CPU, data can be written to PSMODE [1:0] of the CAN module control register (C0CTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for C0LIPT, C0RGPT, C0LOPT, and C0TGPT.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN Global Control register (C0GMCTRL) is cleared.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Releasing CAN stop mode

The CAN stop mode can only be released by writing 01B to the PSMODE [1:0] bits of the C0CTRL register. After releasing the CAN stop mode, the CAN module enters the CAN sleep mode.

When the initialization mode is requested while the CAN module is in the CAN stop mode, that request is ignored; the CPU has to release the stop mode and subsequently CAN sleep mode before entering the initialization mode. It is impossible to enter the other operation mode directly from the CAN stop mode not entering the CAN sleep mode, that request is ignored.

Remark m = 0 to 15

16.11.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the CAN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example of using the power saving modes.

First, put the CAN module in the CAN sleep mode (PSMODE = 01B). Next, put the CPU in the power saving mode. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRxD) in this status, the CINTS5 bit in the CAN module is set to 1. If the CIE5 bit of the C0CTRL register is set to 1, a wakeup interrupt (INTC0WUP) is generated. The CAN module is automatically released from the CAN sleep mode (PSMODE = 00B) and returns to the normal operation mode. The CPU, in response to INTC0WUP, can release its own power saving mode and return to the normal operation mode.

To further reduce the power consumption of the CPU, the internal clocks, including that of the CAN module, may be stopped. In this case, the operating clock supplied to the CAN module is stopped after the CAN module is put in the CAN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRxD) in this status, the CAN module can set the CINTS5 bit to 1 and generate the wakeup interrupt (INTC0WUP) even if it is not supplied with the clock. The other functions, however, do not operate because clock supply to the CAN module is stopped, and the module remains in the CAN sleep mode. The CPU, in response to INTC0WUP, releases its power saving mode, resumes supply of the internal clocks, including the clock to the CAN module, after the oscillation stabilization time has elapsed, and starts instruction execution. The CAN module is immediately released from the CAN sleep mode when clock supply is resumed, and returns to the normal operation mode (PSMODE = 00B).

16.12 Interrupt Function

The CAN module provides 6 different interrupt sources.

The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 16-20. List of CAN Module Interrupt Sources

No.	Interrupt Status Bit		Interrupt Enable Bit		Interrupt Request Signal	Interrupt Source Description
	Name	Register	Name	Register		
1	CINTS0[Note]	C0INTS	CIE0[Note]	C0IE	INTC0TRX	Message frame successfully transmitted from message buffer m
2	CINTS1[Note]	C0INTS	CIE1[Note]	C0IE	INTC0REC	Valid message frame reception in message buffer m
3	CINTS2	C0INTS	CIE2	C0IE	INTC0ERR	CAN module error state interrupt (Supplement 1)
4	CINTS3	C0INTS	CIE3	C0IE		CAN module protocol error interrupt (Supplement 2)
5	CINTS4	C0INTS	CIE4	C0IE		CAN module arbitration loss interrupt
6	CINTS5	C0INTS	CIE5	C0IE	INTC0WUP	CAN module wakeup interrupt from CAN sleep mode (Supplement 3)

Note The IE bit (message buffer interrupt enable bit) in the C0MCTRL register of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

Supplements 1. This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.

2. This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.
3. This interrupt is generated when the CAN module is woken up from the CAN sleep mode because a falling edge is detected at the CAN reception pin (CAN bus transition from recessive to dominant).

Remark m = 0 to 15

16.13 Diagnosis Functions and Special Operational Modes

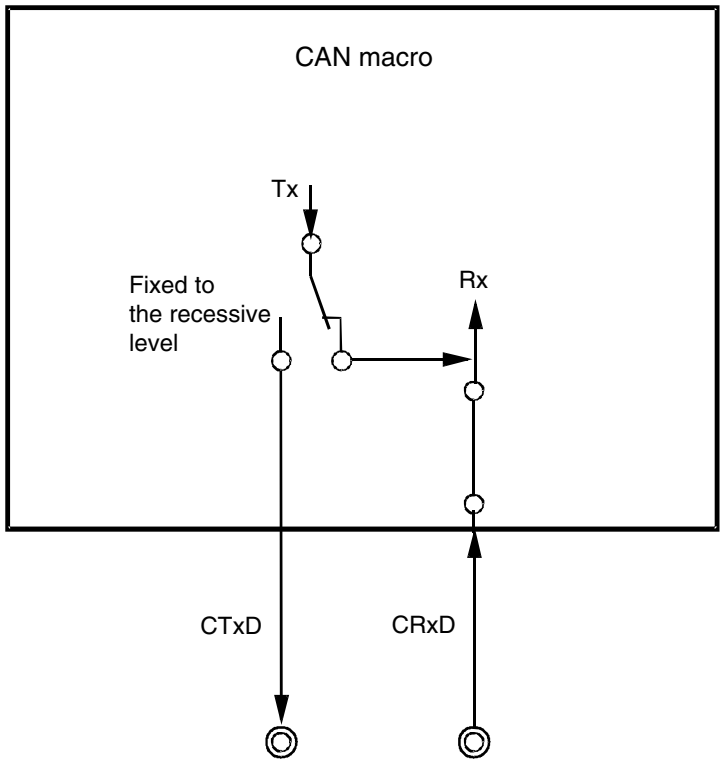
The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of specific CAN communication methods.

16.13.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.

For example, this mode can be used for automatic baud-rate detection. The baud rate in the CAN module is changed until "valid reception" is detected, so that the baud rates in the module match ("valid reception" means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting the VALID bit of the C0CTRL register (1).

Figure 16-33. CAN Module Terminal Connection in Receive-Only Mode



In the receive-only mode, no message frames can be transmitted from the CAN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.

In the receive-only mode, the CAN transmission pin (CTxD) in the CAN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the CAN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the CAN module, the transmission error counter TEC is never updated. Therefore, a CAN module in the receive-only mode does not enter the bus-off state.

Furthermore, ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the VALID bit is set to 1 for the first time.

16.13.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.). All other behavior of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.

The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of the AL bit of the C0CTRL register. When the AL bit is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If the AL bit is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, the TRQ bit in a message buffer defined as a transmit message buffer is cleared to 0 by the following events.

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking the CINTS4 and CINTS3 bits of the C0INTS register respectively, and the type of the error can be identified by reading the LEC[2:0] bits of the C0LEC register.

Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the C0INTS register is set to 1. If the CIE0 bit of the C0IE register is set to 1 at this time, an interrupt request signal is output.

The single-shot mode can be used when emulating time-triggered communication methods (e.g. TTCAN level 1).

Caution The AL bit is only valid in Single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

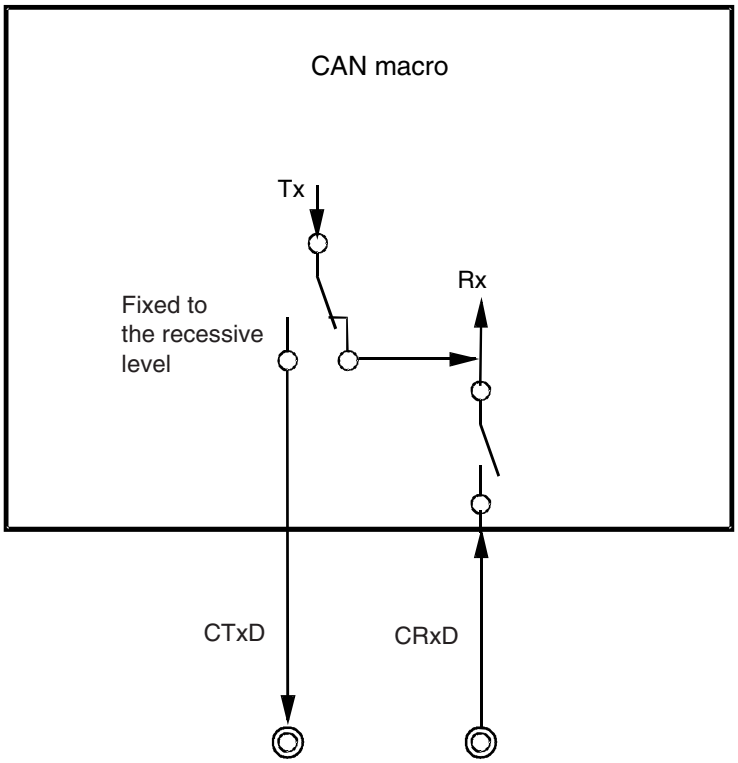
16.13.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the CAN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The CAN transmission pin (CTxD) is fixed to the recessive level.

If the falling edge on the CAN reception pin (CRxD) is detected after the CAN module has entered the CAN sleep mode from the self-test mode, however, the module is released from the CAN sleep mode in the same manner as the other operation modes. To keep the module in the CAN sleep mode, use the CAN reception pin (CRxD) as a port pin.

Figure 16-34. CAN Module Terminal Connection in Self-test Mode



16.13.4 Receive/Transmit Operation in Each Operation Mode

Table 16-21 shows outline of the receive/transmit operation in each operation mode.

Table 16-21. Outline of the Receive/Transmit in Each Operation Mode

Operation Mode	Transmission of data/ remote frame	Transmission of ACK	Transmission of error/ overload frame	Transmission retry	Automatic Block Transmission (ABT)	Set of VALID bit	Store Data to message buffer
Initialization Mode	No	No	No	No	No	No	No
Normal Operation Mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal Operation Mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive-only mode	No	No	No	No	No	Yes	Yes
Single-shot Mode	Yes	Yes	Yes	No Note 1	No	Yes	Yes
Self-test Mode	Yes Note 2	Yes Note 2	Yes Note 2	Yes Note 2	No	Yes Note 2	Yes Note 2

Notes 1. When the arbitration lost occurs, control of re-transmission is possible by the AL bit of C0CTRL register.

2. Each signals are not generated to outside, but generated into the CAN module.

16.14 Time Stamp Function

CAN is an asynchronous, serial protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).

In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

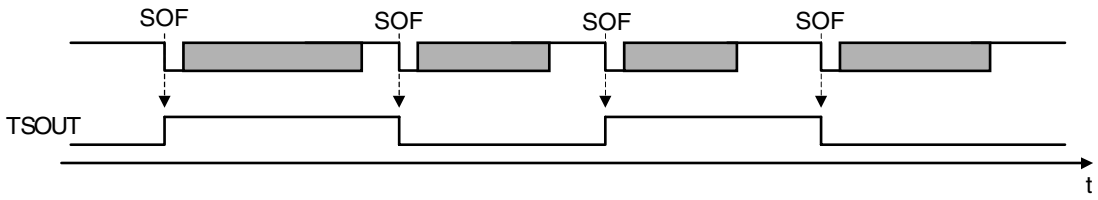
16.14.1 Time stamp function

The CAN controller supports the capturing of timer values triggered by a specific frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. TSOUT signal can be selected from the following two event sources and is specified by the TSSEL bit of the C0TS register.

- SOF event (start of frame) (TSSEL = 0)
- EOF event (last bit of end of frame) (TSSEL = 1)

The TSOUT signal is enabled by setting the TSEN bit of the C0TS register to 1.

Figure 16-35. Timing Diagram of Capture Signal TSOUT



TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in the above timing diagram, the SOF is used as the trigger event source). To capture a timer value by using TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

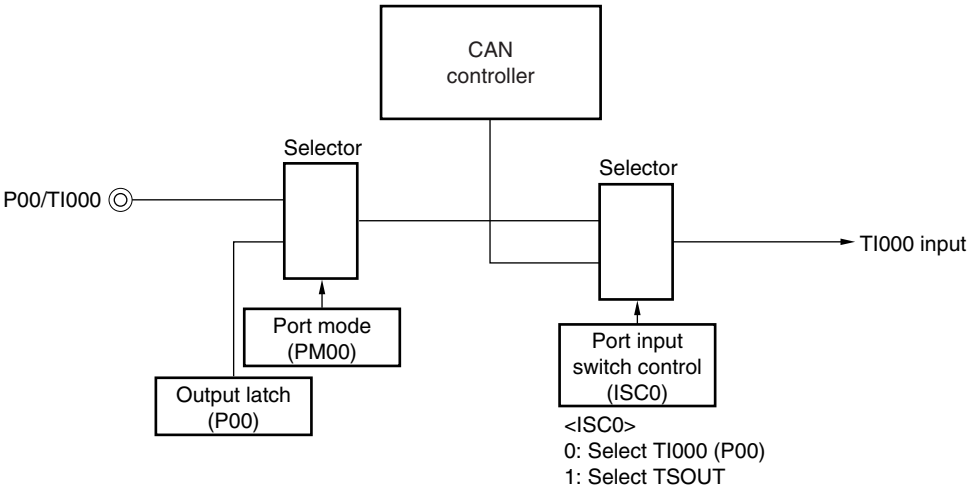
This time stamp function is controlled by the TSLOCK bit of the C0TS register. When TSLOCK is cleared to 0, TSOUT bit toggles upon occurrence of the selected event. If TSLOCK bit is set to 1, TSOUT toggles upon occurrence of the selected event, but the toggle is stopped as the TSEN bit is automatically cleared to 0 as soon as the message storing to the message buffer 0 starts. This suppresses the subsequent toggle occurrence by TSOUT, so that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.

Caution The time stamp function using TSLOCK bit is to stop toggle of TSOUT bit by receiving a data frame in message buffer 0. Therefore, message buffer 0 must be set as a receive message buffer. Since a receive message buffer cannot receive a remote frame, toggle of TSOUT bit cannot be stopped by reception of a remote frame. Toggle of TSOUT bit does not stop when a data frame is received in a message buffer other than message buffer 0.

For these reasons, a data frame cannot be received in message buffer 0 when the CAN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer. In this operation mode, therefore, the function to stop toggle of TSOUT bit by TSLOCK bit cannot be used.

The input source of the timer value according to a trigger signal (TSOUT) can be input to the 16-bit timer/event counter 00 by port input switch control (ISC0), without connectingTI000, externally.

Figure 16-36. Port Input Switch Control



Remark ISC0: Bit 0 of the input switch control register (ISC) (see **Figure 14-19**)

16.15 Baud Rate Settings

16.15.1 Baud rate settings

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN controller, as follows.

(a) 5TQ ≤ SPT (sampling point) ≤ 17 TQ
SPT = TSEG1 + 1
(b) 8 TQ ≤ DBT (data bit time) ≤ 25 TQ
DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT
(c) 1 TQ ≤ SJW (synchronization jump width) ≤ 4TQ
SJW ≤ DBT – SPT
(d) 4 ≤ TSEG1 ≤ 16 [3 (Setting value of TSEG1 [3:0] ≤ 15]
(e) 1 ≤ TSEG2 ≤ 8 [0 (Setting value of TSEG2 [2:0] ≤ 7]

Remark TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)
TSEG1 [3:0]: Bits 3 to 0 of CAN0 bit rate register (C0BTR)
TSEG2 [2:0]: Bits 10 to 8 of CAN0 bit rate register (C0BTR)

Table 16-22 shows the combinations of bit rates that satisfy the above conditions.

Table 16-22. Settable Bit Rate Combinations (1/3)

Valid Bit Rate Setting					C0BTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4

Table 16-22. Settable Bit Rate Combinations (2/3)

Valid Bit Rate Setting					C0BTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7

Table 16-22. Settable Bit Rate Combinations (3/3)

Valid Bit Rate Setting					C0BTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7[Note]	1	2	2	2	0011	001	71.4
7[Note]	1	4	1	1	0100	000	85.7
6[Note]	1	1	2	2	0010	001	66.7
6[Note]	1	3	1	1	0011	000	83.3
5[Note]	1	2	1	1	0010	000	80.0
4[Note]	1	1	1	1	0001	000	75.0

Note Setting with a DBT value of 7 or less is valid only when the value of the C0BRP register is other than 00H.

Caution The values in Table 16-22 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

16.15.2 Representative examples of baud rate settings

Tables 16-23 and **16-24** show representative examples of baud rate setting.

Table 16-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5

Caution The values in Table 16-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 16-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

Caution The values in Table 16-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 16-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5

Caution The values in Table 16-24 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 16-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (2/2)

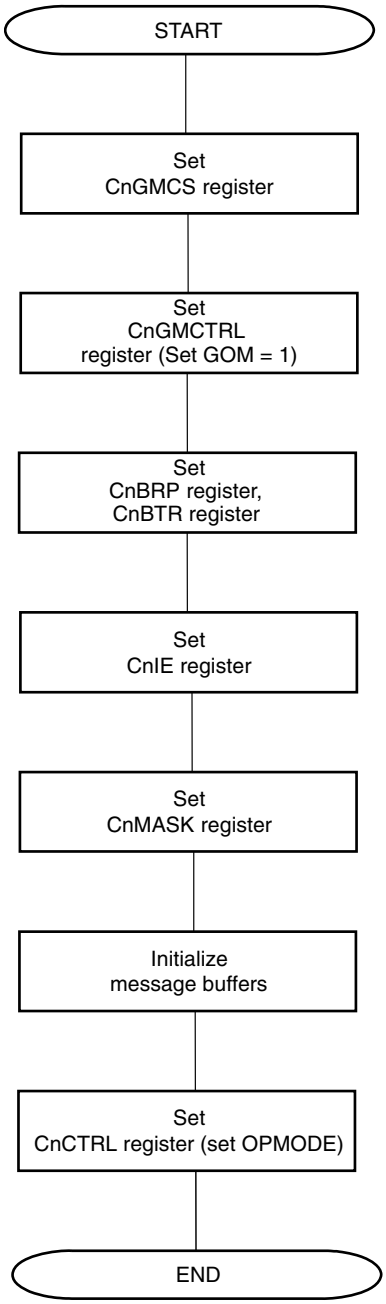
Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

Caution The values in Table 16-24 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

16.16 Operation of CAN Controller

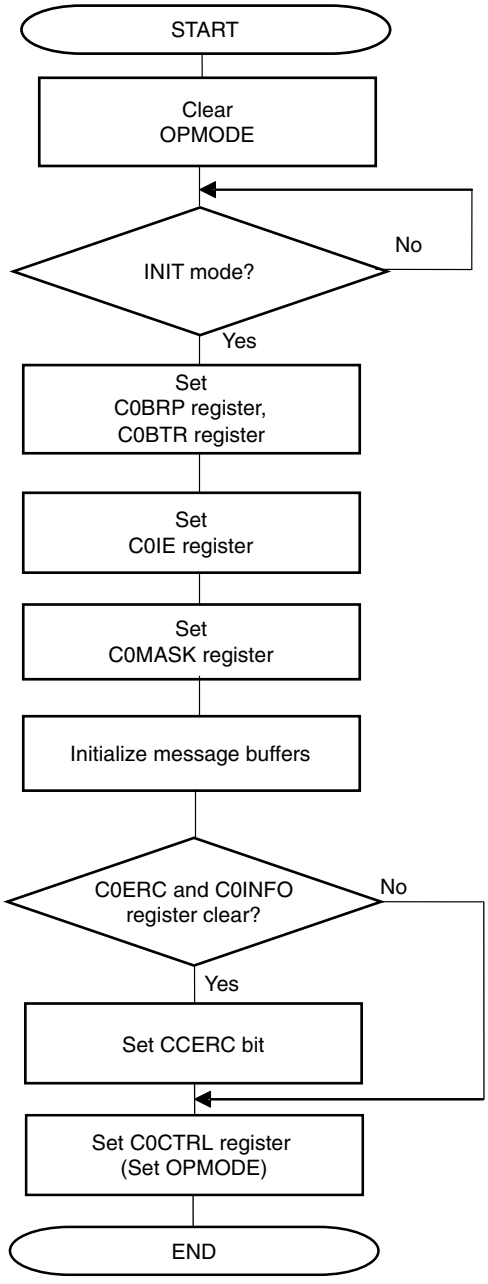
Remark m = 0 to 15

Figure 16-37. Initialization



Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

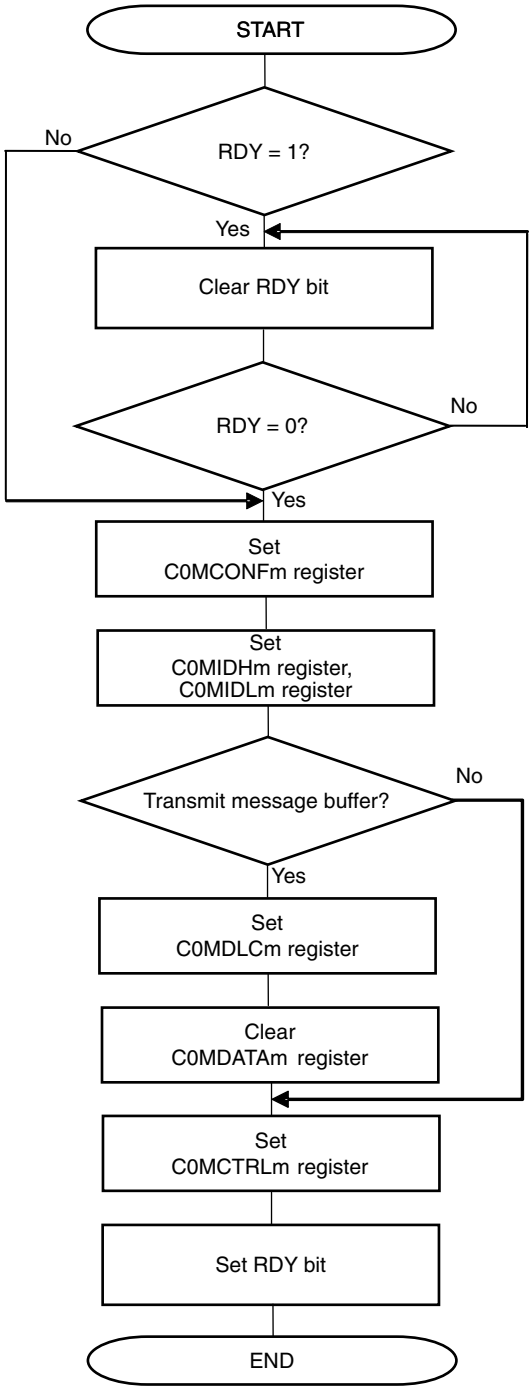
Figure 16-38. Re-initialization



Caution After setting the CAN module to the initialization mode, avoid setting the module to another operation mode immediately after. If it is necessary to immediately set the module to another operation mode, be sure to access registers other than the C0CTRL and C0GMCTRL registers (e.g. set a message buffer).

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 16-39. Message Buffer Initialization



Cautions 1. Before a message buffer is initialized, the RDY bit must be cleared.

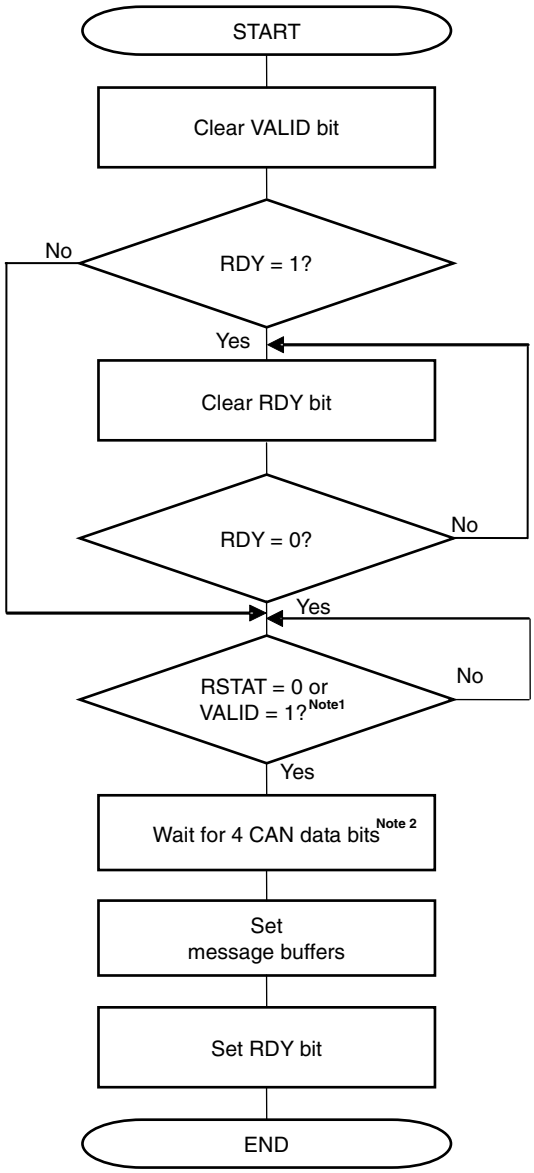
2. Make the following settings for message buffers not used by the application.

- Clear the RDY, TRQ, and DN bits of the C0MCTRLm register to 0.

- Clear the MA0 bit of the C0MCONFm register to 0.

Figure 16-40 shows the processing for a receive message buffer (MT [2:0] bits of C0MCONFm register = 001B to 101B).

Figure 16-40. Message Buffer Redefinition



Notes 1. Confirm that a message is being received because RDY bit must be set after a message is completely received.

2. Avoid message buffer redefinition during store operation of message reception by waiting additional 4 CAN data bits.

Figure 16-41 shows the processing for a transmit message buffer during transmission (MT [2:0] bits of C0MCONFm register = 000B).

Figure 16-41. Message Buffer Redefinition during Transmission

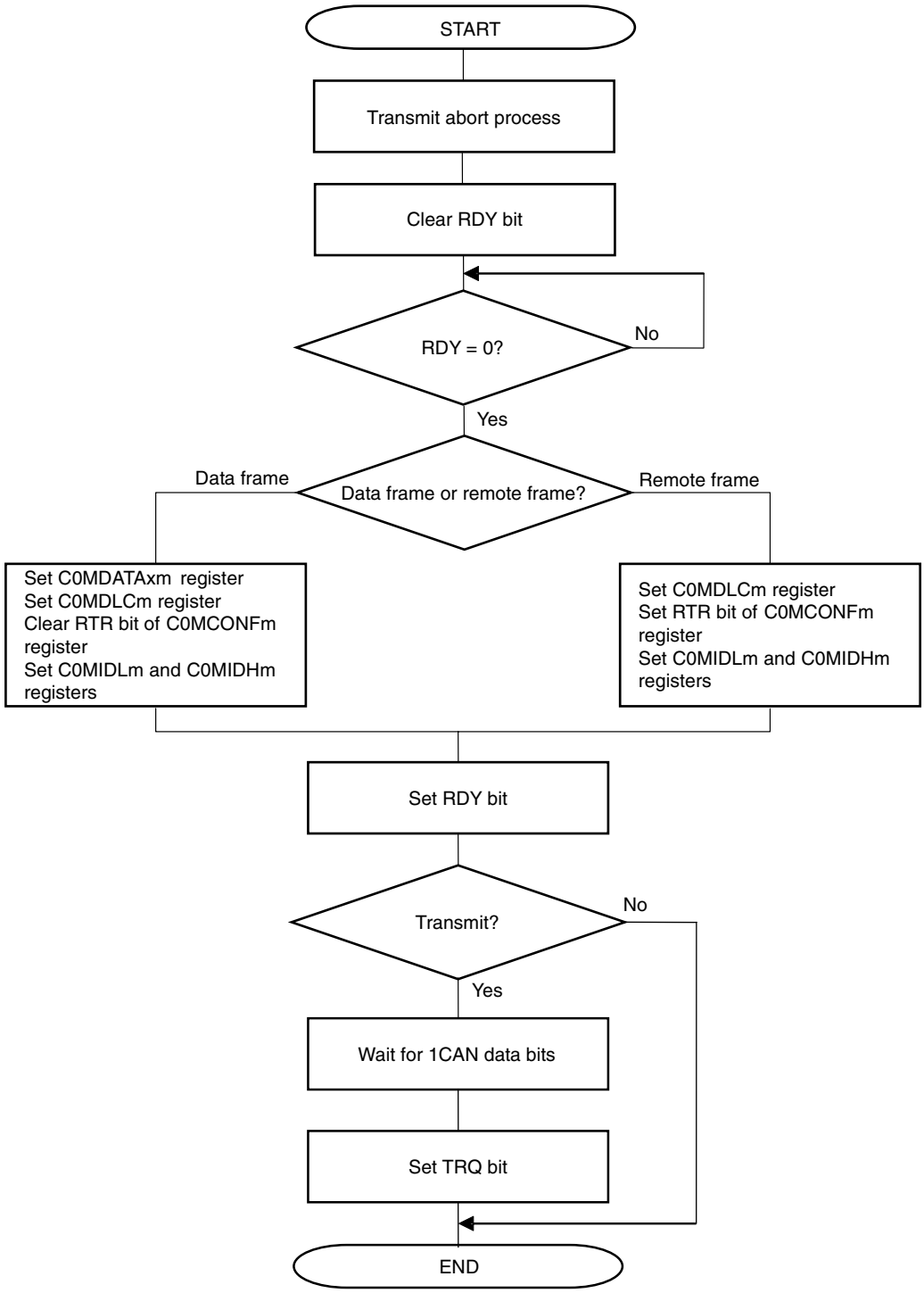
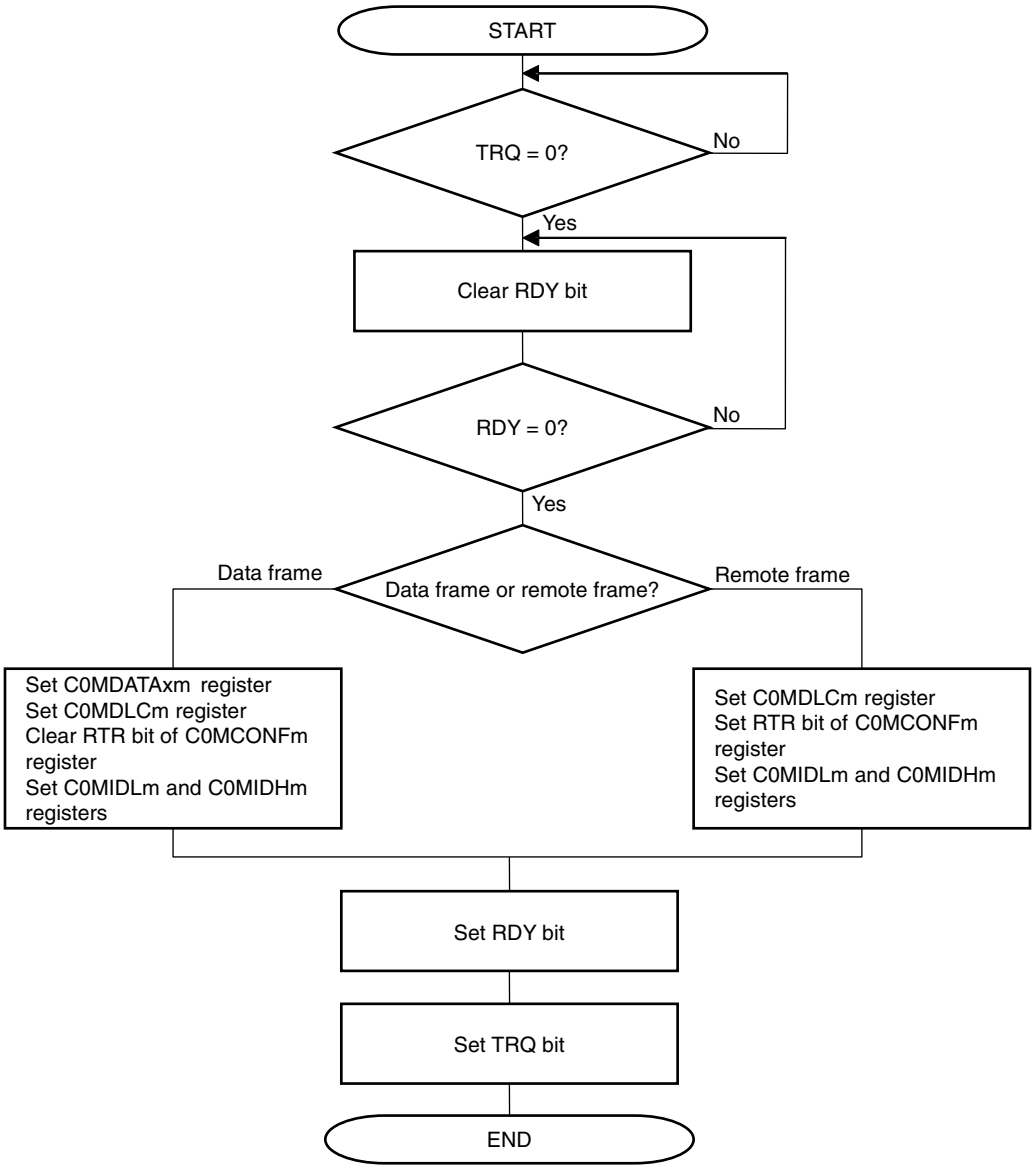


Figure 16-42 shows the processing for a transmit message buffer (MT [2:0] bits of C0MCONFm register = 000B).

Figure 16-42. Message Transmit Processing

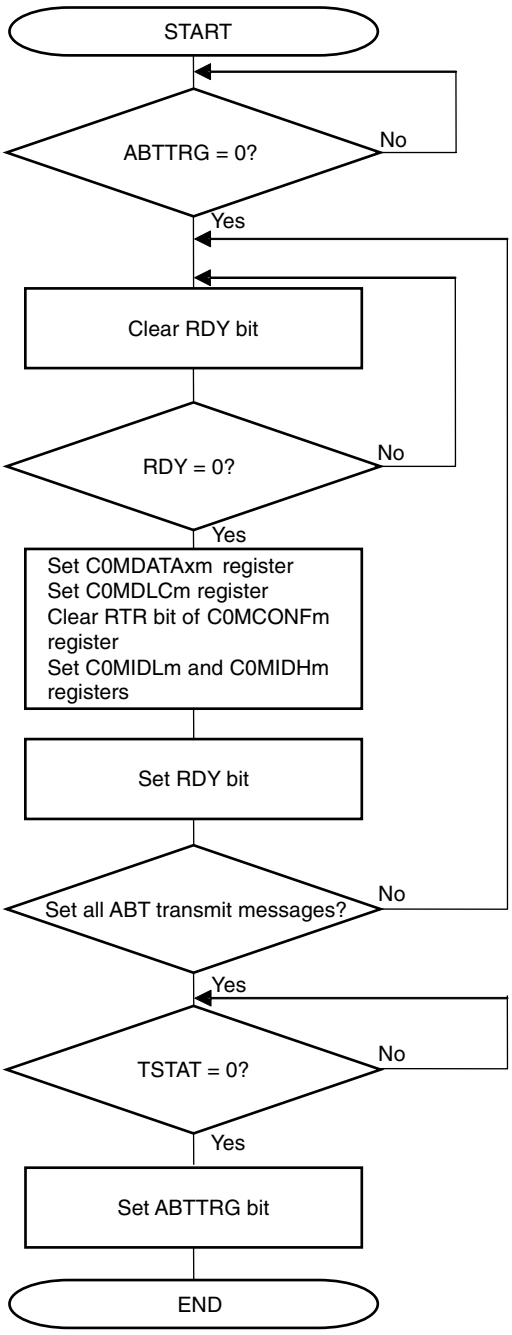


Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Figure 16-43 shows the processing for a transmit message buffer (MT [2:0] bits of C0MCONFm register = 000B).

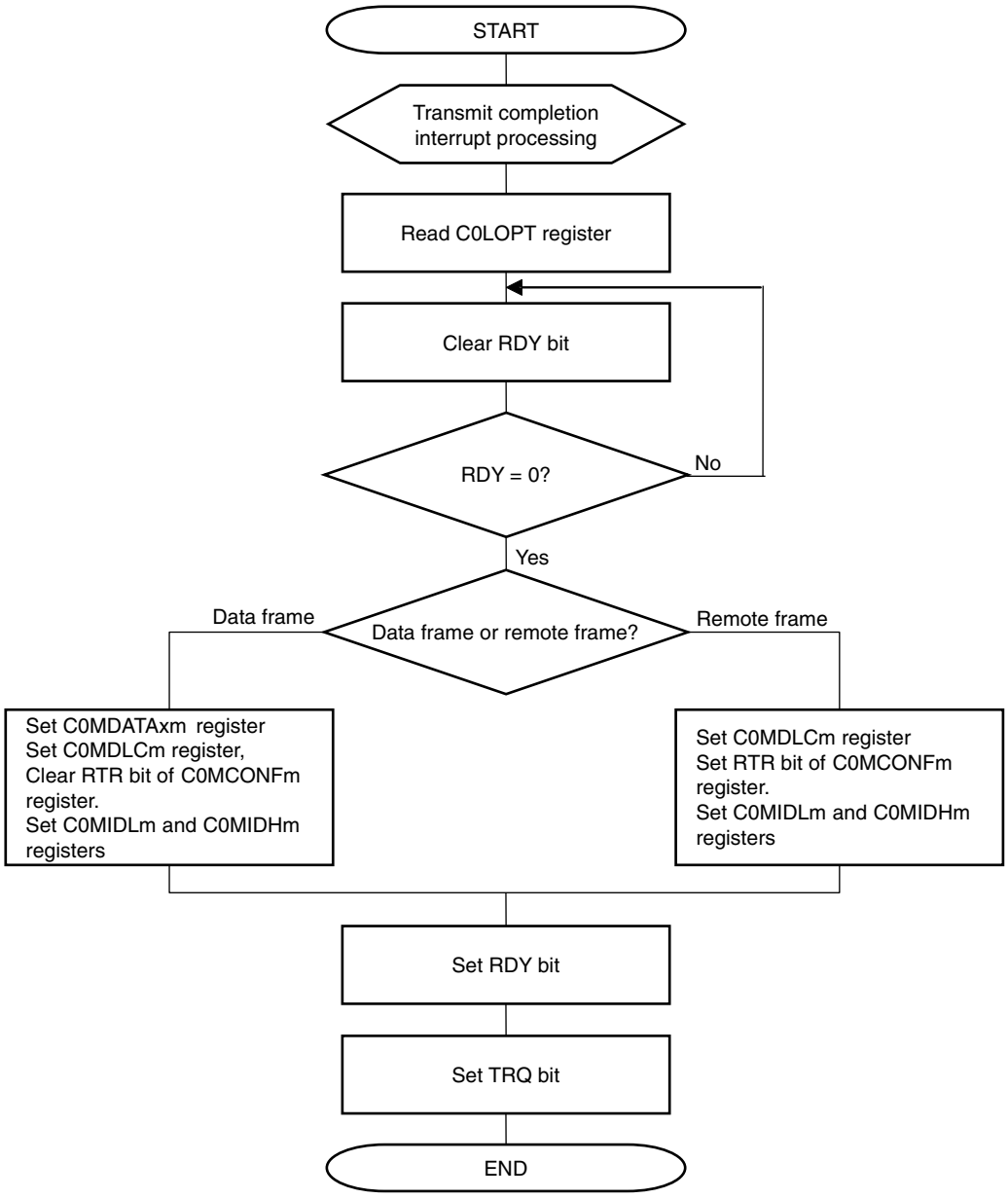
Figure 16-43. ABT Message Transmit Processing



Caution The ABTTRG bit should be set to 1 after the TSTAT bit is cleared to 0. Checking the TSTAT bit and setting the ABTTRG bit to 1 must be processed continuously.

Remark This processing (normal operation mode with ABS) can only be applied to message buffers 0 to 7. For message buffers other than the ABT message buffers, refer to **Figure 16-42**.

Figure 16-44. Transmission via Interrupt (Using C0LOPT register)

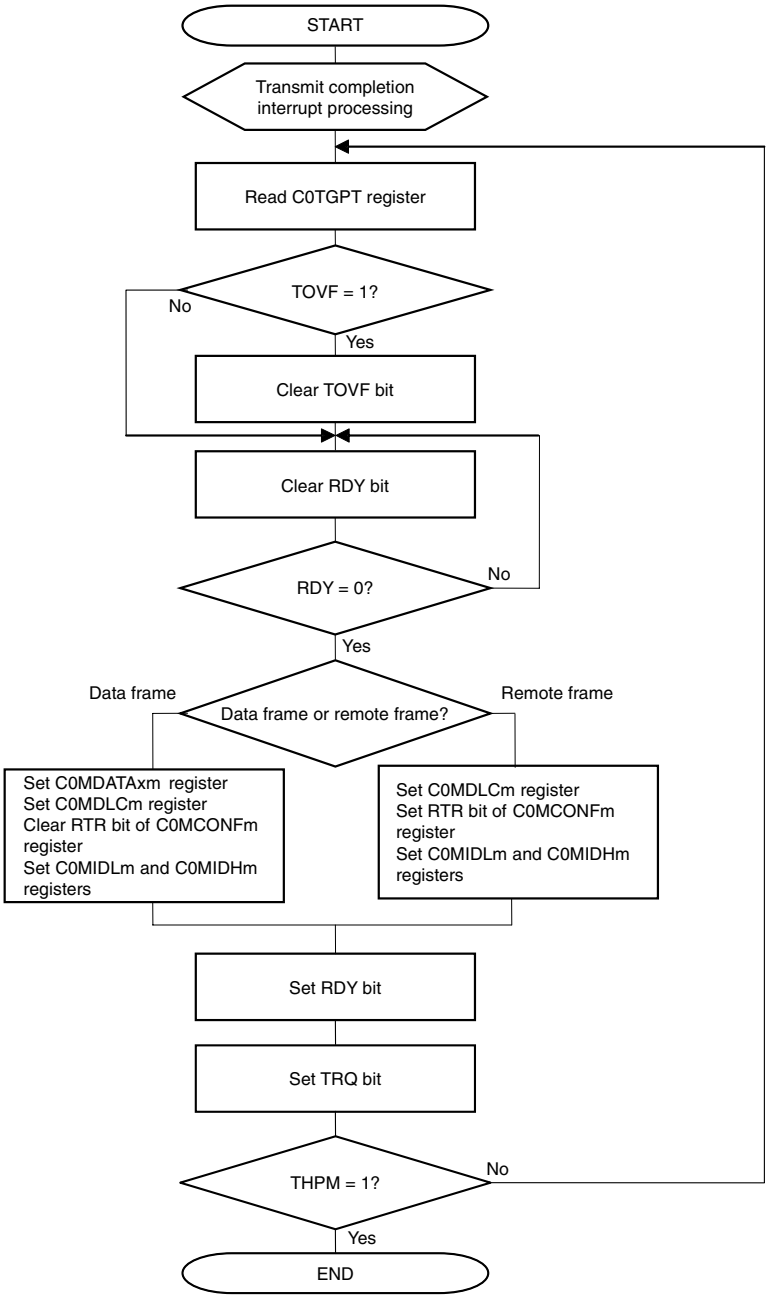


Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing TX interrupts.

Figure 16-45. Transmit via Interrupt (Using C0TGPT register)

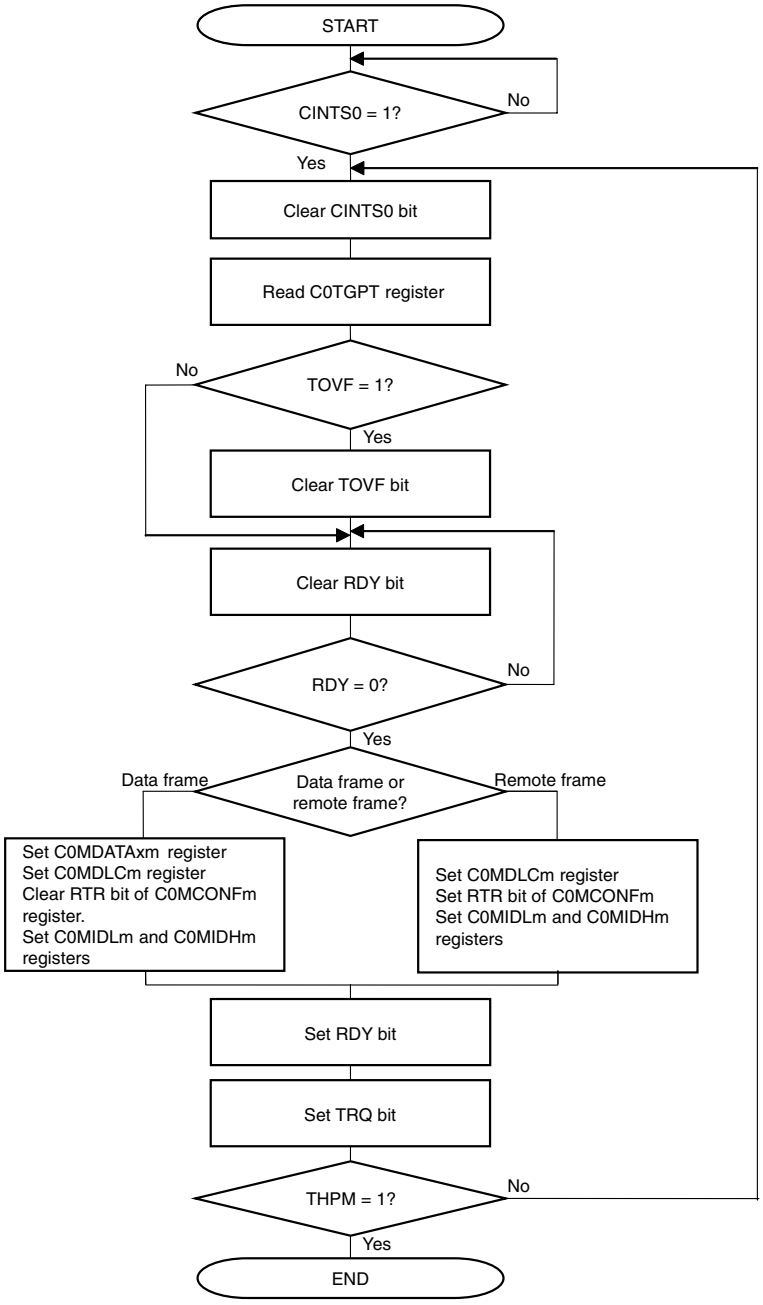


Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing TX interrupts.

Figure 16-46. Transmission via Software Polling

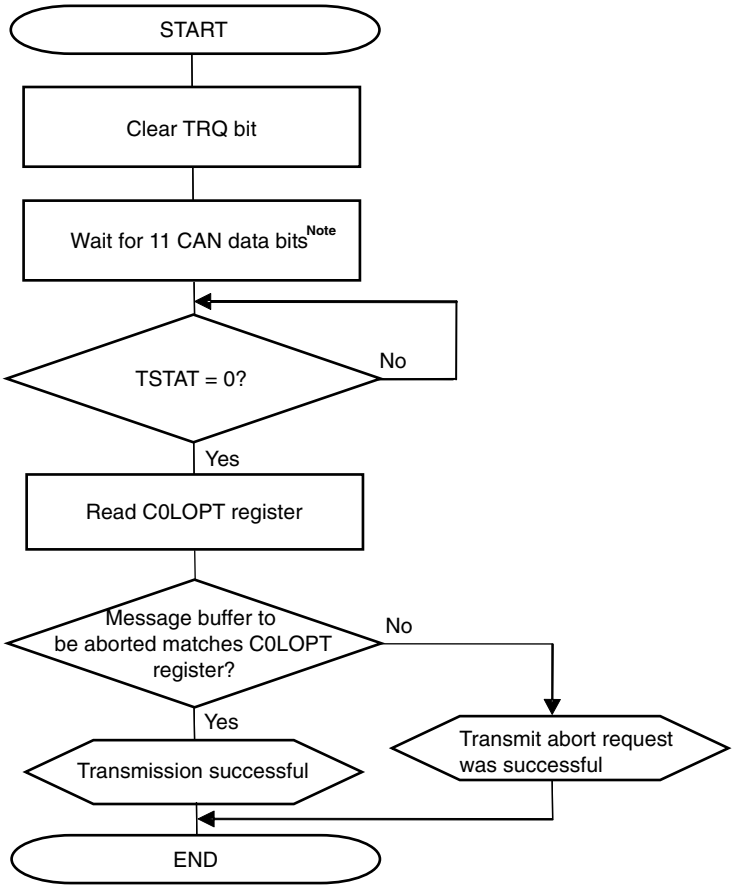


Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remark Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

Figure 16-47. Transmission Abort Processing (Except Normal Operation Mode with ABT)



Note There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

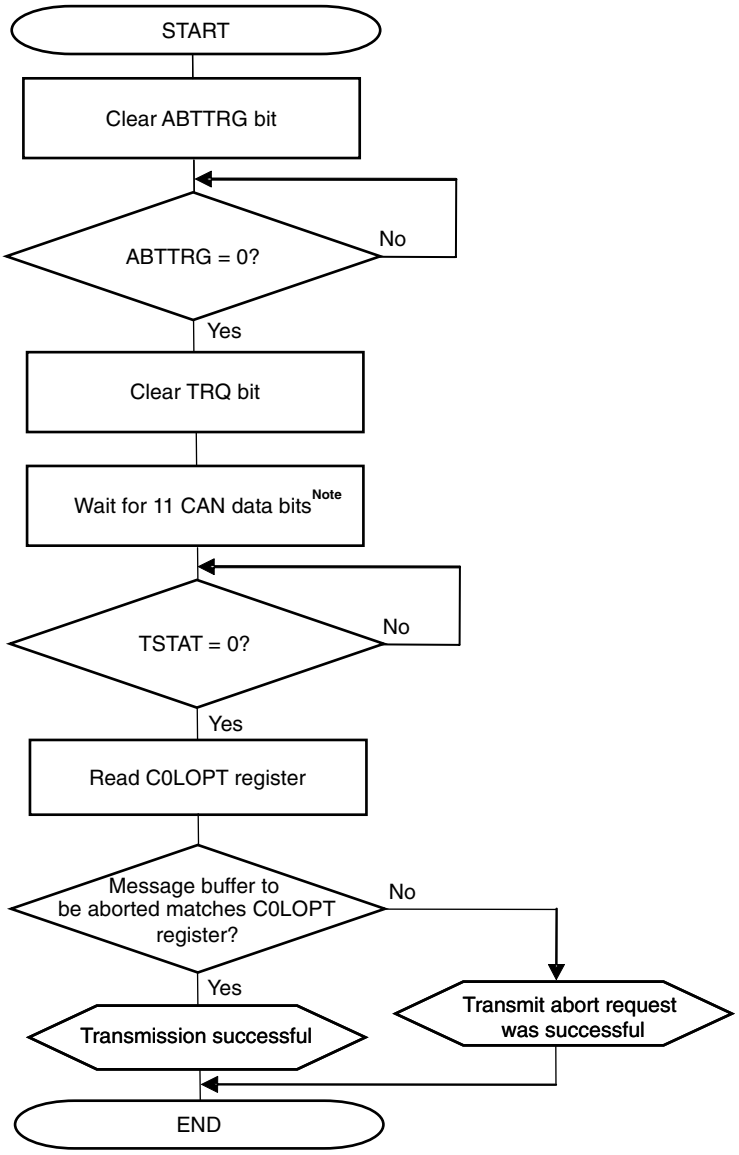
Cautions 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

Figure 16-48. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)



Note There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

Cautions 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

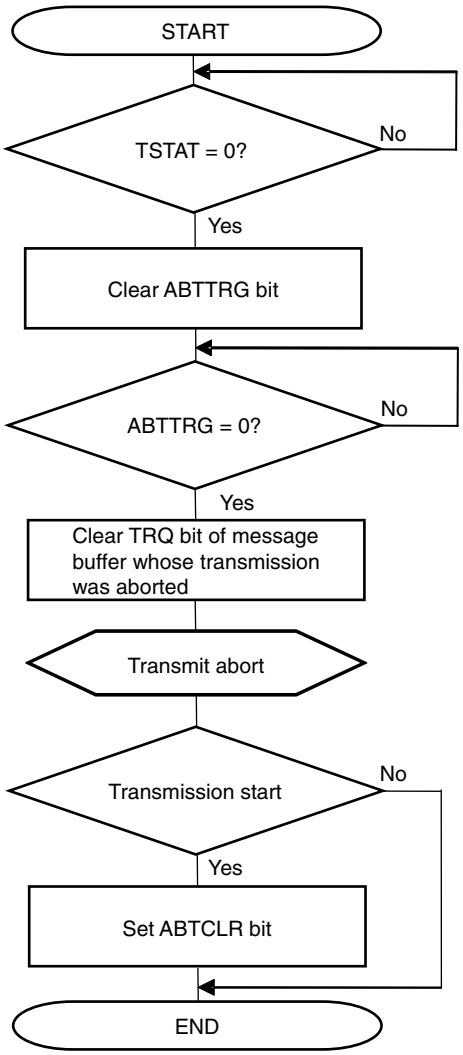
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

Figure 16-49 shows the processing not to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

Figure 16-49. ABT Transmission Abort Processing (Normal Operation Mode with ABT)

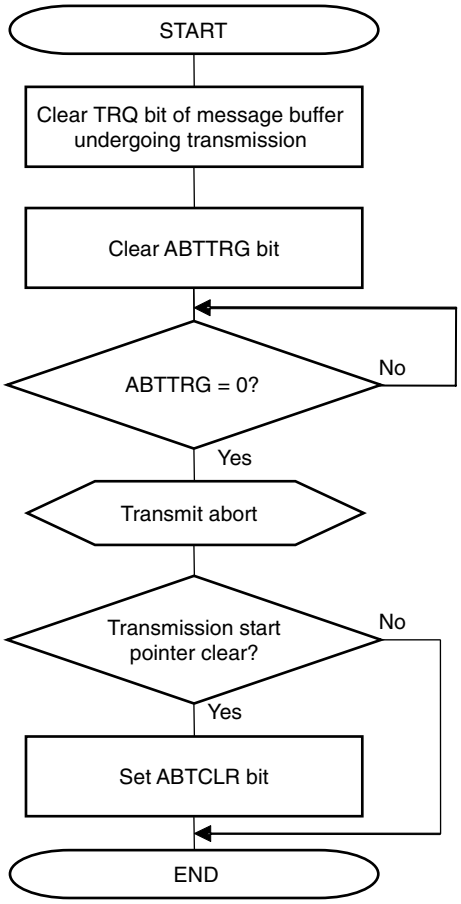


Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode transition request after ABTTRG bit is cleared (after ABT mode is aborted) following the procedure shown in Figure 16-49 or 16-50. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 16-47.

Figure 16-50 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

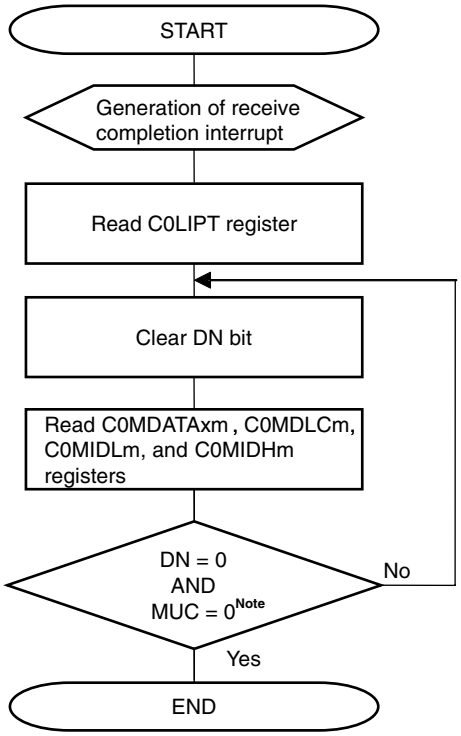
Figure 16-50. ABT Transmission Request Abort Processing (Normal Operation Mode with ABT)



Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode request after ABTTRG is cleared (after ABT mode is stopped) following the procedure shown in Figure 16-49 or 16-50. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 16-47.

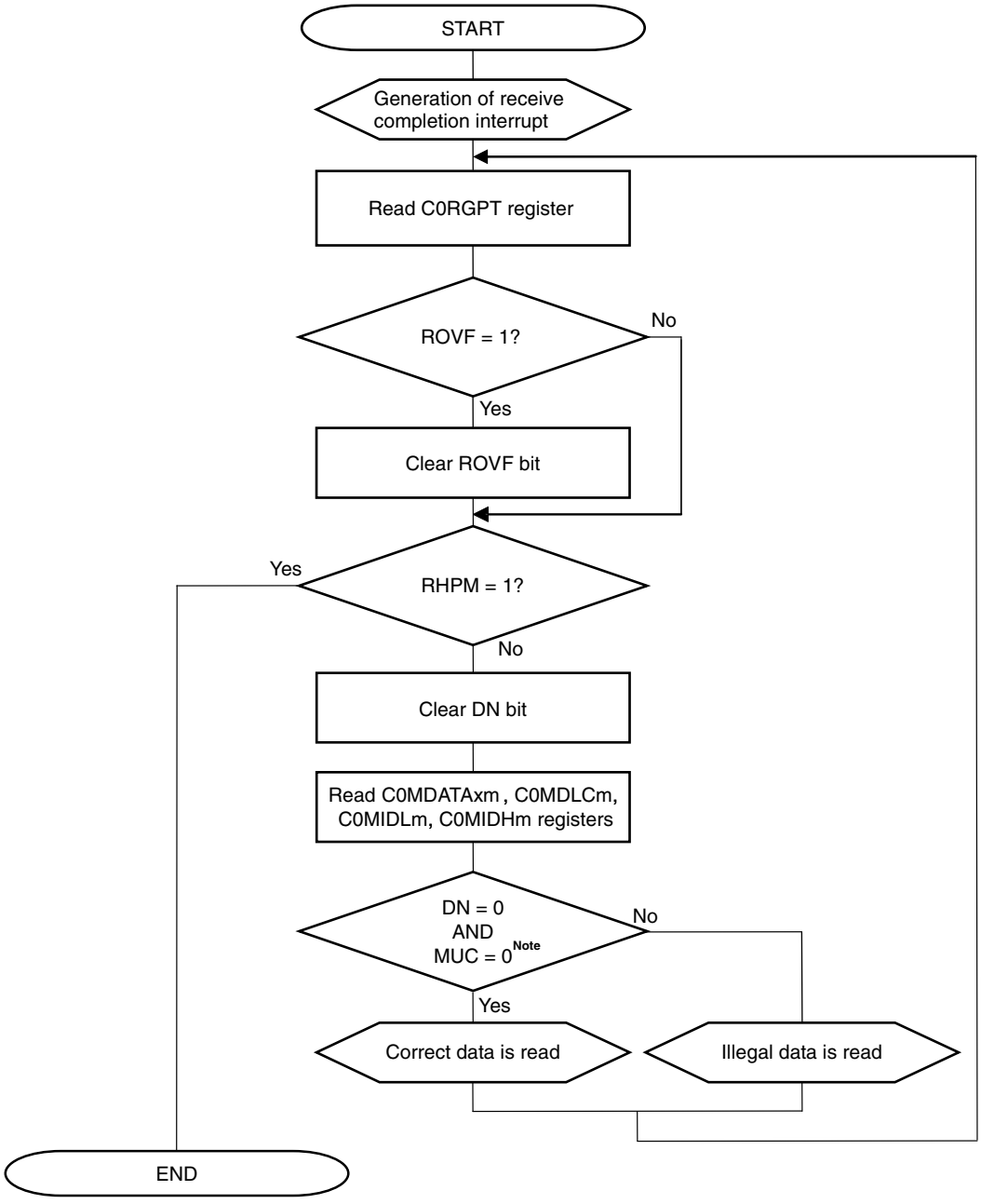
Figure 16-51. Reception via Interrupt (Using C0LIPT Register)



Note Check the MUC and DN bits using one read access.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing RX interrupts.

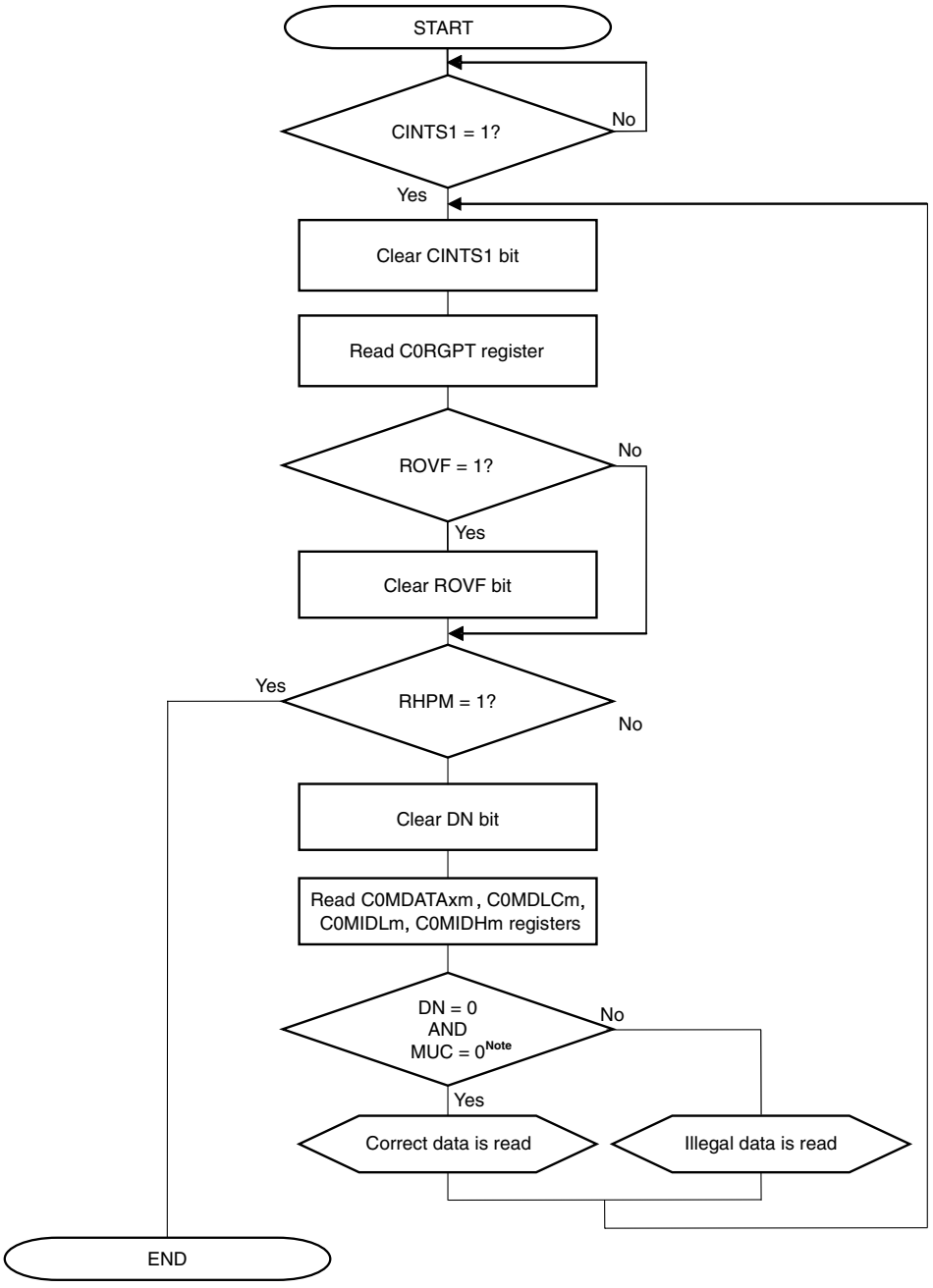
Figure 16-52. Reception via Interrupt (Using C0RGPT Register)



Note Check the MUC and DN bits using one read access.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing RX interrupts.

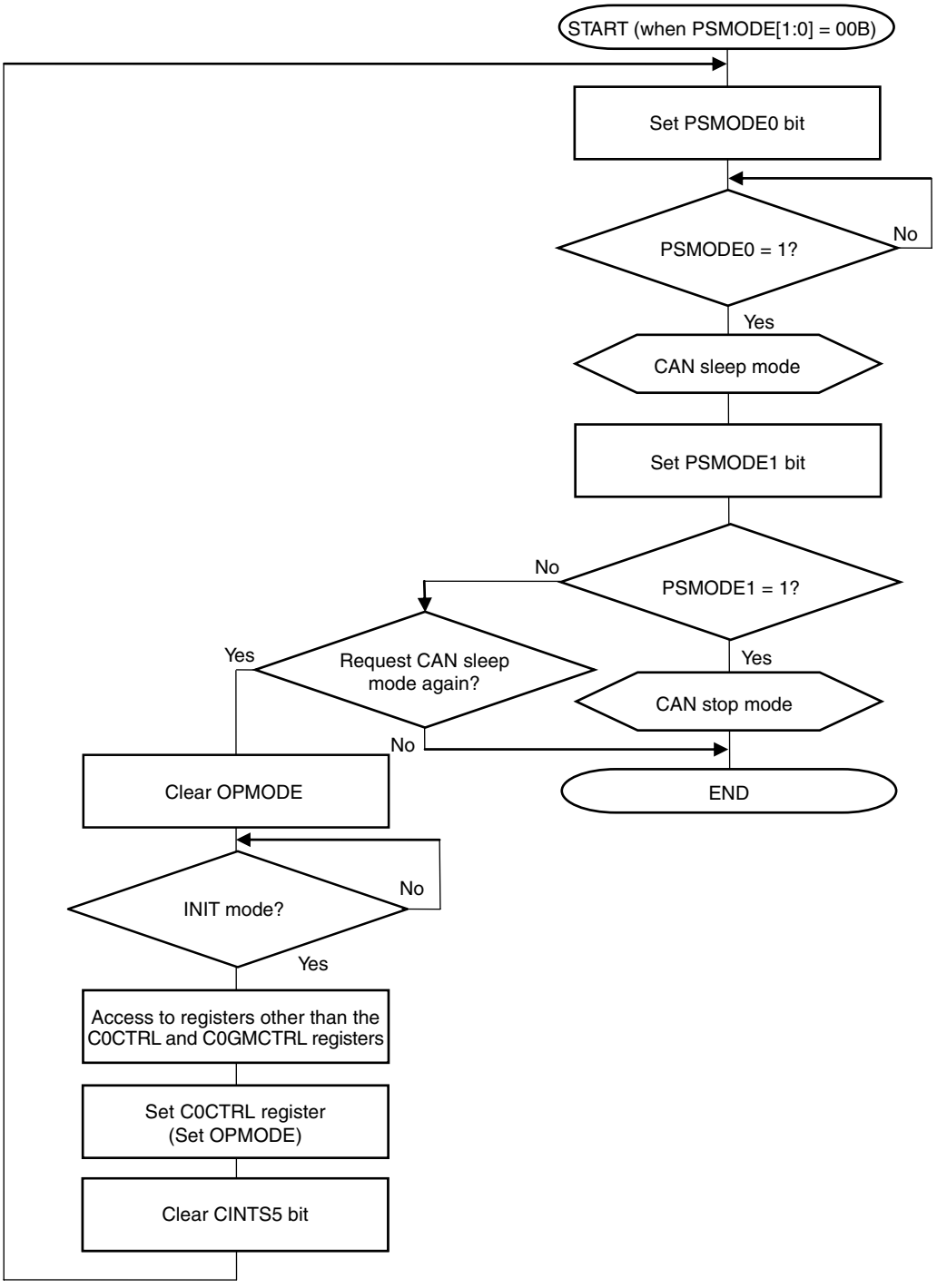
Figure 16-53. Reception via Software Polling



Note Check the MUC and DN bits using one read access.

Remark Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

Figure 16-54. Setting CAN Sleep Mode/Stop Mode



Caution To abort transmission before making a request for the CAN sleep mode, perform processing according to Figures 16-47 and 16-48.

<R> **Figure 16-55. Clear CAN Sleep/Stop Mode**

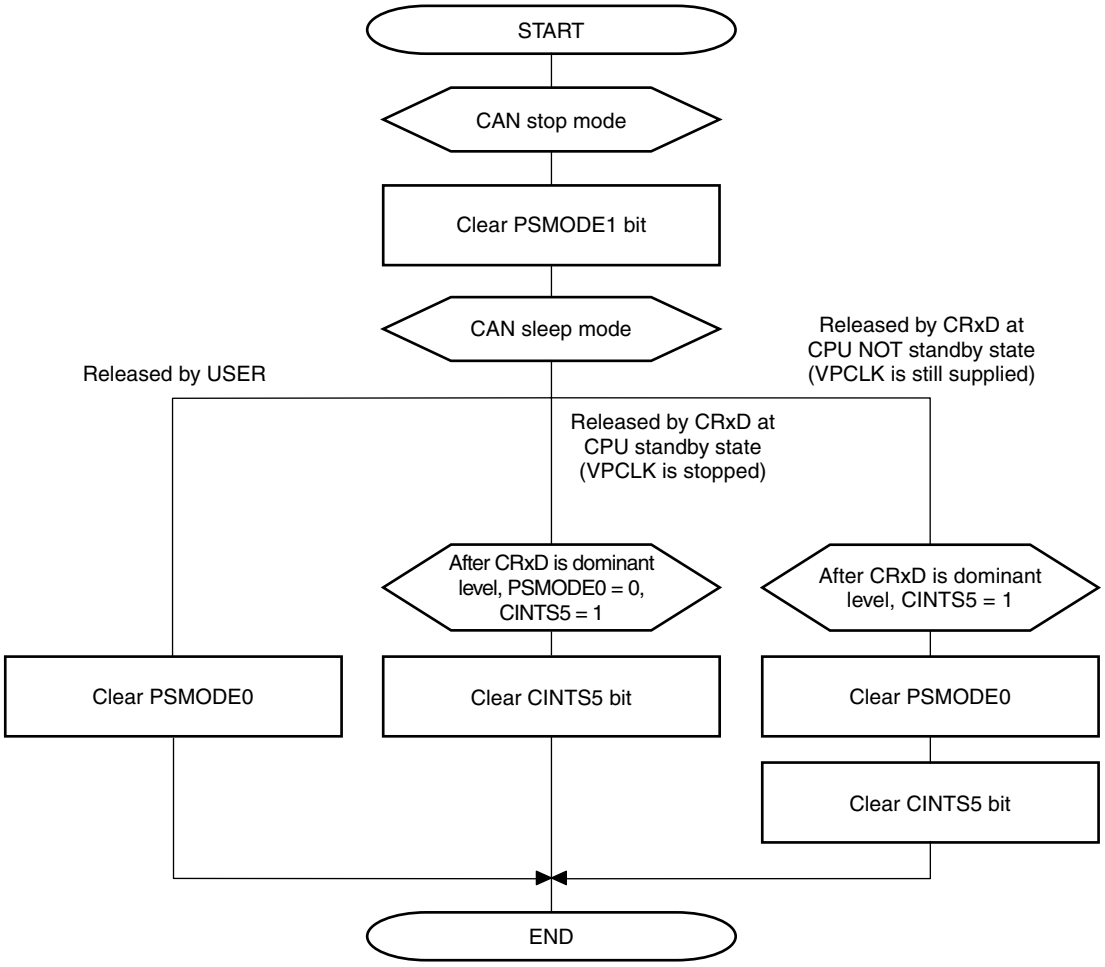
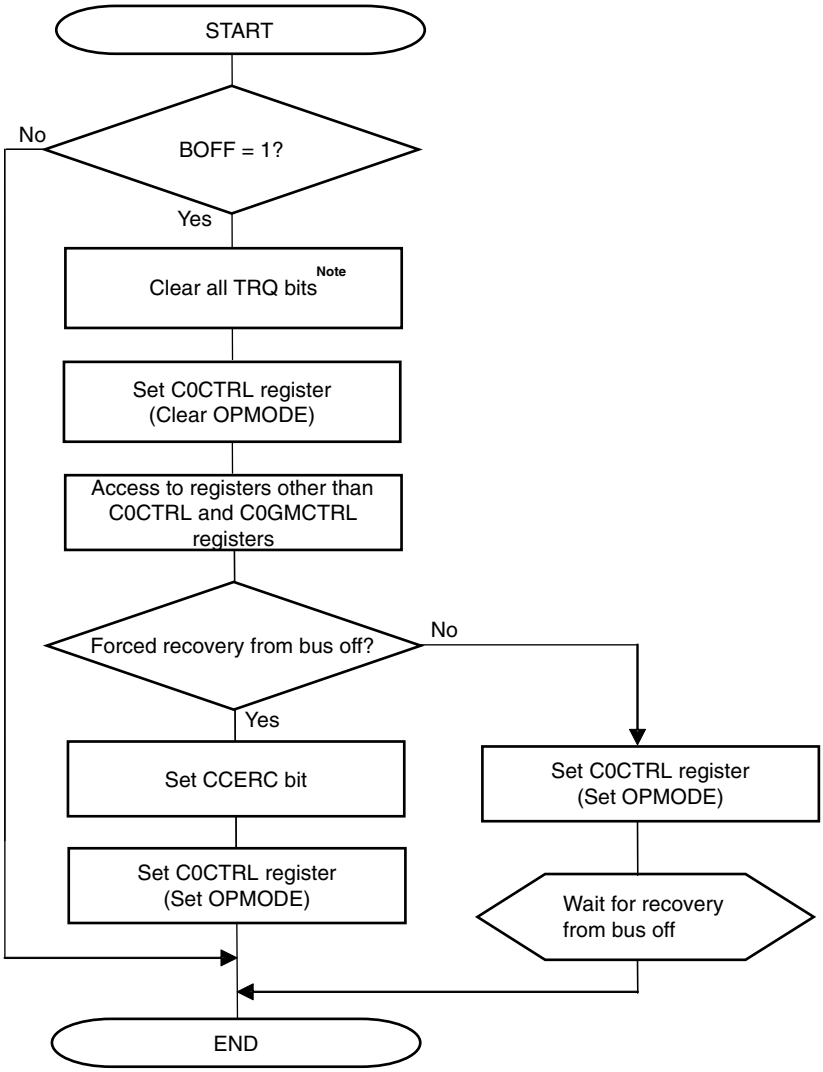


Figure 16-56. Bus-Off Recovery (Expect Normal Operation Mode with ABT)

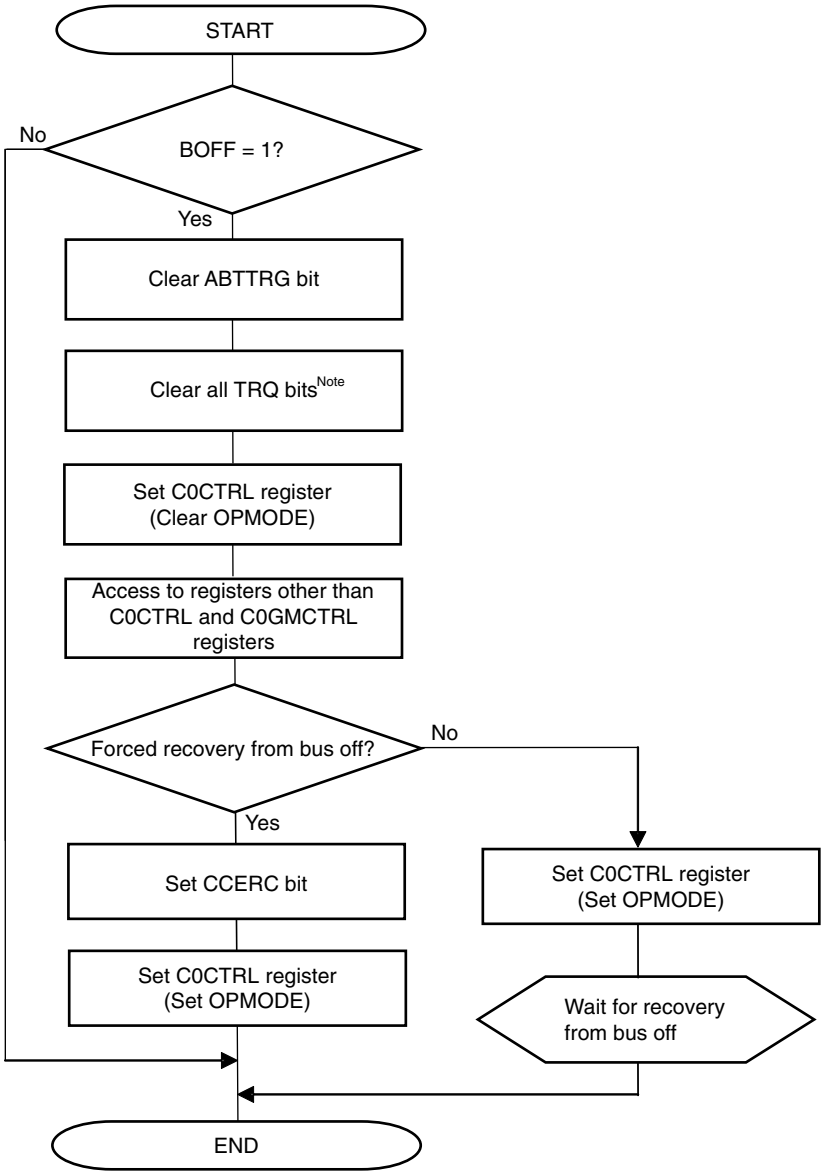


Note Clear all TRQ bits when re-initialization of message buffer is executed by clearing RDY bit before bus-off recovery sequence is started.

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.
Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 16-57. Bus-Off Recovery (Normal Operation Mode with ABT)



Note Clear all TRQ bits when re-initialization of message buffer is executed by clearing RDY bit before bus-off recovery sequence is started.

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.
Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 16-58. Normal Shutdown Process

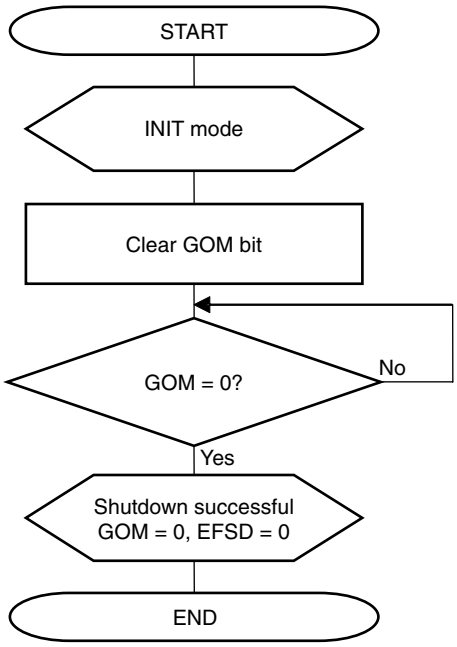
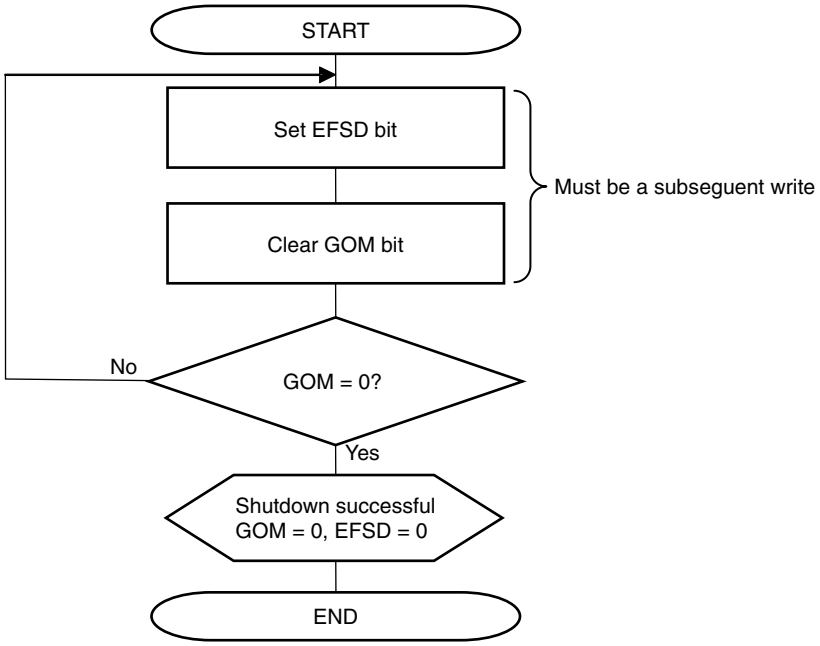


Figure 16-59. Forced Shutdown Process



Caution Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.

Figure 16-60. Error Handling

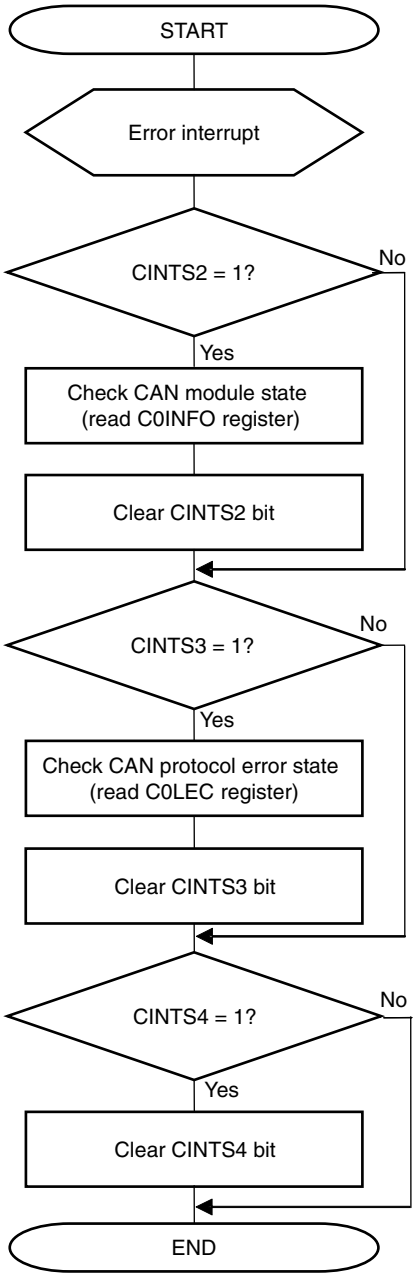
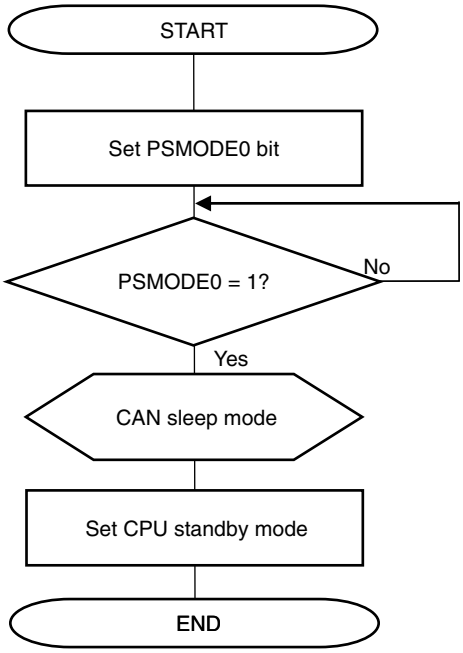
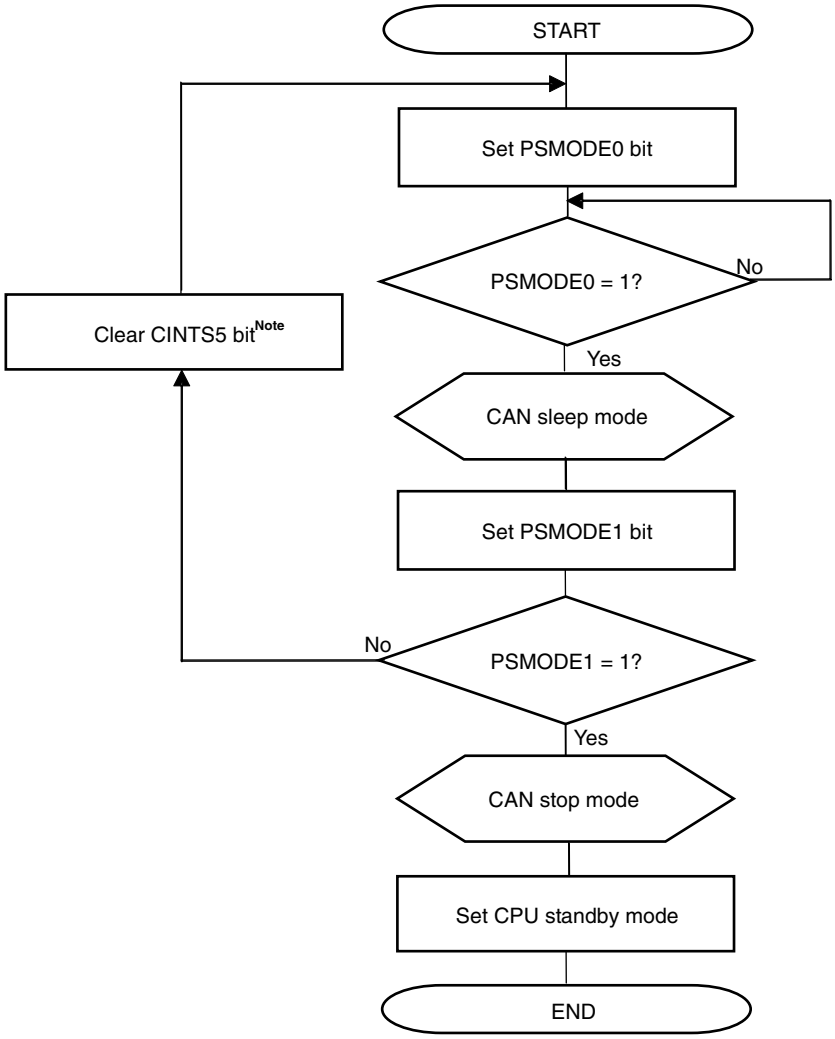


Figure 16-61. Setting CPU Standby (from CAN Sleep Mode)



Caution Before the CPU is set in the CPU standby mode, please check the CAN sleep mode or not. However, after check of the CAN sleep mode, until the CPU is set in the CPU standby mode, the CAN sleep mode may be cancelled by wakeup from CAN bus.

Figure 16-62. Setting CPU Standby (from CAN Stop Mode)



Note During wakeup interrupts

Caution The CAN stop mode can only be released by writing 01B to the PSMODE[1:0] bit of the C0CTRL register and not by a change in the CAN bus state.

CHAPTER 17 INTERRUPT FUNCTIONS

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, see **Table 17-1**.

A standby release signal is generated and STOP and HALT modes are released.

<R> 8 external interrupt requests and 29 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

<R> A total of 38 interrupt sources exist for maskable and software interrupts. In addition, they also have up to four reset sources (see **Table 17-1**).

Table 17-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTLVI	Low-voltage detection[Note 3]	Internal	0004H	(A)
	1	INTP0	Pin input edge detection	External	0006H	(B)
	2	INTP1			0008H	
	3	INTP2	Pin input edge detection		000AH	
		INTTM002	Match between TM02 and CR002 (when compare register is specified), TI012 pin valid edge detection (when capture register is specified)			
	4	INTP3	Pin input edge detection		000CH	
		INTTM012	Match between TM02 and CR012 (when compare register is specified), TI002 pin valid edge detection (when capture register is specified)			
	5	INTP4	Pin input edge detection		000EH	
		INTTM003	Match between TM03 and CR003 (when compare register is specified), TI013 pin valid edge detection (when capture register is specified)			
	6	INTP5	Pin input edge detection		0010H	
		INTTM013	Match between TM03 and CR013 (when compare register is specified), TI003 pin valid edge detection (when capture register is specified)			
	7	INTC0ERR	AFCAN0 error occurrence	Internal	0012H	(A)
	8	INTC0WUP	AFCAN0 wakeup		0014H	
	9	INTC0REC	AFCAN0 reception completion		0016H	
	10	INTC0TRX	AFCAN0 transmission completion		0018H	
	11	INTSRE60	UART60 reception error generation		001AH	
	12	INTSR60	End of UART60 reception		001CH	
	13	INTST60	End of UART60 transmission		001EH	
	14	INTCSI10	End of CSI10 transmission		0020H	
		INTSRE61	UART61 reception error generation			
	15	INTP6	Pin input edge detection	External	0022H	(B)
		INTSR61	End of UART61 reception	Internal		(A)
	16	INTP7	Pin input edge detection	External	0024H	(B)
		INTST61	End of UART61 transmission	Internal		(A)

Notes 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 17-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 0.

Table 17-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	17	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)	Internal	0026H	(A)
	18	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		0028H	
	19	INTTM50	Match between TM50 and CR50 (when compare register is specified)		002AH	
	20	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		002CH	
<R>	21	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		002EH	
	22	INTAD	End of A/D conversion		0030H	
	23	INTWTI	Watch timer reference time interval signal		0032H	
		INTDMU	DMU operation end			
	24	INTTM51[Note 3]	Match between TM51 and CR51 (when compare register is specified)		0034H	
	25	INTWT	Watch timer overflow		0036H	
	26	INTCSI11	End of CSI11 communication		0038H	
	27	INTTM001	Match between TM01 and CR001 (when compare register is specified), TI011 pin valid edge detection (when capture register is specified)		003AH	
	28	INTTM011	Match between TM01 and CR011 (when compare register is specified), TI001 pin valid edge detection (when capture register is specified)		003CH	
Software	–	BRK	BRK instruction execution	–	003EH	(C)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on clear			
		LVI	Low-voltage detection[Note 4]			
		WDT	WDT overflow			

Notes 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

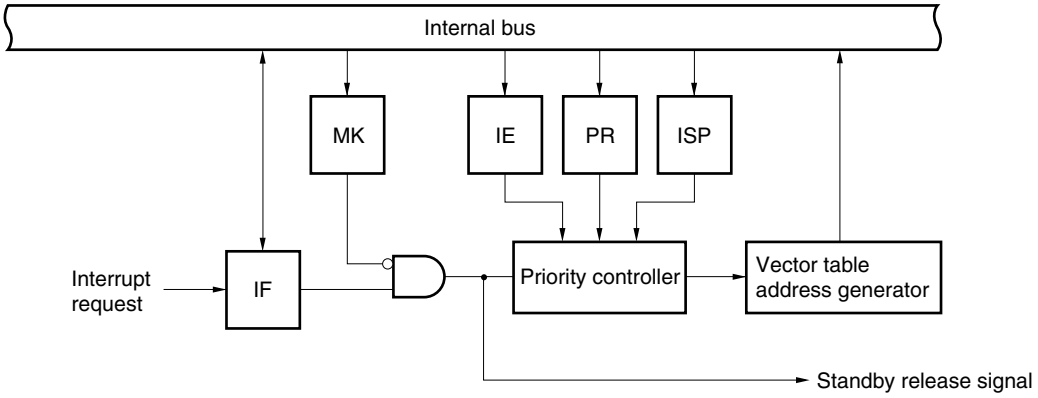
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 17-1.

3. When the 8-bit timer/event counter 51 is used in the carrier generator mode, the interrupt source is INTTM5H1 (see Figure 9-13 Transfer Timing).

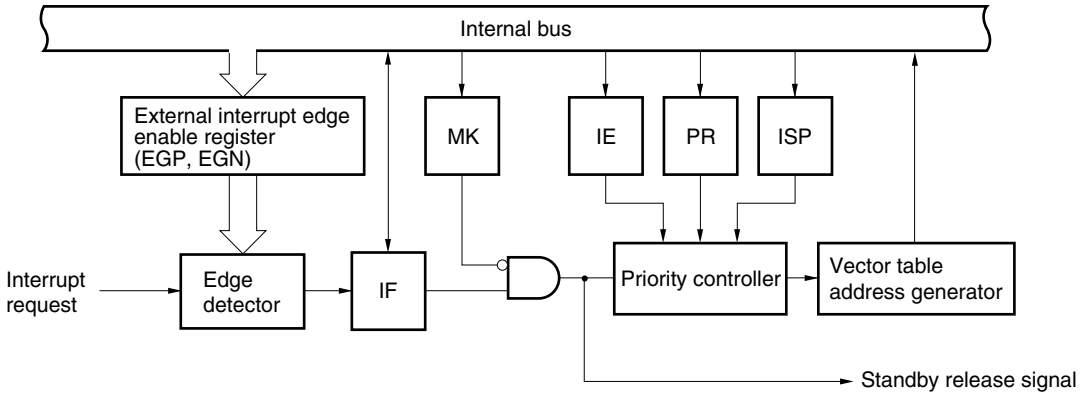
4. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

Figure 17-1. Basic Configuration of Interrupt Function

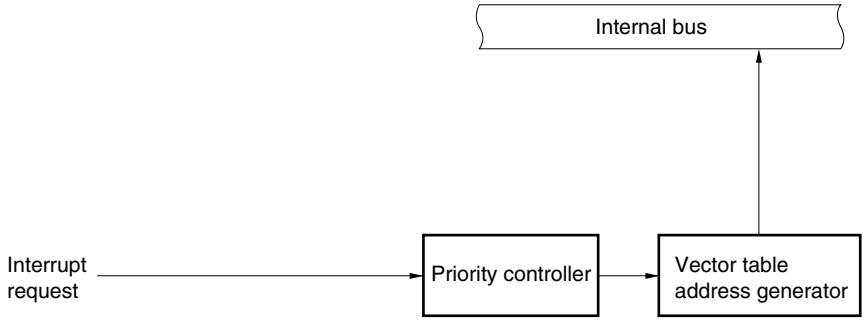
(A) Internal maskable interrupt



(B) External maskable interrupt (INTP0 to INTP7)



(C) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

17.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L, IF1H)
- Interrupt mask flag register (MK0L, MK0H, MK1L, MK1H)
- Priority specification flag register (PR0L, PR0H, PR1L, PR1H)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 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

Interrupt Request	Interrupt Request Flag		Register	Interrupt Mask Flag		Register	Priority Specification Flag		Register
INTLVI	LVIIF		IF0L	LVIMK		MK0L	LVIPR		PR0L
INTP0	PIF0			PMK0			PPR0		
INTP1	PIF1			PMK1			PPR1		
INTP2	PIF2	DUALIF3 Note 1		PMK2	DUALMK3 Note 2		PPR2	DUALPR3 Note 2	
INTTM002	TMIF002			TMMK002			TMPR002		
INTP3	PIF3	DUALIF4 Note 1		PMK3	DUALMK4 Note 2		PPR3	DUALPR4 Note 2	
INTTM012	TMIF012			TMMK012			TMPR012		
INTP4	PIF4	DUALIF5 Note 1		PMK4	DUALMK5 Note 2		PPR4	DUALPR5 Note 2	
INTTM003	TMIF003			TMMK003			TMPR003		
INTP5	PIF5	DUALIF6 Note 1		PMK5	DUALMK6 Note 2		PPR5	DUALPR6 Note 2	
INTTM013	TMIF013			TMMK013			TMPR013		
INTC0ERR	C0ERRIF			C0ERRMK			C0ERRPR		
INTC0WUP	C0WUPIF		IF0H	C0WUPMK		MK0H	C0WUPPR		PR0H
INTC0REC	C0RECIF			C0RECMK			C0RECPR		
INTC0TRX	C0TRXIF			C0TRXMK			C0TRXPR		
INTSRE60	SREIF60			SREMK60			SREPR60		
INTSR60	SRIF60			SRMK60			SRPR60		
INTST60	STIF60			STMK60			STPR60		
INTCSI10	CSIIF10	DUALIF0 Note 1		CSIMK10	DUALMK0 Note 2		CSIPR10	DUALPR0 Note 2	
INTSRE61	SREIF61			SREMK61			SREPR61		
INTP6	PIF6	DUALIF1 Note 1		PMK6	DUALMK1 Note 2		PPR6	DUALPR1 Note 2	
INTSR61	SRIF61			SRMK61			SRPR61		
INTP7	PIF7	DUALIF2 Note 1	IF1L	PMK7	DUALMK2 Note 2	MK1L	PPR7	DUALPR2 Note 2	PR1L
INTST61	STIF61			STMK61			STPR61		
INTTMH1	TMIFH1			TMMKH1			TMPRH1		
INTTMH0	TMIFH0			TMMKH0			TMPRH0		
INTTM50	TMIF50			TMMK50			TMPR50		
INTTM000	TMIF000			TMMK000			TMPR000		
INTTM010	TMIF010			TMMK010			TMPR010		
INTAD	ADIF			ADMK			ADPR		
INTWTI	WTIIF	DUALIF7 Note 1		WTIMK	DUALMK7 Note 2		WTIPR	DUALPR7 Note 2	
INTDMU	DMUIF			DMUMK			DMUPR		
INTTM51 Note 3	TMIF51		IF1H	TMMK51		MK1H	TMPR51		PR1H
INTWT	WTIF			WTMK			WTPR		
INTCSI11	CSIIF11			CSIMK11			CSIPR11		
INTTM001	TMIF001			TMMK001			TMPR001		
INTTM011	TMIF011			TMMK011			TMPR011		

Notes 1. If either of the two types of interrupt sources is generated, these flags are set (1).

2. Both types of interrupt sources are supported.

3. When the 8-bit timer/event counter 51 is used in the carrier generator mode, the interrupt source is INTTM5H1 (see Figure 9-13 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 read with a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	C0ERRIF	DUALIF6 PIF5 TMIF013	DUALIF5 PIF4 TMIF003	DUALIF4 PIF3 TMIF012	DUALIF3 PIF2 TMIF002	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	DUALIF1 PIF6 SRIF61	DURLIF0 CSIIF10 SREIF61	STIF60	SRIF60	SREIF60	C0TRXIF	C0RECIF	C0WUPIF

<R>

Address: FFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	DUALIF7 WTIIF DMUIF	ADIF	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	DUALIF2 PIF7 STIF61

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF1H	0	0	0	TMIF011	TMIF001	CSIIF11	WTIF	TMIF51

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Be sure to set bits 5 to 7 of IF1H to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

Cautions 3. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).
If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, and MK1H are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, and MK1L and MK1H are combined to form 16-bit registers MK0 and MK1, they are set with a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 17-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	C0ERRMK	DUALMK6 PMK5 TMMK013	DUALMK5 PMK4 TMMK003	DUALMK4 PMK3 TMMK012	DUALMK3 PMK2 TMMK002	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	DUALMK1 PMK6 SRMK61	DURLMK0 CSIMK10 SREMK61	STMK60	SRMK60	SREMK60	C0TRXMK	C0RECMK	C0WUPMK

<R>

Address: FFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	DUALMK7 WTIMK DMUMK	ADMK	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	DUALMK2 PMK7 STMK61

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK1H	1	1	1	TMMK011	TMMK001	CSIMK11	WTMK	TMMK51

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 5 to 7 of MK1H to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, PR1L, and PR1H are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H, and PR1L and PR1H are combined to form 16-bit registers PR0 and PR1, they are set with a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 17-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	C0ERRPR	DUALPR6 PPR5 TMPR073	DUALPR5 PPR4 TMPR003	DUALPR4 PPR3 TMPR012	DUALPR3 PPR2 TMPR002	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	DUALPR1 PPR6 SRPR61	DURLPR0 CSIPR10 SREPR61	STPR60	SRPR60	SREPR60	C0TRXPR	C0RECPR	C0WUPPR

<R>

Address: FFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR1L	DUALPR7 WTIPR DMUPR	ADPR	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	DUALPR2 PPR7 STPR61

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR1H	1	1	1	TMPR011	TMPR001	CSIPR11	WTPR	TMPR51

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bit 5 to 7 of PR1H to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP7.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 17-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	EGP7	EPG6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 17-3 shows the ports corresponding to EGPn and EGNn.

Table 17-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	External Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3
EGP4	EGN4	P33	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P72	INTP6
EGP7	EGN7	P73	INTP7

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 7

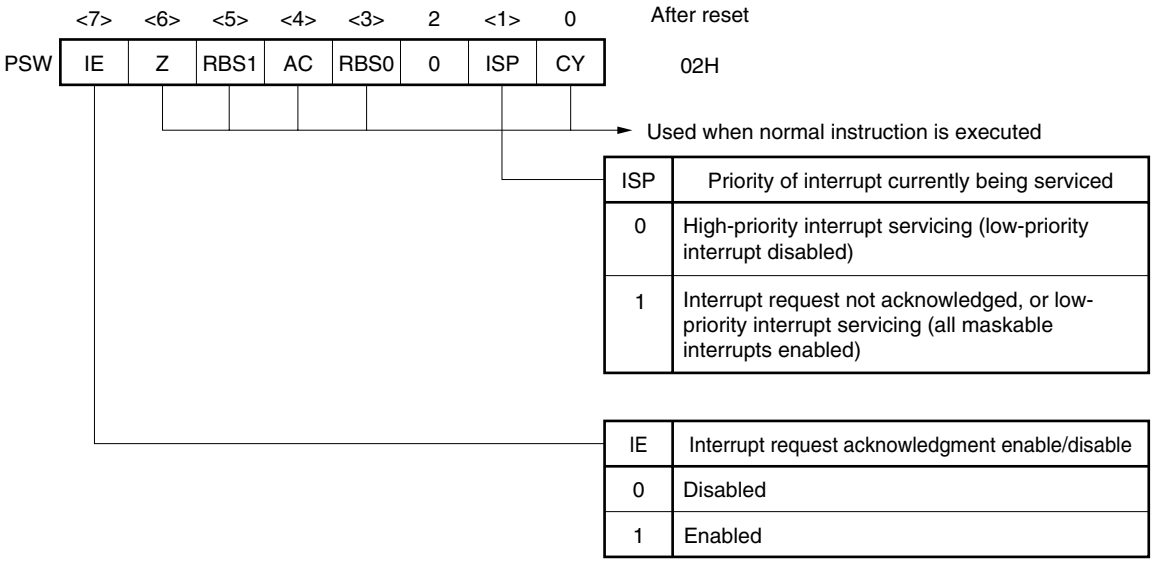
(5) Program status word (PSW)

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 17-6. Format of Program Status Word



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

17.4 Interrupt Servicing Operations

17.4.1 Maskable interrupt acknowledgement

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 interrupt servicing is performed are listed in Table 17-4 below.

For the interrupt request acknowledgement timing, see **Figures 17-8** and **17-9**.

Table 17-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

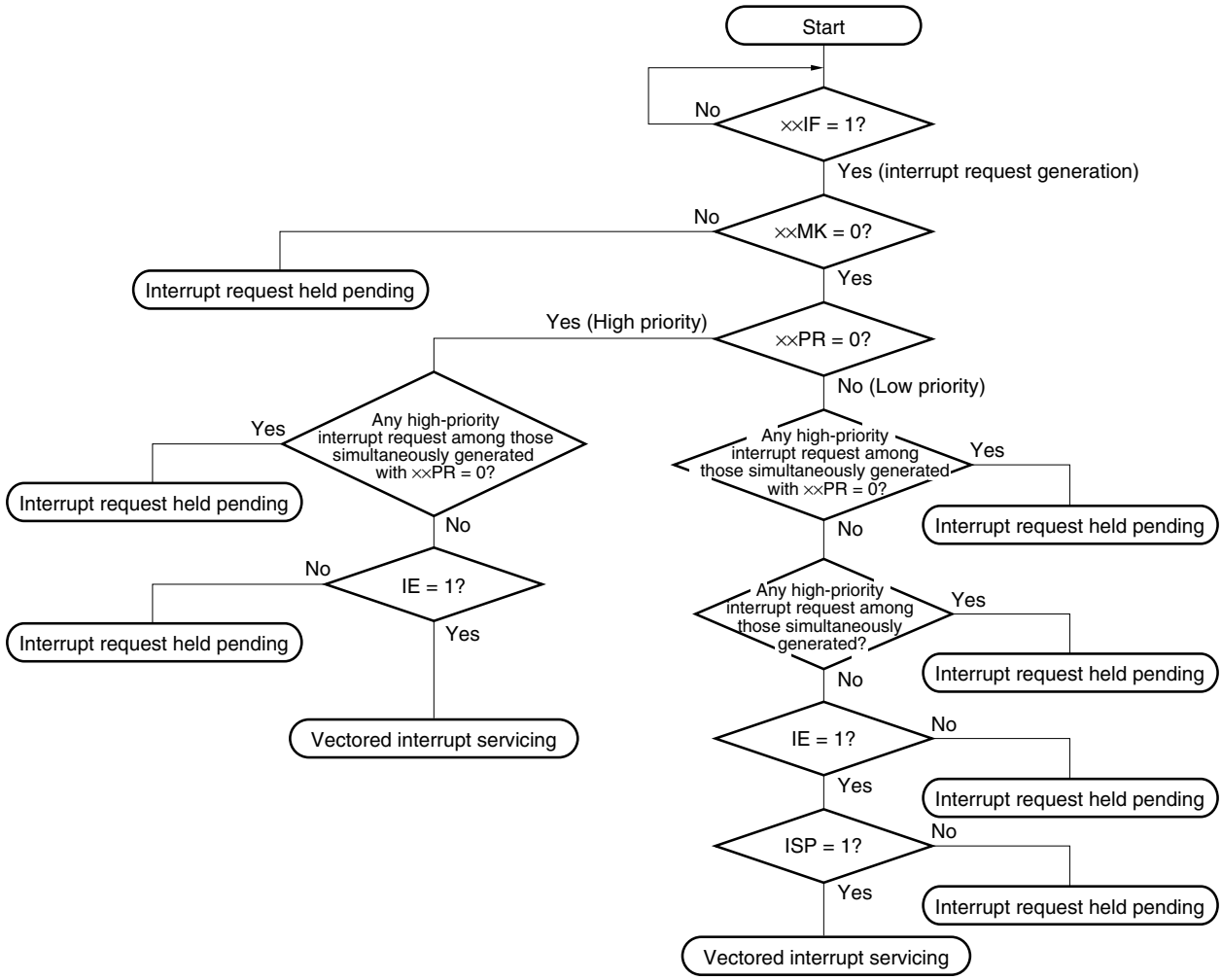
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 17-7 shows the interrupt request acknowledgement 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 17-7. Interrupt Request Acknowledgement Processing Algorithm



××IF: Interrupt request flag

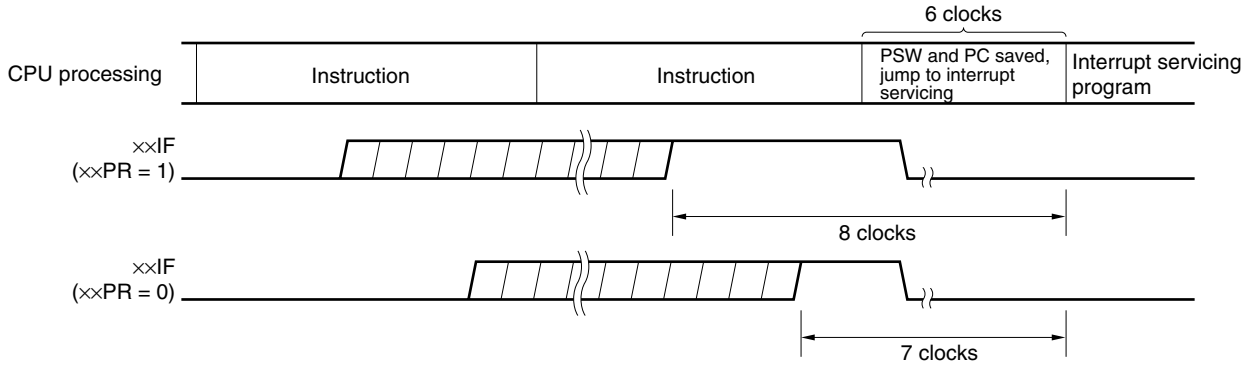
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgement of maskable interrupt request (1 = Enable, 0 = Disable)

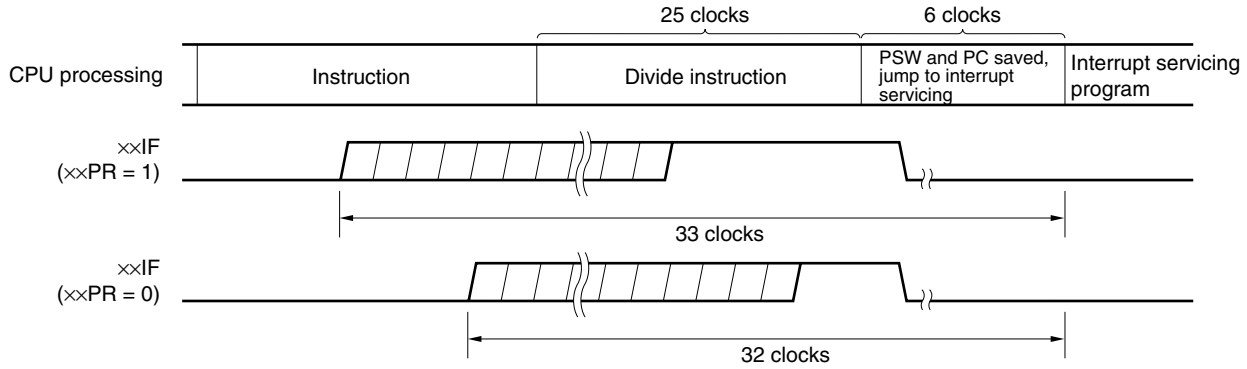
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 17-8. Interrupt Request Acknowledgement Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 17-9. Interrupt Request Acknowledgement Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

17.4.2 Software interrupt request acknowledgement

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 acknowledgement enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgement 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 acknowledgement.

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 17-10 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 acknowledgement is disabled.

IE = 1: Interrupt request acknowledgement is enabled.

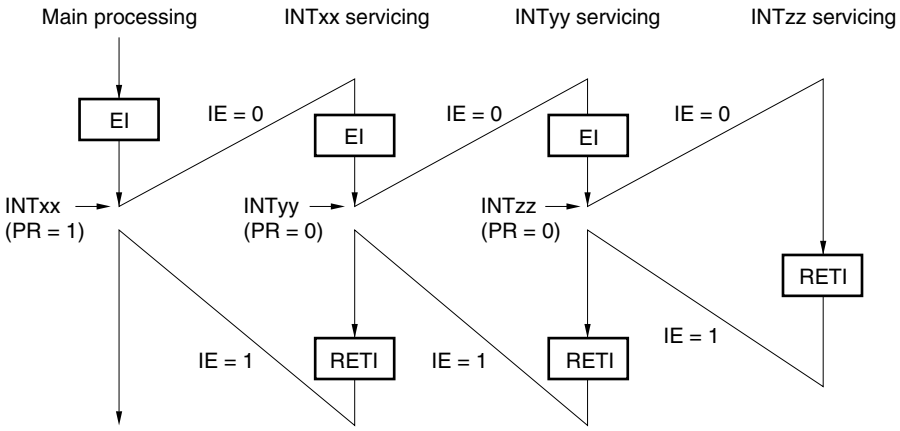
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

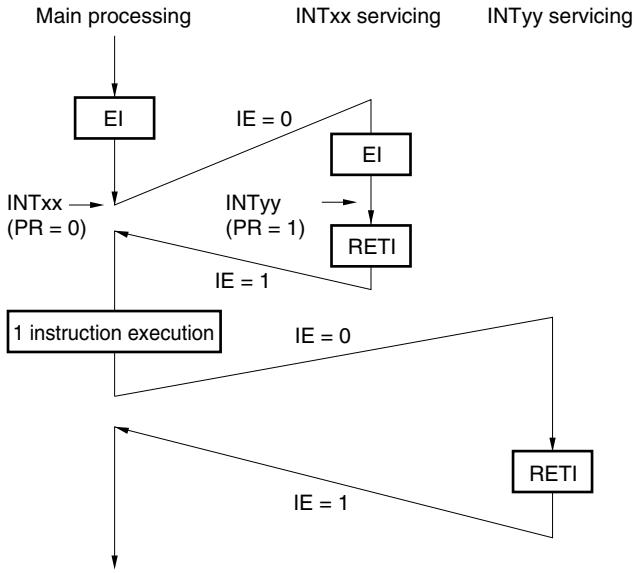
Figure 17-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

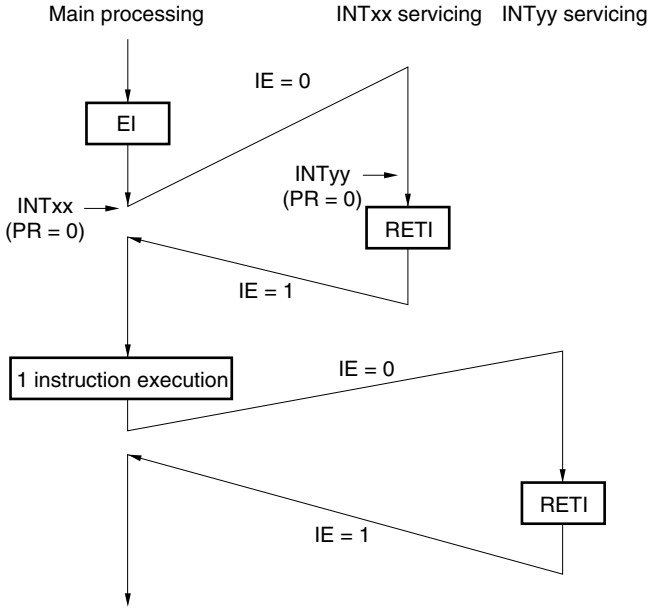


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 17-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledgement disabled

17.4.4 Interrupt request hold

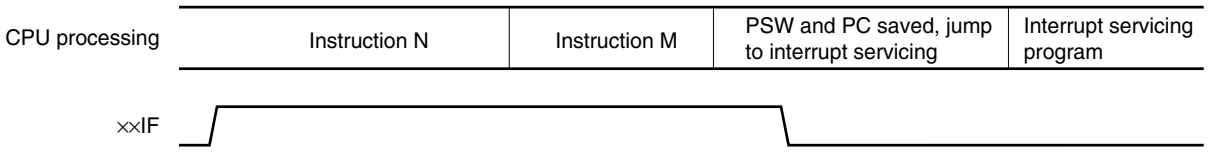
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgement is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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 17-11 shows the timing at which interrupt requests are held pending.

Figure 17-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (instruction request).

CHAPTER 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 Standby function

The standby function is designed to reduce the operating current of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, internal high-speed oscillator, internal low-speed oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations.

(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, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The subsystem clock oscillation cannot be stopped. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

18.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 6 CLOCK GENERATOR**.

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

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

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 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

MOST 11	MOST 13	MOST 14	MOST 15	MOST 16	Oscillation stabilization time status				
						fx = 4 MHz	fx = 5 MHz	fx = 10 MHz	fx = 20 MHz
1	0	0	0	0	2^{11}/fx min.	512 μs min.	409.6 μs min.	204.8 μs min.	102.4 μs min.
1	1	0	0	0	2^{13}/fx min.	2.05 ms min.	1.64 ms min.	819.2 μs min.	409.6 μs min.
1	1	1	0	0	2^{14}/fx min.	4.10 ms min.	3.27 ms min.	1.64 ms min.	819.2 μs min.
1	1	1	1	0	2^{15}/fx min.	8.19 ms min.	6.55 ms min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	2^{16}/fx min.	16.38 ms min.	13.11 ms min.	6.55 ms min.	3.27 ms min.

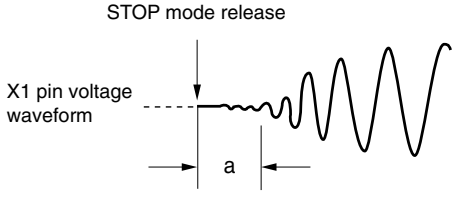
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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 time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization 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 = 4 MHz	f_X = 5 MHz	f_X = 10 MHz	f_X = 20 MHz
0	0	1	$2^{11}/f_X$	512 μs min.	409.6 μs min.	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_X$	2.05 ms min.	1.64 ms min.	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_X$	4.10 ms min.	3.27 ms min.	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_X$	8.19 ms min.	6.55 ms min.	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_X$	16.38 ms min.	13.11 ms min.	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.

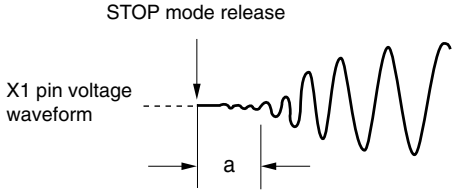
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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 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, internal high-speed oscillation clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 18-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained	
		f_X	Status before HALT mode was set is retained	Operation continues (cannot be stopped)	Status before HALT mode was set is retained
		f_{EXCLK}	Operates or stops by external clock input		Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
		f_{EXCLKS}	Operates or stops by external clock input		
	f_{RL}		Status before HALT mode was set is retained		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before HALT mode was set is retained		
Port (latch)			Status before HALT mode was set is retained		
16-bit timer/event counter		00	Operable		
		01			
		02			
		03			
8-bit timer/event counter		50			
		51			
8-bit timer		H0			
		H1			
Watch timer					
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
Clock output			Operable		
Buzzer output					
A/D converter					
Serial interface		UART60			
		UART61			
		CSI10			
		CSI11			
CAN controller					
Multiplier/divider					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{EXCLKS}: External subsystem clock
f_{RL}: Internal low-speed oscillation clock

Table 18-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock	
			When CPU Is Operating on XT1 Clock (f_{XT})	When CPU Is Operating on External Subsystem Clock (f_{EXCLKS})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{RH}	Status before HALT mode was set is retained	
		f_X		
		f_{EXCLK}	Operates or stops by external clock input	
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained
		f_{EXCLKS}	Operates or stops by external clock input	Operation continues (cannot be stopped)
	f_{RL}		Status before HALT mode was set is retained	
CPU			Operation stopped	
Flash memory			Operation stopped	
RAM			Status before HALT mode was set is retained	
Port (latch)			Status before HALT mode was set is retained	
16-bit timer/event counter		00 Note	Operable	
		01 Note		
		02 Note		
		03 Note		
8-bit timer/event counter		50 Note		
		51 Note		
8-bit timer		H0		
		H1		
Watch timer				
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.	
Clock output			Operable	
Buzzer output			Operable. However, operation disabled when peripheral hardware clock (f_{PRS}) is stopped.	
A/D converter				
Serial interface	UART60		Operable	
	UART61			
	CSI10 Note			
	CSI11 Note			
CAN controller				
Multiplier/divider				
Power-on-clear function				
Low-voltage detection function				
External interrupt				

Note When the CPU is operating on the subsystem clock and the internal high-speed oscillation clock has been stopped, do not start operation of these functions on the external clock input from peripheral hardware pins.

(Remark is listed on the next page.)

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{EXCLKS}: External subsystem clock
f_{RL}: Internal low-speed oscillation clock

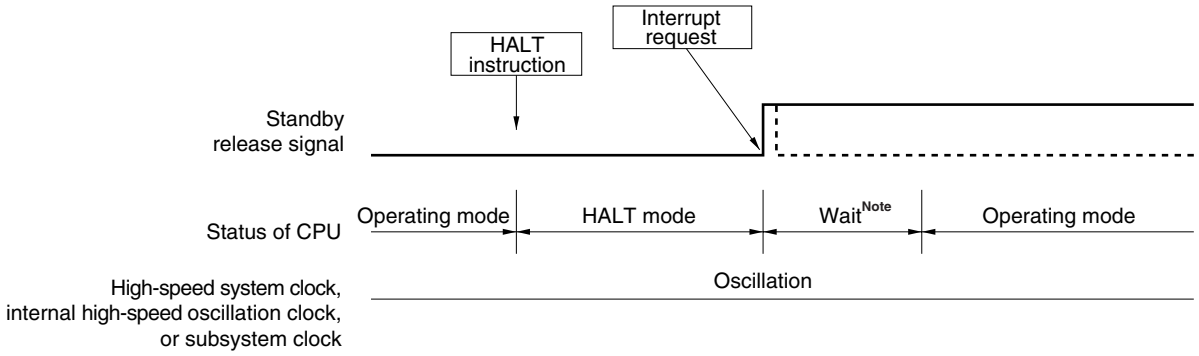
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement 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: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

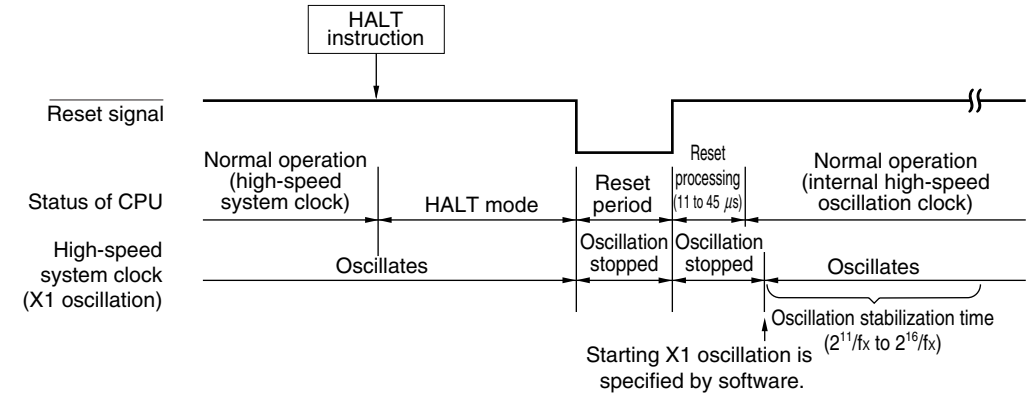
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

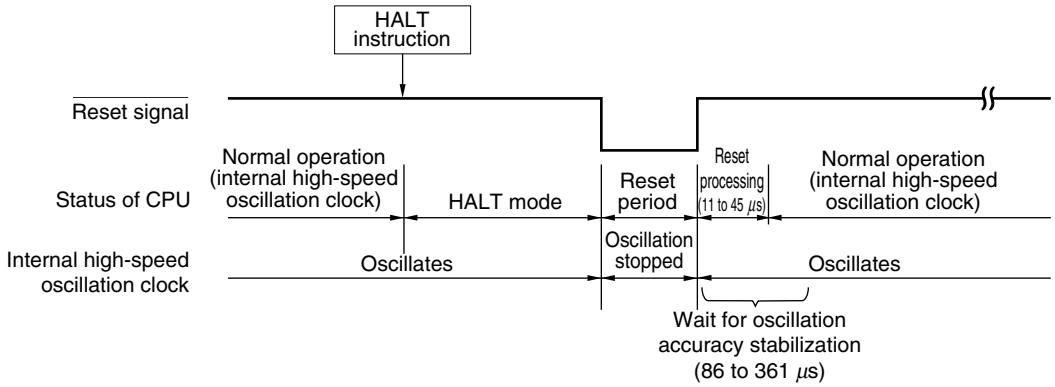
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 18-4. HALT Mode Release by Reset

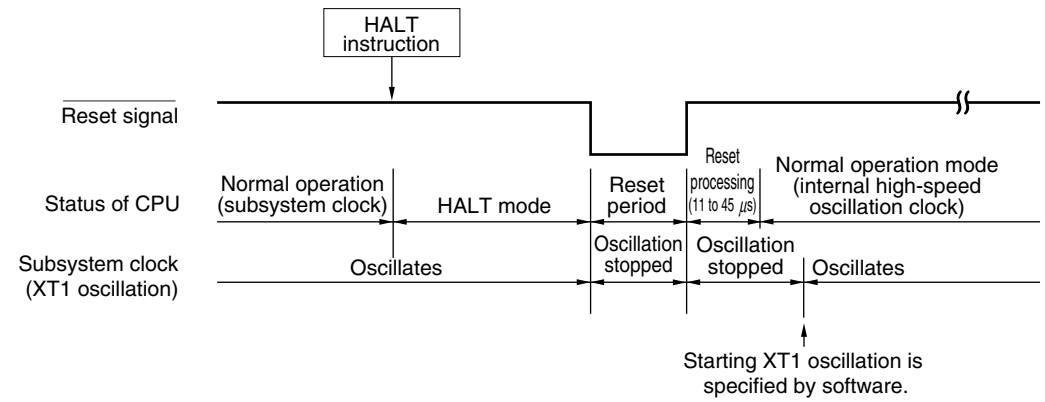
(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Table 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 signal input	–	–	×	×	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 main system clock.

Caution Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 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_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Stopped		
		f_X			
		f_{EXCLK}	Input invalid		
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
		f_{EXCLKS}	Operates or stops by external clock input		
	f_{RL}		Status before STOP mode was set is retained		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before STOP mode was set is retained		
Port (latch)			Status before STOP mode was set is retained		
16-bit timer/event counter		00 Note	Operation stopped		
		01 Note			
		02 Note			
		03 Note			
8-bit timer/event counter		50 Note	Operable only when TI50 is selected as the count clock		
		51 Note	Operable only when TI51 is selected as the count clock		
8-bit timer		H0	Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation		
		H1	Operable only when f_{RL}, $f_{RL}/2^7$, $f_{RL}/2^9$ is selected as the count clock		
Watch timer			Operable only when subsystem clock 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.		
Clock output			Operable only when subsystem clock is selected as the count clock		
Buzzer output			Operation stopped		
A/D converter					
Serial interface	UART60		Operable only when TM50 output is selected as the serial clock during 8-bit timer/event counter 50 operation		
	UART61				
	CSI10 Note		Operable only when external clock is selected as the serial clock		
	CSI11 Note				
CAN controller			Operable. Can be woken up from sleep mode.		
Multiplier/divider			Operation stopped		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Note Do not start operation of these functions on the external clock input from peripheral hardware pins in the stop mode.

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

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{EXCLKS}: External subsystem clock
f_{RL}: Internal low-speed oscillation clock

Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

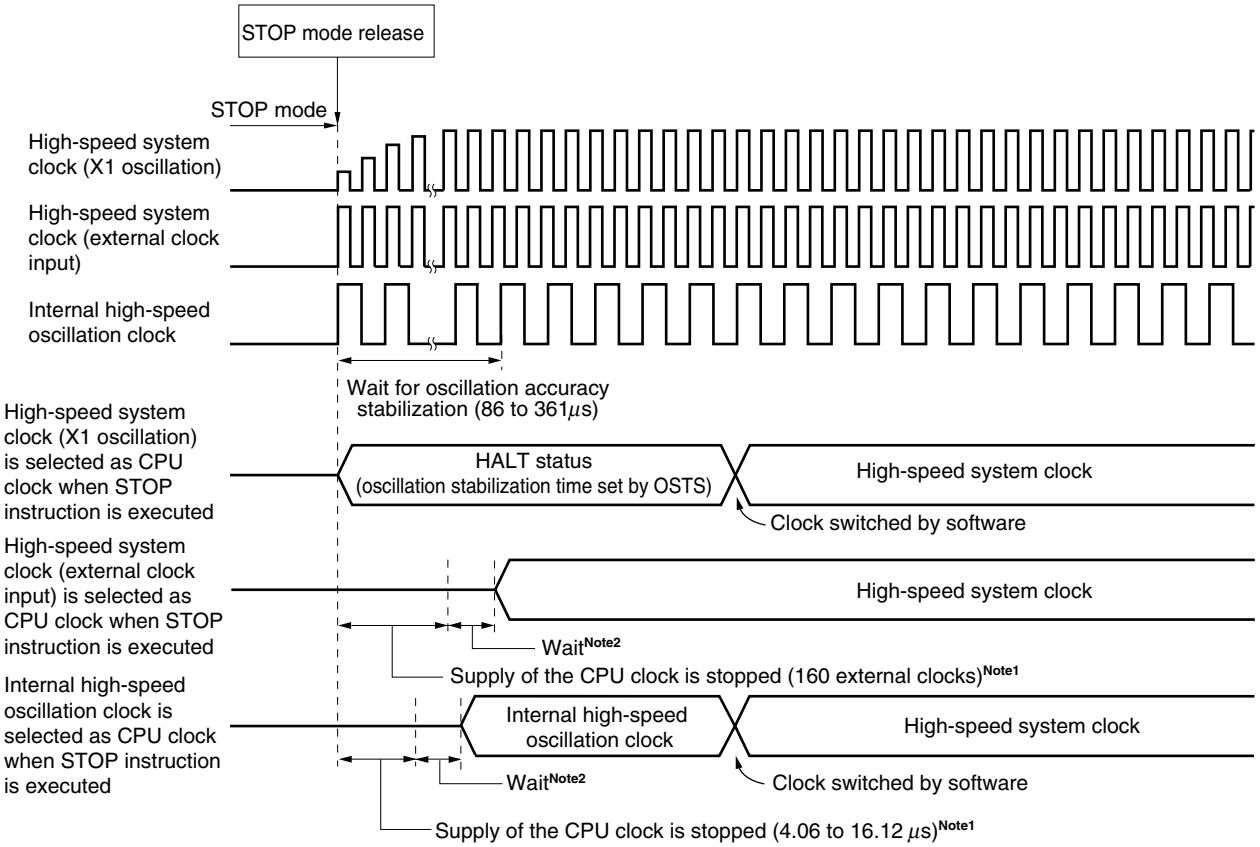
2. Even if "internal low-speed oscillator can be stopped by software" is selected by the option byte, the internal low-speed oscillator continues in the STOP mode in the status before the STOP mode is set. To stop the internal low-speed oscillator in the STOP mode, stop it by software and then execute the STOP instruction.

3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the internal high-speed oscillation clock before the next execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillator to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

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

(2) STOP mode release

Figure 18-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Notes 1. When AMPH = 1

2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

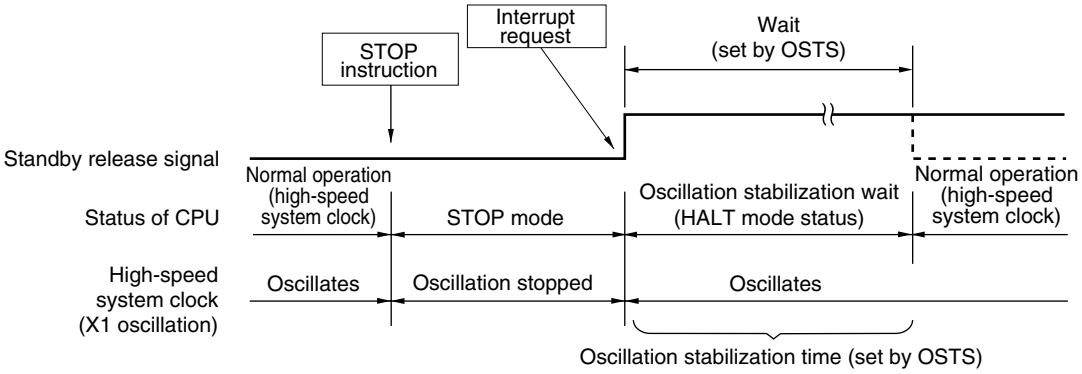
The STOP mode can be released by the following two sources.

(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.

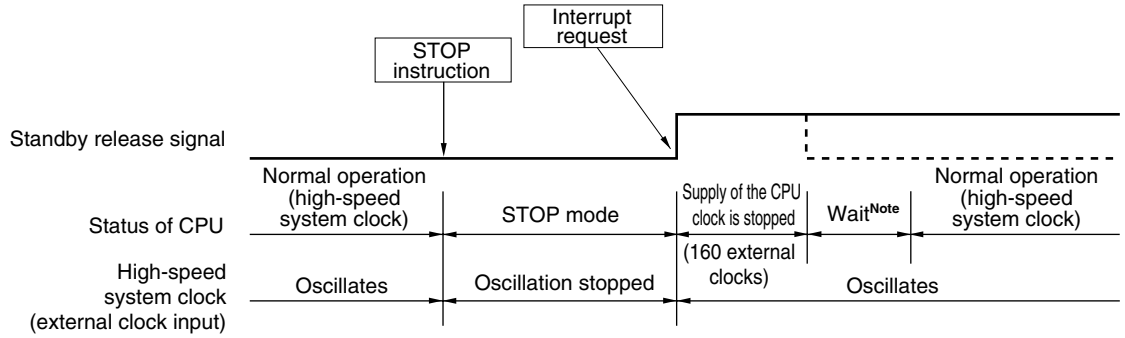
Figure 18-6. STOP Mode Release by Interrupt Request Generation (1/2)

(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 (1/2)

- When AMPH = 1



Note The wait time is as follows:

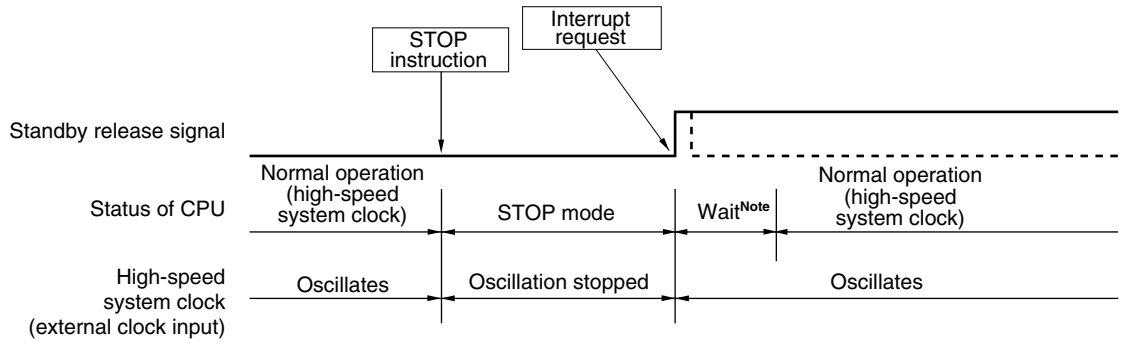
- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

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)

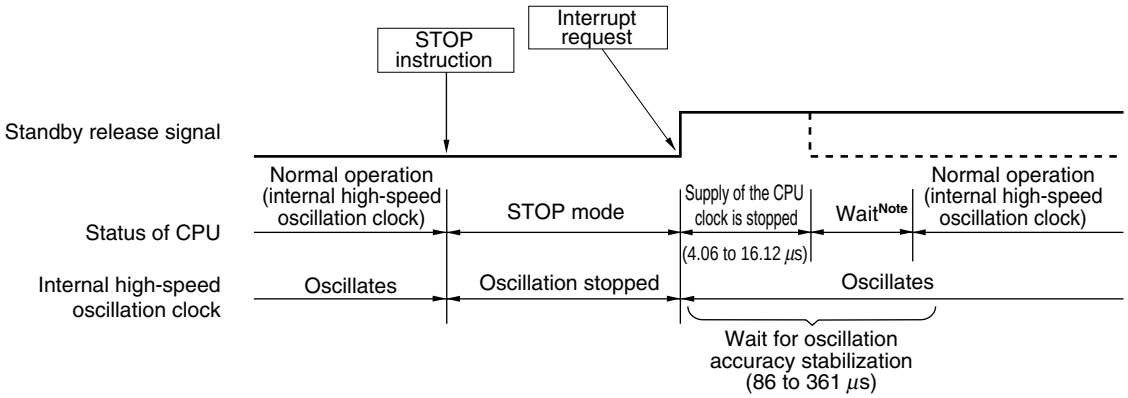
(2) When high-speed system clock (external clock input) is used as CPU clock (2/2)

- When AMPH = 0

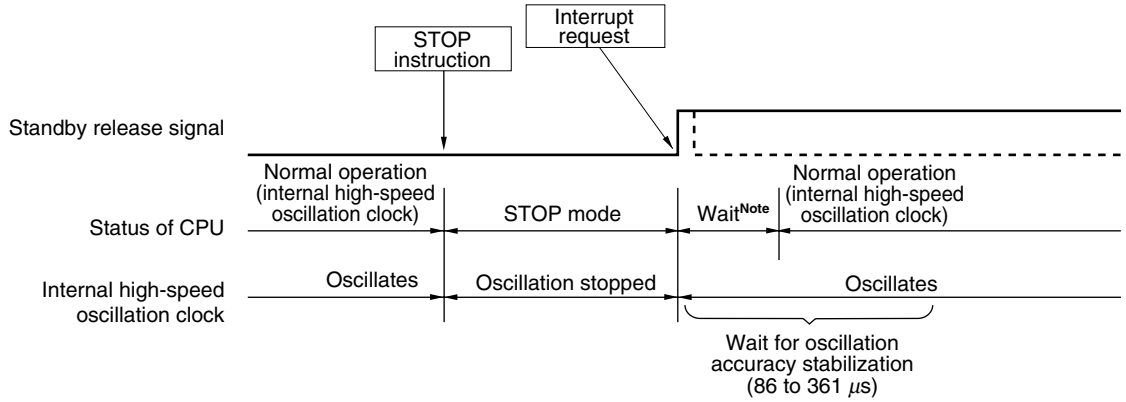


(3) When internal high-speed oscillation clock is used as CPU clock

- When AMPH = 1



- When AMPH = 0



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

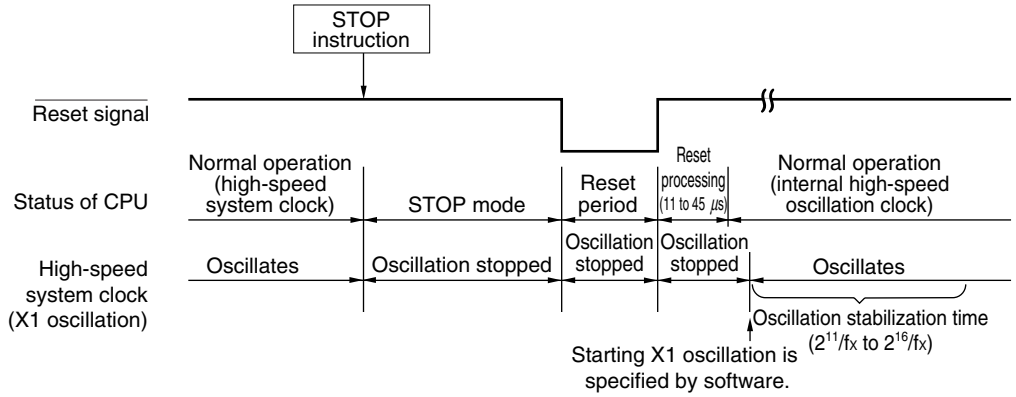
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

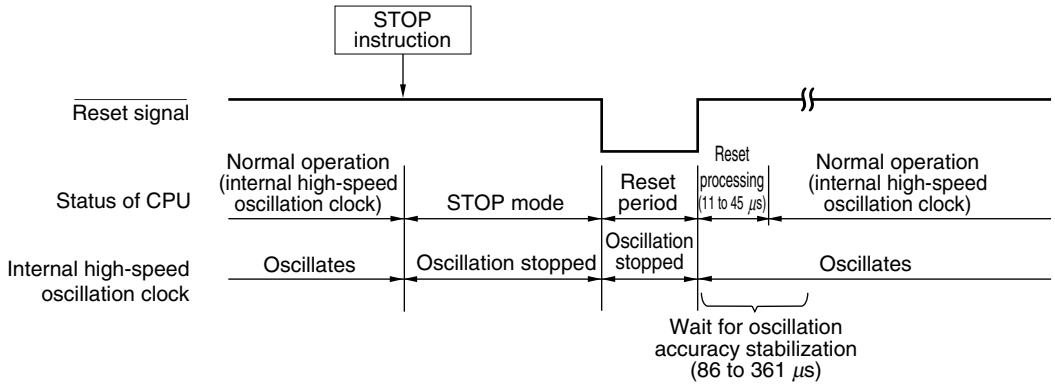
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 18-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Table 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 signal input	–	–	×	×	Reset processing

×: don't care

CHAPTER 19 RESET FUNCTION

The following four operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(4) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

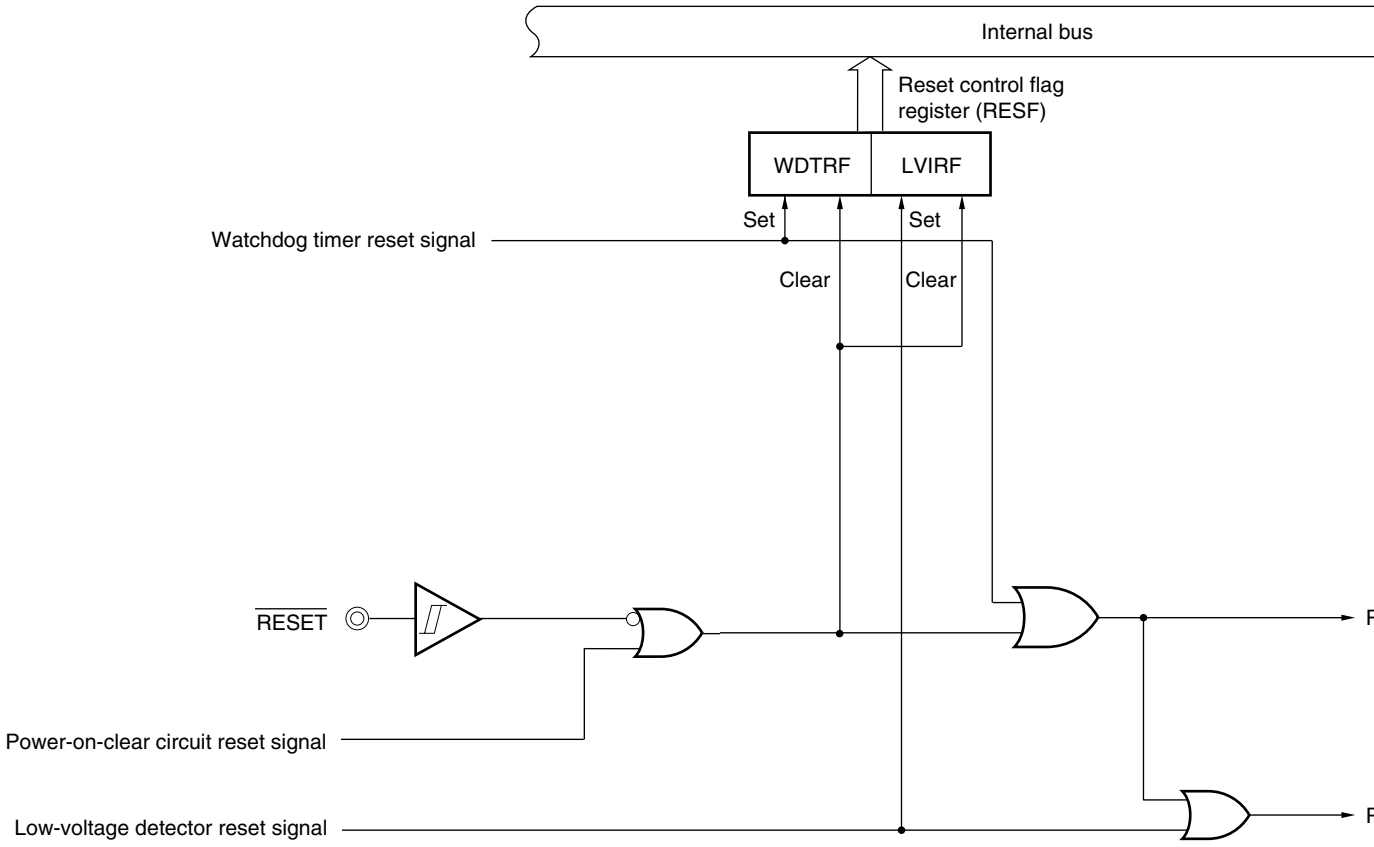
External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is generated.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Tables 19-1 and 19-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release, except for P130, which is low-level output.

When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 19-2** to **19-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**) after reset processing.

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input and external subsystem clock input become invalid.
3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance, except for P130, which is set to low-level output.

Figure 19-1. Block Diagram of Reset Function

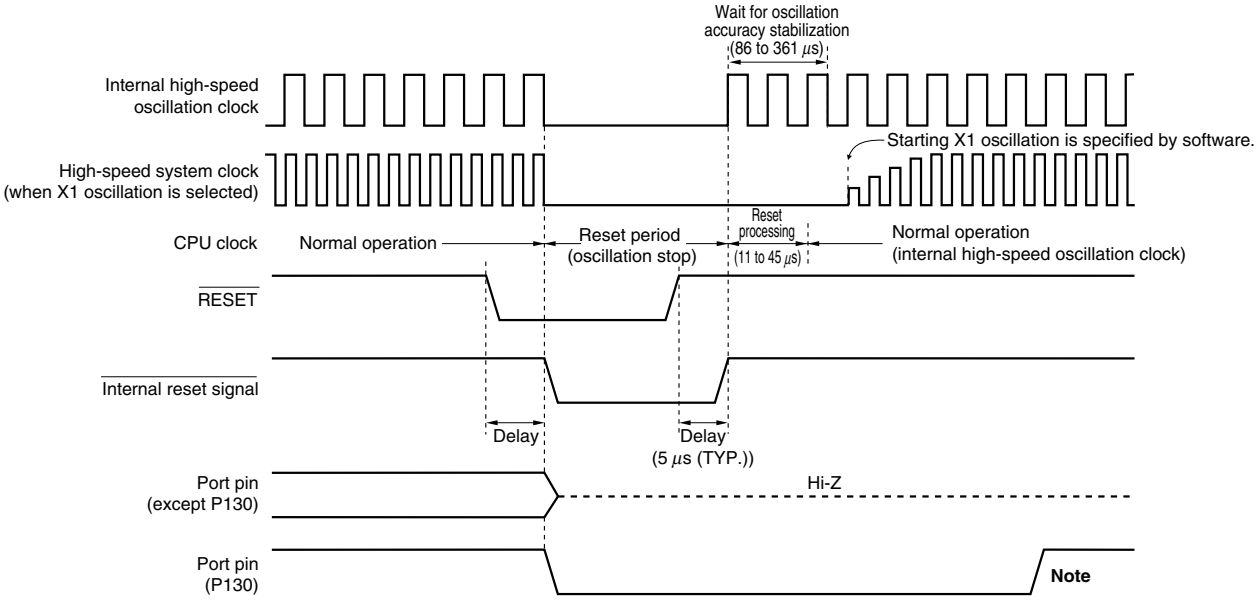


Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level selection register

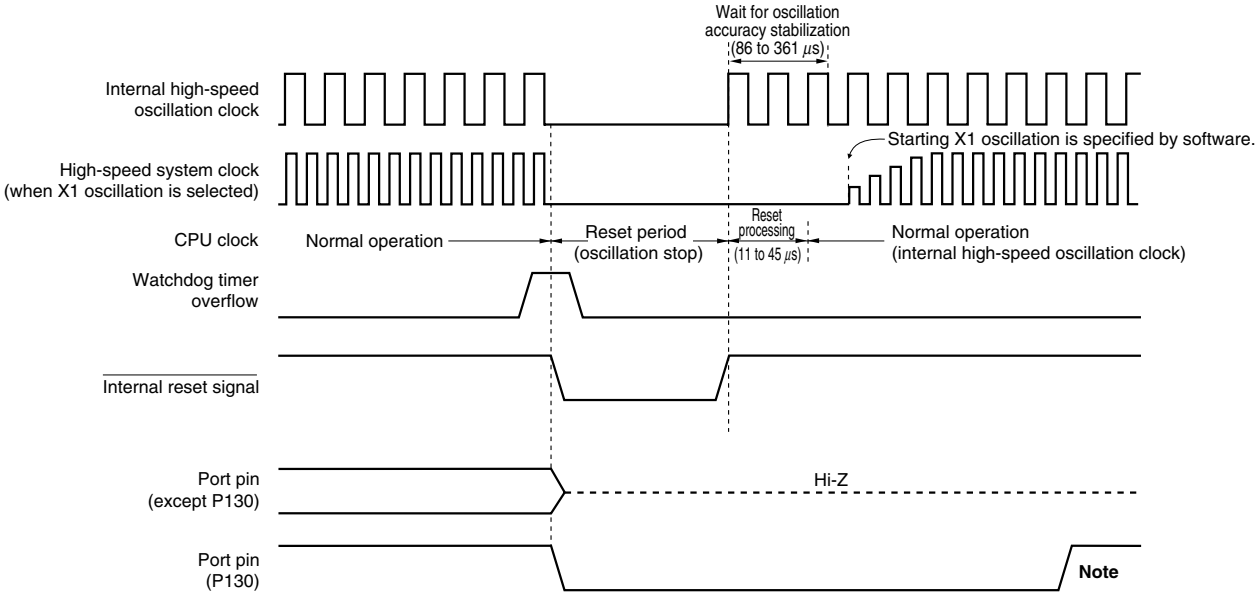
Figure 19-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

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

Figure 19-3. Timing of Reset Due to Watchdog Timer Overflow

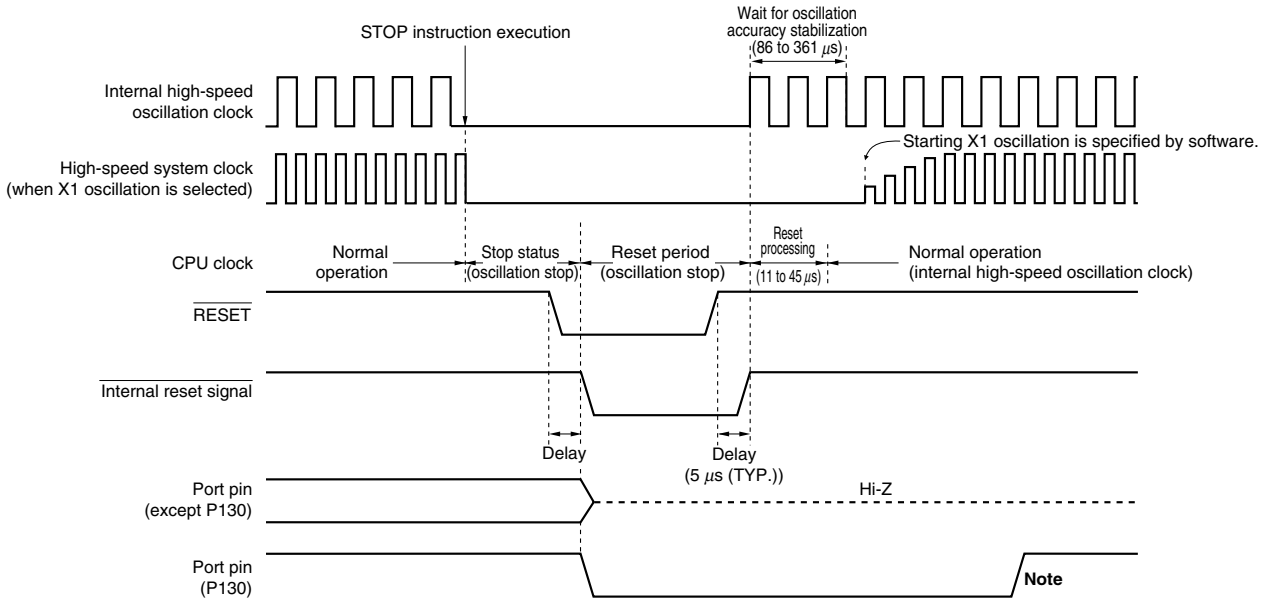


Note Set P130 to high-level output by software.

Caution A watchdog timer internal reset resets the watchdog timer.

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

Figure 19-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

Remarks 1. When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.

2. For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**.

Table 19-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{RH}	Operation stopped
		f_X	Operation stopped (pin is I/O port mode)
		f_{EXCLK}	Clock input invalid (pin is I/O port mode)
	Subsystem clock	f_{XT}	Operation stopped (pin is I/O port mode)
		f_{EXCLKS}	Clock input invalid (pin is I/O port mode)
	f_{RL}		Operation stopped
CPU			
Flash memory			
RAM			
Regulator			Operable
Port (latch)			Operation stopped
16-bit timer/event counter		00	
		01	
		02	
		03	
8-bit timer/event counter		50	
		51	
8-bit timer		H0	
		H1	
Watch timer			
Watchdog timer			
Clock output			
Buzzer output			
A/D converter			
Serial interface		UART60	
		UART61	
		CSI10	
		CSI11	
CAN controller			
Multiplier/divider			
Power-on-clear function			Operable
Low-voltage detection function			Operation stopped
External interrupt			

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 oscillation clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 oscillation clock
f_{EXCLKS}: External subsystem clock
f_{RL}: Internal low-speed oscillation clock

Table 19-2. Hardware Statuses After Reset Acknowledgment (1/3)

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0, P1, P3 to P9, P12, P13) (output latches)		00H
Port mode registers	PM0, PM1, PM3 to PM9, PM12	FFH
	PM13	FEH
Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU13)		00H
Internal expansion RAM size switching register (IXS)		0CH[Note 3]
Internal memory size switching register (IMS)		CFH[Note 3]
Bank select register (BANK)		00H
Processor clock control register (PCC)		01H
Clock operation mode select register (OSCCTL)		00H
Internal oscillator mode register (RCM)		00H[Note 4]
Main clock mode register (MCM)		00H
Main OSC control register (MOC)		80H
Oscillation stabilization time select register (OSTS)		05H
Oscillation stabilization time counter status register (OSTC)		00H
16-bit timer/event counters 00-03	Timer counters 00-03 (TM00-TM03)	0000H
	Capture/compare registers 000-003, 010-013(CR000-CR003, CR010-CR013)	0000H
	Mode control registers 00-03 (TMC00-TMC03)	00H
	Prescaler mode registers 00-03 (PRM00-PRM03)	00H
	Capture/compare control registers 00-03 (CRC00-CRC03)	00H
	Timer output control registers 00-03 (TOC00- TOC03)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. The initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) after a reset release are constant (IMS = CFH, IXS = 0CH) in all the 78K0/FE2 products, regardless of the internal memory capacity. Therefore, after a reset is released, be sure to set the following values for each product.

Flash Memory Version (78K0/FE2)	IMS	IXS
μPD78F0887	CCH	08H
μPD78F0888	CFH	08H
μPD78F0889	CCH	04H
μPD78F0890	CCH	00H

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

Table 19-2. Hardware Statuses After Reset Acknowledgment (2/3)

Hardware		Status After Reset Acknowledgment[Note 1]
8-bit timer/event counters 50, 51	Timer counters 50, 51 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	00H
	Timer clock selection registers 50, 51 (TCL50, TCL51)	00H
	Mode control registers 50, 51 (TMC50, TMC51)	00H
8-bit timers H0, H1	Compare registers 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	Mode registers (TMHMD0, TMHMD1)	00H
	Carrier control register 1 (TMCYC1)[Note 2]	00H
Watch timer	Operation mode register (WTM)	00H
Clock output/buzzer output controller	Clock output selection register (CKS)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 3]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	00H
Serial interface UART60, UART61	Receive buffer register 60, 61 (RXB60, RXB61)	FFH
	Transmit buffer register 60, 61 (TXB60, TXB61)	FFH
	Asynchronous serial interface operation mode register 60, 61 (ASIM60, ASIM61)	01H
	Asynchronous serial interface reception error status register 60, 61 (ASIS60, ASIS61)	00H
	Asynchronous serial interface transmission status register 60, 61 (ASIF60, ASIF61)	00H
	Clock selection register 60, 61 (CKSR60, CKSR61)	00H
	Baud rate generator control register 60, 61 (BRGC60, BRGC61)	FFH
	Asynchronous serial interface control register 60, 61 (ASICL60, ASICL61)	16H
	Input switch control register (ISC)	00H
Serial interfaces CSI10, CSI11	Transmit buffer registers 10, 11 (SOTB10, SOTB11)	00H
	Serial I/O shift registers 10, 11 (SIO10, SIO11)	00H
	Serial operation mode registers 10, 11 (CSIM10, CSIM11)	00H
	Serial clock selection registers 10, 11 (CSIC10, CSIC11)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. 8-bit timer H1 only.

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

Table 19-2. Hardware Statuses After Reset Acknowledgment (3/3)

Hardware		Status After Reset Acknowledgment[Note 1]
Multiplier/divider	Remainder data register 0 (SDR0)	0000H
	Multiplication/division data register A0 (MDA0H, MDA0L)	0000H
	Multiplication/division data register B0 (MDB0)	0000H
	Multiplier/divider control register 0 (DMUC0)	00H
Reset function	Reset control flag register (RESF)	00H[Note2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note2]
	Low-voltage detection level selection register (LVIS)	00H[Note2]
Interrupt	Request flag registers 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	Mask flag registers 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	Priority specification flag registers 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2 These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
RESF	WDTRF bit	Cleared (0)	Cleared (0)	Set (1)	Held
	LVIRF bit			Held	Set (1)
LVIM		Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS					

19.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/FE2. 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 input by power-on-clear (POC) circuit, and reading RESF clear 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

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held
LVIRF			Held	Set (1)

CHAPTER 20 MULTIPLIER/DIVIDER

20.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 16 bits = 32 bits, 16-bit remainder (division)

20.2 Configuration of Multiplier/Divider

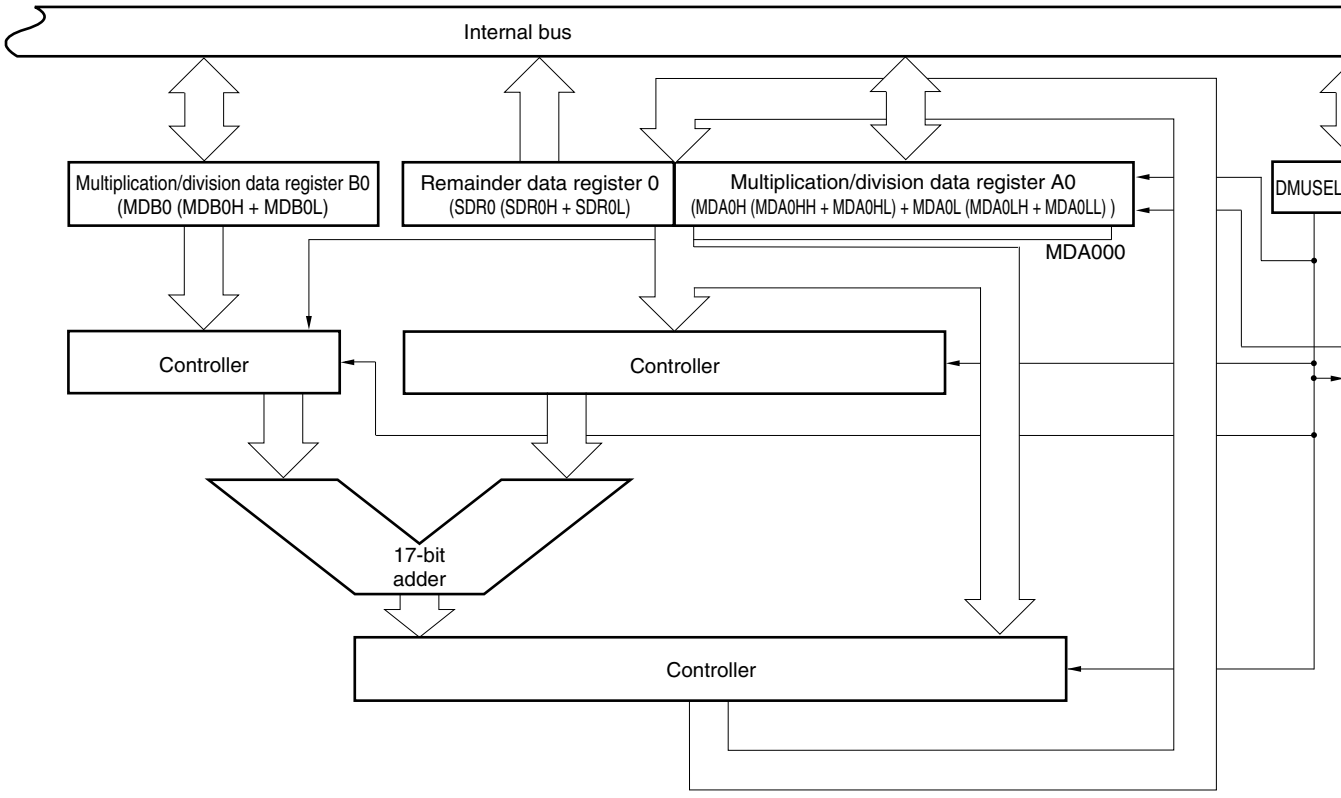
The multiplier/divider includes the following hardware.

Table 20-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Remainder data register 0 (SDR0) Multiplication/division data registers A0 (MDA0H, MDA0L) Multiplication/division data registers B0 (MDB0)
Control register	Multiplier/divider control register 0 (DMUC0)

Figure 20-1 shows the block diagram of the multiplier/divider.

Figure 20-1. Block Diagram of Multiplier/Divider



(1) Remainder data register 0 (SDR0)

SDR0 is a 16-bit register that stores a remainder. This register stores 0 in the multiplication mode and the remainder of an operation result in the division mode.

SDR0 can be read by an 8-bit or 16-bit memory manipulation instruction.

Reset signal generation clears SDR0 to 0000H.

Figure 20-2. Format of Remainder Data Register 0 (SDR0)

Address: FF44H, FF45H After reset: 0000H R

Symbol	FF45H (SDR0H)								FF44H (SDR0L)							
SDR0	SDR 015	SDR 014	SDR 013	SDR 012	SDR 011	SDR 010	SDR 009	SDR 008	SDR 007	SDR 006	SDR 005	SDR 004	SDR 003	SDR 002	SDR 001	SDR 000

Cautions 1. The value read from SDR0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1) is not guaranteed.

2. SDR0 is reset when the operation is started (when DMUE is set to 1).

(2) Multiplication/division data register A0 (MDA0H, MDA0L)

MDA0 is a 32-bit register that sets a 16-bit multiplier A in the multiplication mode and a 32-bit dividend in the division mode, and stores the 32-bit result of the operation (higher 16 bits: MDA0H, lower 16 bits: MDA0L).

Figure 20-3. Format of Multiplication/Division Data Register A0 (MDA0H, MDA0L)

Address: FF4AH, FF4BH, FF4CH, FF4DH After reset: 0000H, 0000H R/W

Symbol	FF4DH (MDA0HH)								FF4CH (MDA0HL)							
MDA0H	MDA 031	MDA 030	MDA 029	MDA 028	MDA 027	MDA 026	MDA 025	MDA 024	MDA 023	MDA 022	MDA 021	MDA 020	MDA 019	MDA 018	MDA 017	MDA 016

Symbol	FF4BH (MDA0LH)								FF4AH (MDA0LL)							
MDA0L	MDA 015	MDA 014	MDA 013	MDA 012	MDA 011	MDA 010	MDA 009	MDA 008	MDA 007	MDA 006	MDA 005	MDA 004	MDA 003	MDA 002	MDA 001	MDA 000

Cautions 1. MDA0H is cleared to 0 when an operation is started in the multiplication mode (when multiplier/divider control register 0 (DMUC0) is set to 81H).

2. Do not change the value of MDA0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

3. The value read from MDA0 during operation processing (while DMUE is 1) is not guaranteed.

The functions of MDA0 when an operation is executed are shown in the table below.

Table 20-2. Functions of MDA0 During Operation Execution

DMUSEL0	Operation Mode	Setting	Operation Result
0	Division mode	Dividend	Division result (quotient)
1	Multiplication mode	Higher 16 bits: 0, Lower 16 bits: Multiplier A	Multiplication result (product)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDA0 (bits 15 to 0) × MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0)

- Register configuration during division

  <Dividend> <Divisor> <Quotient> <Remainder>

  MDA0 (bits 31 to 0) ÷ MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0) … SDR0 (bits 15 to 0)

MDA0 fetches the calculation result as soon as the clock is input, when bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is set to 1.
MDA0H and MDA0L can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDA0H and MDA0L to 0000H.

(3) Multiplication/division data register B0 (MDB0)

MDB0 is a register that stores a 16-bit multiplier B in the multiplication mode and a 16-bit divisor in the division mode.
MDB0 can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDB0 to 0000H.

Figure 20-4. Format of Multiplication/Division Data Register B0 (MDB0)

Address: FFAEH, FFAFH After reset: 0000H R/W

Symbol	FFAFH (MDB0H)								FFAEH (MDB0L)							
MDB0	MDB 015	MDB 014	MDB 013	MDB 012	MDB 011	MDB 010	MDB 009	MDB 008	MDB 007	MDB 006	MDB 005	MDB 004	MDB 003	MDB 002	MDB 001	MDB 000

Cautions 1. Do not change the value of MDB0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

2. Do not clear MDB0 to 0000H in the division mode. If set, undefined operation results are stored in MDA0 and SDR0.

20.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by multiplier/divider control register 0 (DMUC0).

(1) Multiplier/divider control register 0 (DMUC0)

DMUC0 is an 8-bit register that controls the operation of the multiplier/divider.

DMUC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears DMUC0 to 00H.

Figure 20-5. Format of Multiplier/Divider Control Register 0 (DMUC0)

Address: FF42H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0

DMUE[Note]	Operation start/stop
0	Stops operation
1	Starts operation

DMUSEL0	Operation mode (multiplication/division) selection
0	Division mode
1	Multiplication mode

Note When DMUE is set to 1, the operation is started. DMUE is automatically cleared to 0 after the operation is complete.

Cautions 1. If DMUE is cleared to 0 during operation processing (when DMUE is 1), the operation result is not guaranteed. If the operation is completed while the clearing instruction is being executed, the operation result is guaranteed, provided that the interrupt flag is set.

2. Do not change the value of DMUSEL0 during operation processing (while DMUE is 1). If it is changed, undefined operation results are stored in multiplication/division data register A0 (MDA0) and remainder data register 0 (SDR0).

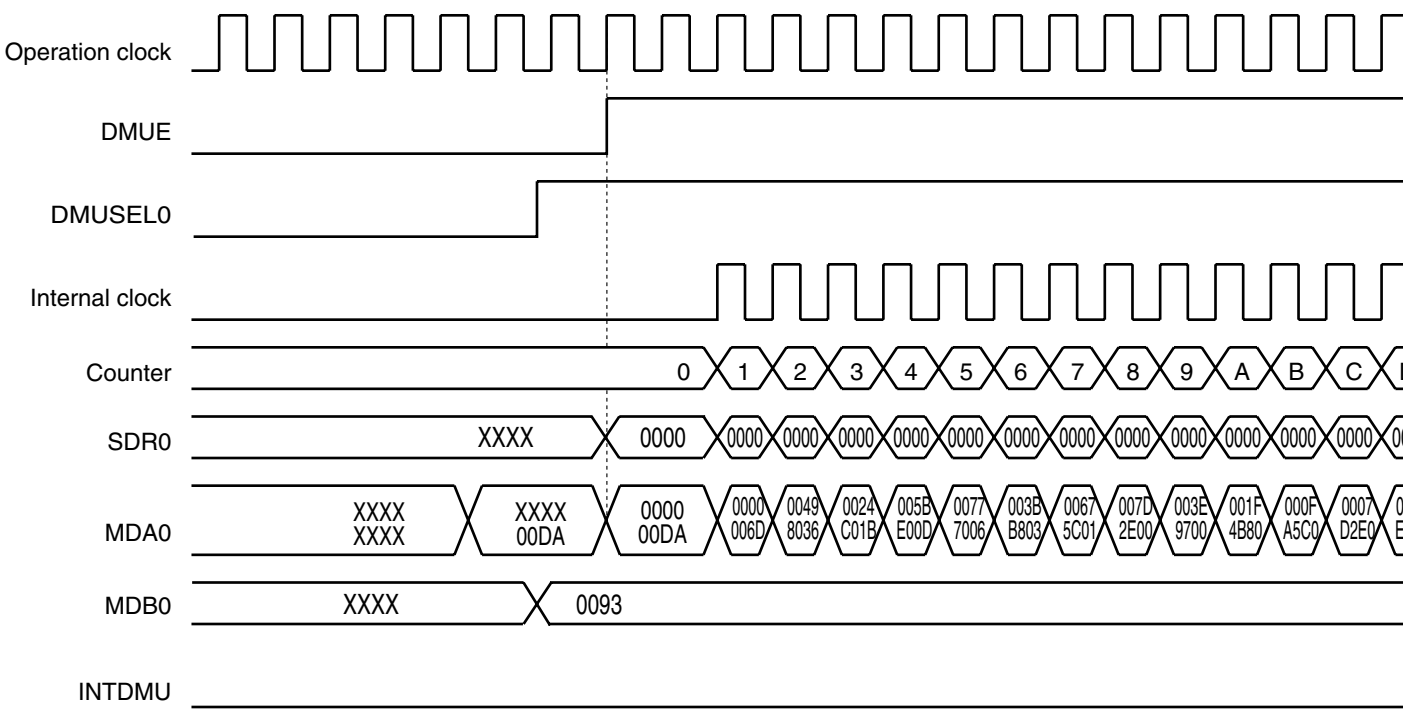
3. If DMUE is cleared to 0 during operation processing (while DMUE is 1), the operation processing is stopped. To execute the operation again, set multiplication/division data register A0 (MDA0), multiplication/division data register B0 (MDB0), and multiplier/divider control register 0 (DMUC0), and start the operation (by setting DMUE to 1).

20.4 Operations of Multiplier/Divider

20.4.1 Multiplication operation

- Initial setting
  1. Set operation data to multiplication/division data register A0L (MDA0L) and multiplication/division data register B0 (MDB0).
  2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 1. Operation will start.
- During operation
  3. The operation will be completed when 16 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers during operation, and therefore the read values of these registers are not guaranteed).
- End of operation
  4. The operation result data is stored in the MDA0L and MDA0H registers.
  5. DMUE is cleared to 0 (end of operation).
  6. After the operation, an interrupt request signal (INTDMU) is generated.
- Next operation
  7. To execute multiplication next, start from the initial setting in **20.4.1 Multiplication operation**.
  8. To execute division next, start from the initial setting in **20.4.2 Division operation**.

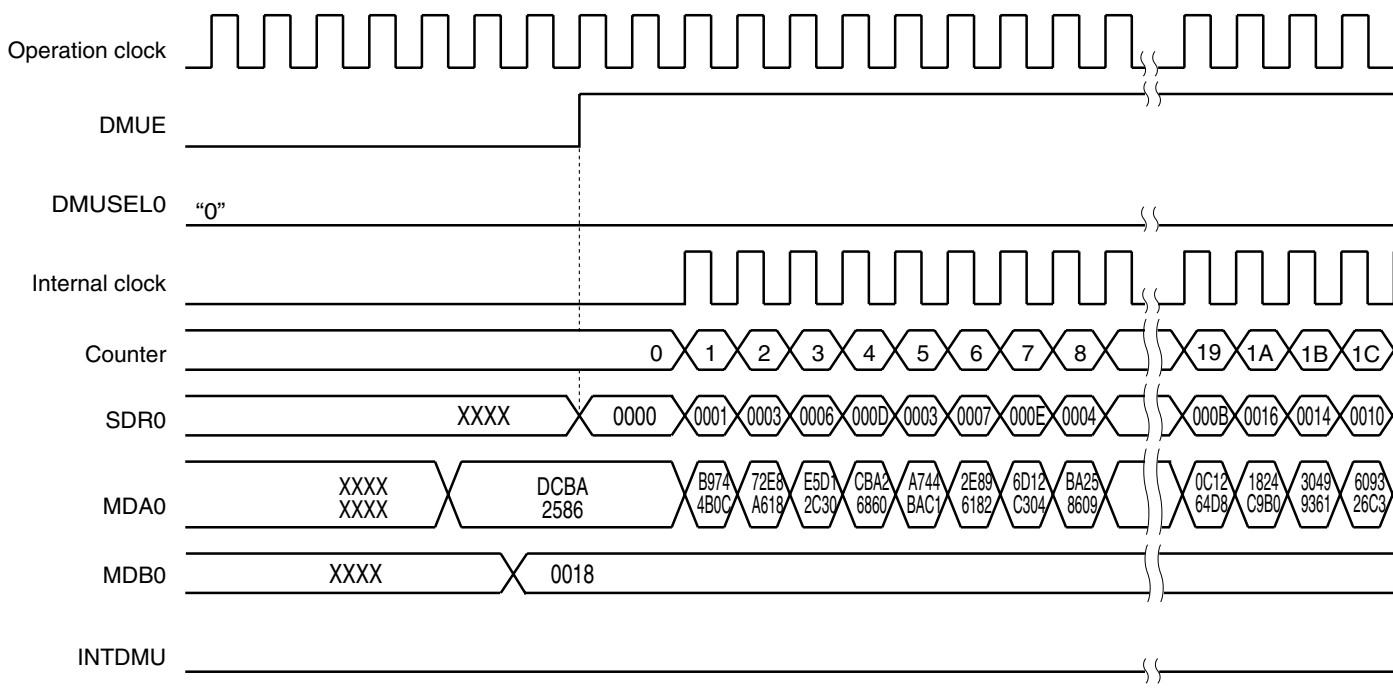
Figure 20-6. Timing Chart of Multiplication Operation (00DAH × 0093H)



20.4.2 Division operation

- Initial setting
  1. Set operation data to multiplication/division data register A0 (MDA0L and MDA0H) and multiplication/division data register B0 (MDB0).
  2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 0 and 1, respectively. Operation will start.
- During operation
  3. The operation will be completed when 32 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers and remainder data register 0 (SDR0) during operation, and therefore the read values of these registers are not guaranteed).
- End of operation
  4. The result data is stored in the MDA0L, MDA0H, and SDR0 registers.
  5. DMUE is cleared to 0 (end of operation).
  6. After the operation, an interrupt request signal (INTDMU) is generated.
- Next operation
  7. To execute multiplication next, start from the initial setting in **20.4.1 Multiplication operation**.
  8. To execute division next, start from the initial setting in **20.4.2 Division operation**.

Figure 20-7. Timing Chart of Division Operation (DCBA2586H ÷ 0018H)



CHAPTER 21 POWER-ON-CLEAR CIRCUIT

21.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
  In the 1.59 V POC mode (option byte: LVISTART = 0), the reset signal is released when the supply voltage (V_{DD}) exceeds 1.59 V ±0.15 V.
  In the 2.7 V/1.59 V POC mode (option byte: LVISTART = 1), the reset signal is released when the supply voltage (V_{DD}) exceeds 2.7 V ±0.2 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V), generates internal reset signal when $V_{DD} < V_{POC}$.

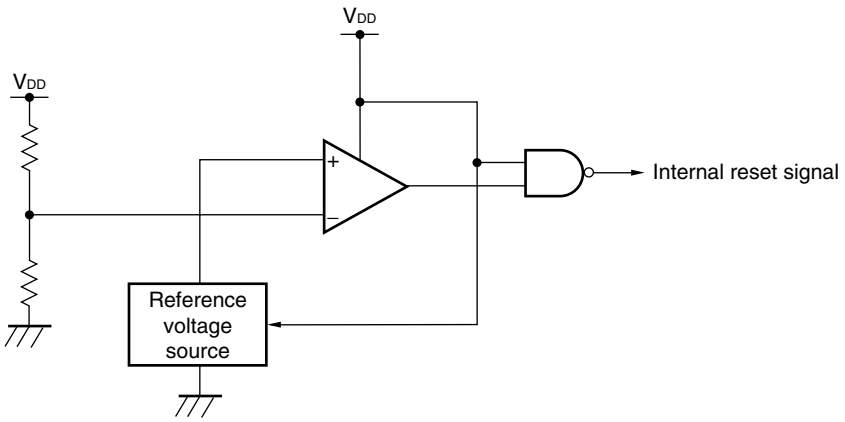
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark The 78K0/FE2 incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset cause is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT) or low-voltage-detector (LVI). RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT or LVI. For details of RESF, see **CHAPTER 19 RESET FUNCTION**.

21.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 21-1.

Figure 21-1. Block Diagram of Power-on-Clear Circuit



21.3 Operation of Power-on-Clear Circuit

(1) In 1.59 V POC mode (option byte: LVISTART = 0)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POC} = 1.59 V ±0.15 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{POC}$.

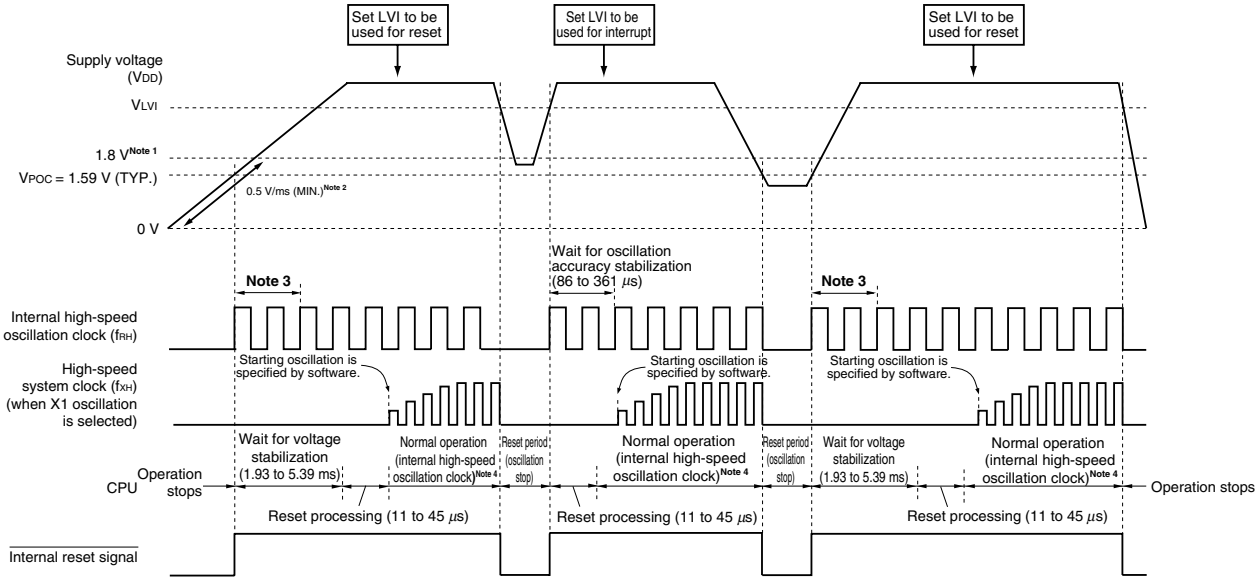
(2) In 2.7 V/1.59 V POC mode (option byte: LVISTART = 1)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{DDPOC} = 2.7 V ±0.2 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{DDPOC}$.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) In 1.59 V POC mode (option byte: 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 voltage rises to 1.8 V at a rate slower than 0.5 V/ms (MIN.) on power application, input a low level to the $\overline{\text{RESET}}$ pin after power application and before the voltage reaches 1.8 V, or set the 2.7 V/1.59 V POC mode by using an option byte (LVISTART = 1).

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

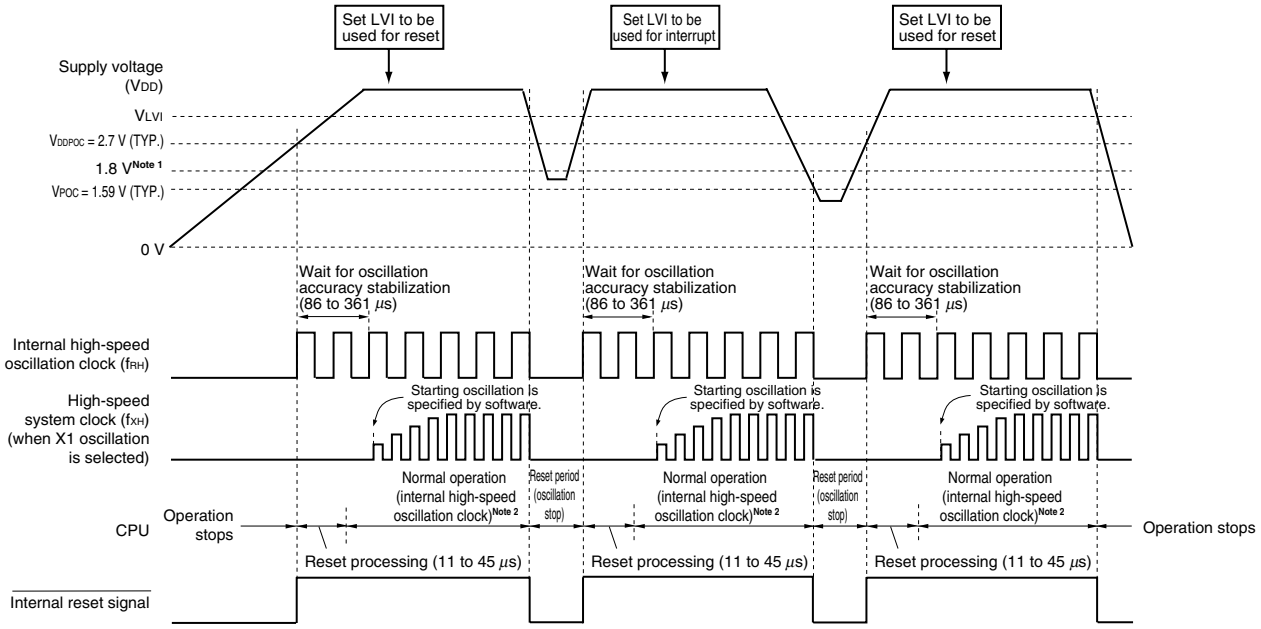
4. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

Remark V_{LVI} : LVI detection voltage
V_{POC} : POC detection voltage

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) In 2.7 V / 1.59V POC mode (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 internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

2. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) within 1.93 ms, the power supply oscillation stabilization time of 0 to 5.39 ms is automatically generated before reset processing.

Remark V_{LVI} : LVI detection voltage
V_{POC} : POC detection voltage

21.4 Cautions for Power-on-Clear Circuit

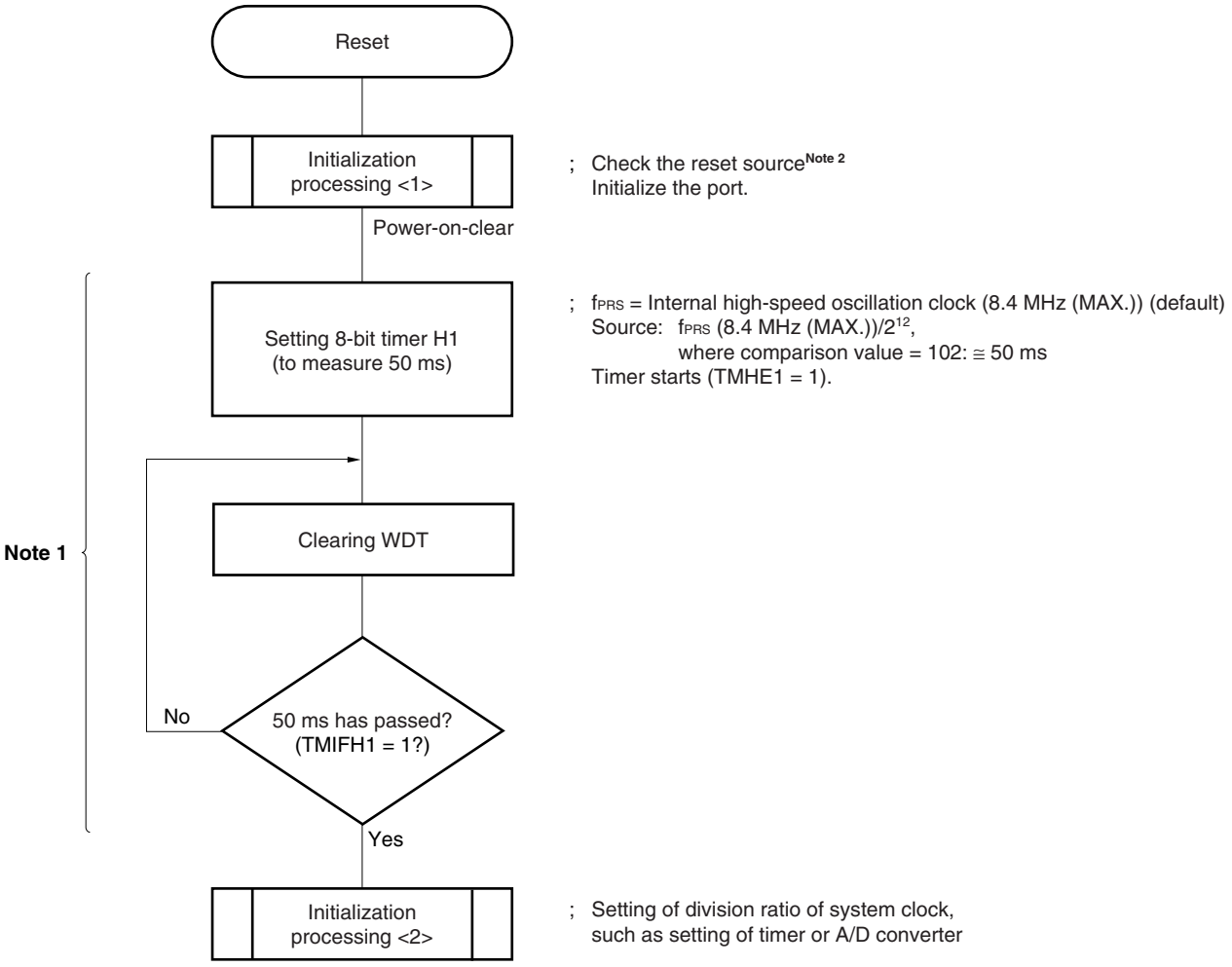
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 21-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

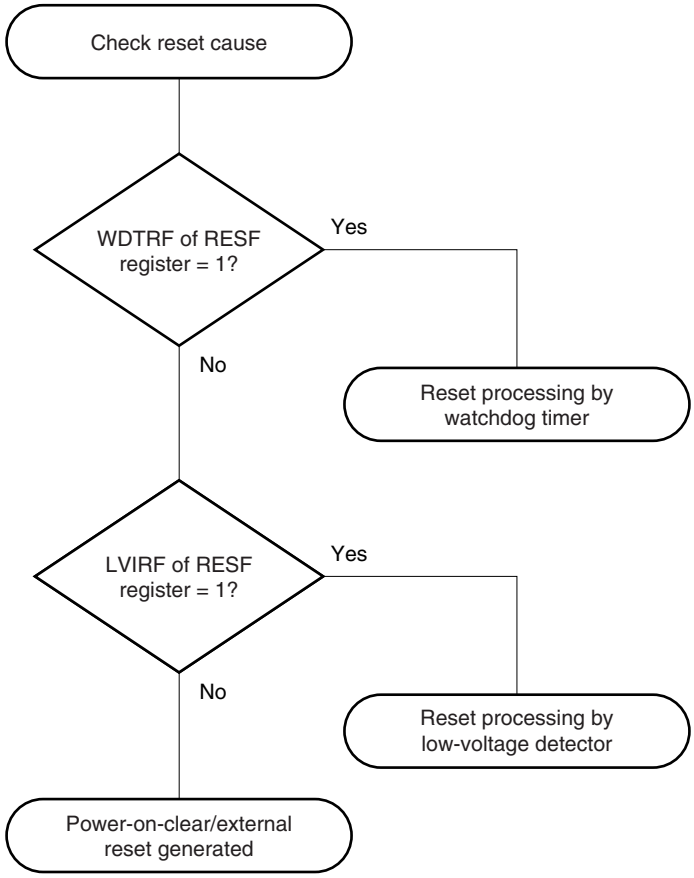


Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Figure 21-3. Example of Software Processing After Release of Reset (2/2)

• Checking reset cause



CHAPTER 22 LOW-VOLTAGE DETECTOR

22.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has the following functions.

- The LVI circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVI}) or the input voltage from an external input pin (EXLVI) with the detection voltage (V_{EXLVI} = 1.21 V (TYP.): fixed), and generates an internal reset or internal interrupt signal.
- The supply voltage (V_{DD}) or input voltage from an external input pin (EXLVI) can be selected by software.
- Reset or interrupt function can be selected by software.
- Detection levels (16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage (V_{DD}) (LVISEL = 0)		Selection Level Detection of Input Voltage from External Input Pin (EXLVI) (LVISEL = 1)	
Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).	Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).
Generates an internal reset signal when $V_{DD} < V_{LVI}$ and releases the reset signal when $V_{DD} \geq V_{LVI}$.	Generates an internal interrupt signal when V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).	Generates an internal reset signal when EXLVI < V_{EXLVI} and releases the reset signal when EXLVI ≥ V_{EXLVI}.	Generates an internal interrupt signal when EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).

Remark LVISEL: Bit 2 of low-voltage detection register (LVIM)
LVIMD: Bit 1 of LVIM

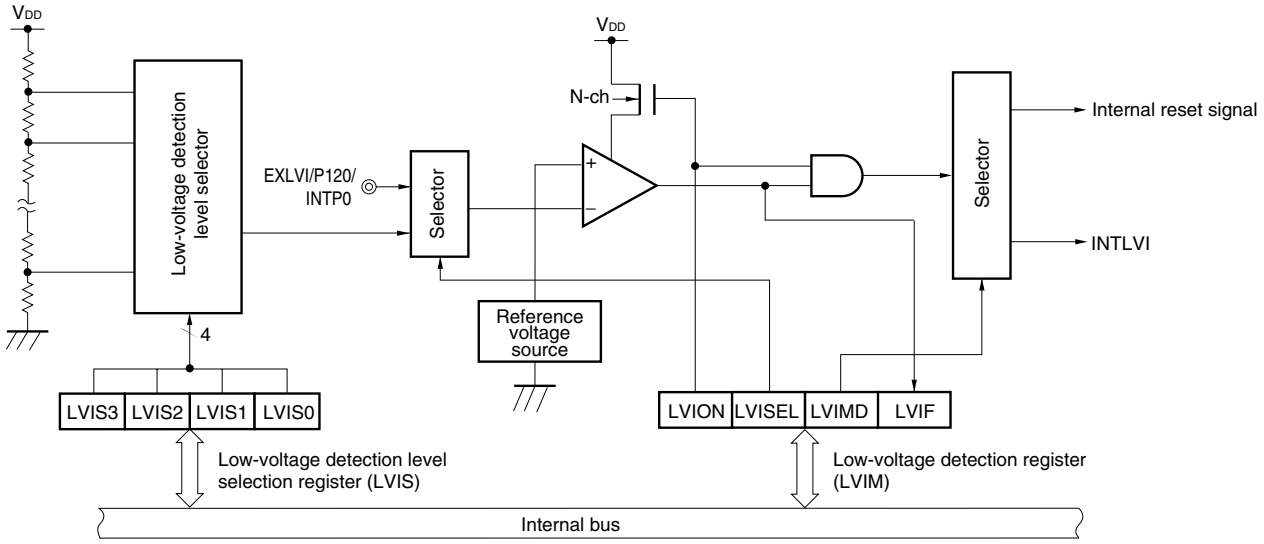
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 19 RESET FUNCTION**.

22.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 22-1.

Figure 22-1. Block Diagram of Low-Voltage Detector



22.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level selection register (LVIS)
- Port mode register 12 (PM12)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

Reset signal generation clears LVIM to 00H.

Figure 22-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFBEH After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION[Notes 2, 3]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL[Note 2]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD[Note 2]	Low-voltage detection operation mode (interrupt/reset) selection
0	• LVISEL = 0: Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$). • LVISEL = 1: Generates an interrupt signal when the input voltage from an external input pin (EXLVI) drops lower than the detection voltage (V_{EXLVI}) (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).
1	• LVISEL = 0: Generates an internal reset signal when the supply voltage (V_{DD}) < detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$. • LVISEL = 1: Generates an internal reset signal when the input voltage from an external input pin (EXLVI) < detection voltage (V_{EXLVI}) and releases the reset signal when EXLVI ≥ V_{EXLVI}.

LVIF[Note 4]	Low-voltage detection flag
0	• LVISEL = 0: Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}), or when operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when operation is disabled
1	• LVISEL = 0: Supply voltage (V_{DD}) < detection voltage (V_{LVI}) • LVISEL = 1: Input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI})

Notes 1. Bit 0 is read-only.

2. LVION, LVIMD, and LVISEL are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

3. 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 and minimum pulse width (10 μs (MAX.)) when LVION is set to 1 until the voltage is confirmed at LVIF.

4. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

Cautions 1. To stop LVI, follow either of the procedures below.

- **When using 8-bit memory manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

2. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

(2) Low-voltage detection level selection register (LVIS)

This register selects the low-voltage detection level.

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

Reset signal generation clears LVIS to 00H.

Figure 22-3. Format of Low-Voltage Detection Level Selection Register (LVIS)

Address: FFBFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	0	0	0	V_{LVI0} (4.24 V ±0.1 V)
0	0	0	1	V_{LVI1} (4.09 V ±0.1 V)
0	0	1	0	V_{LVI2} (3.93 V ±0.1 V)
0	0	1	1	V_{LVI3} (3.78 V ±0.1 V)
0	1	0	0	V_{LVI4} (3.62 V ±0.1 V)
0	1	0	1	V_{LVI5} (3.47 V ±0.1 V)
0	1	1	0	V_{LVI6} (3.32 V ±0.1 V)
0	1	1	1	V_{LVI7} (3.16 V ±0.1 V)
1	0	0	0	V_{LVI8} (3.01 V ±0.1 V)
1	0	0	1	V_{LVI9} (2.85 V ±0.1 V)
1	0	1	0	V_{LVI10} (2.70 V ±0.1 V)
1	0	1	1	V_{LVI11} (2.55 V ±0.1 V)
1	1	0	0	V_{LVI12} (2.39 V ±0.1 V)
1	1	0	1	V_{LVI13} (2.24 V ±0.1 V)
1	1	1	0	V_{LVI14} (2.08 V ±0.1 V)
1	1	1	1	V_{LVI15} (1.93 V ±0.1 V)

Cautions 1. Be sure to clear bits 4 to 7 to 0.

2. Do not change the value of LVIS during LVI operation.

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI} = 1.21 V (TYP.)) is fixed. Therefore, setting of LVIS is not necessary.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM12 to FFH.

Figure 22-4. Format of Port Mode Register 12 (PM12)

Address: FF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	PM124	PM123	PM122	PM121	PM120

PM12n	P12n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

22.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V (TYP.)), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

(2) Used as interrupt (LVIMD = 0)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}). When V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$), generates an interrupt signal (INTLVI).
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V (TYP.)). When EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}), generates an interrupt signal (INTLVI).

While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

Remark LVIMD: Bit 1 of low-voltage detection register (LVIM)
LVISEL: Bit 2 of LVIM

22.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
  - <3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for an operation stabilization time and minimum pulse width (10 μs (MAX.)).
  - <6> Wait until it is checked that (supply voltage (V_{DD}) $\geq$ detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.
  - <7> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

  Figure 22-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

  Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

  2. If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

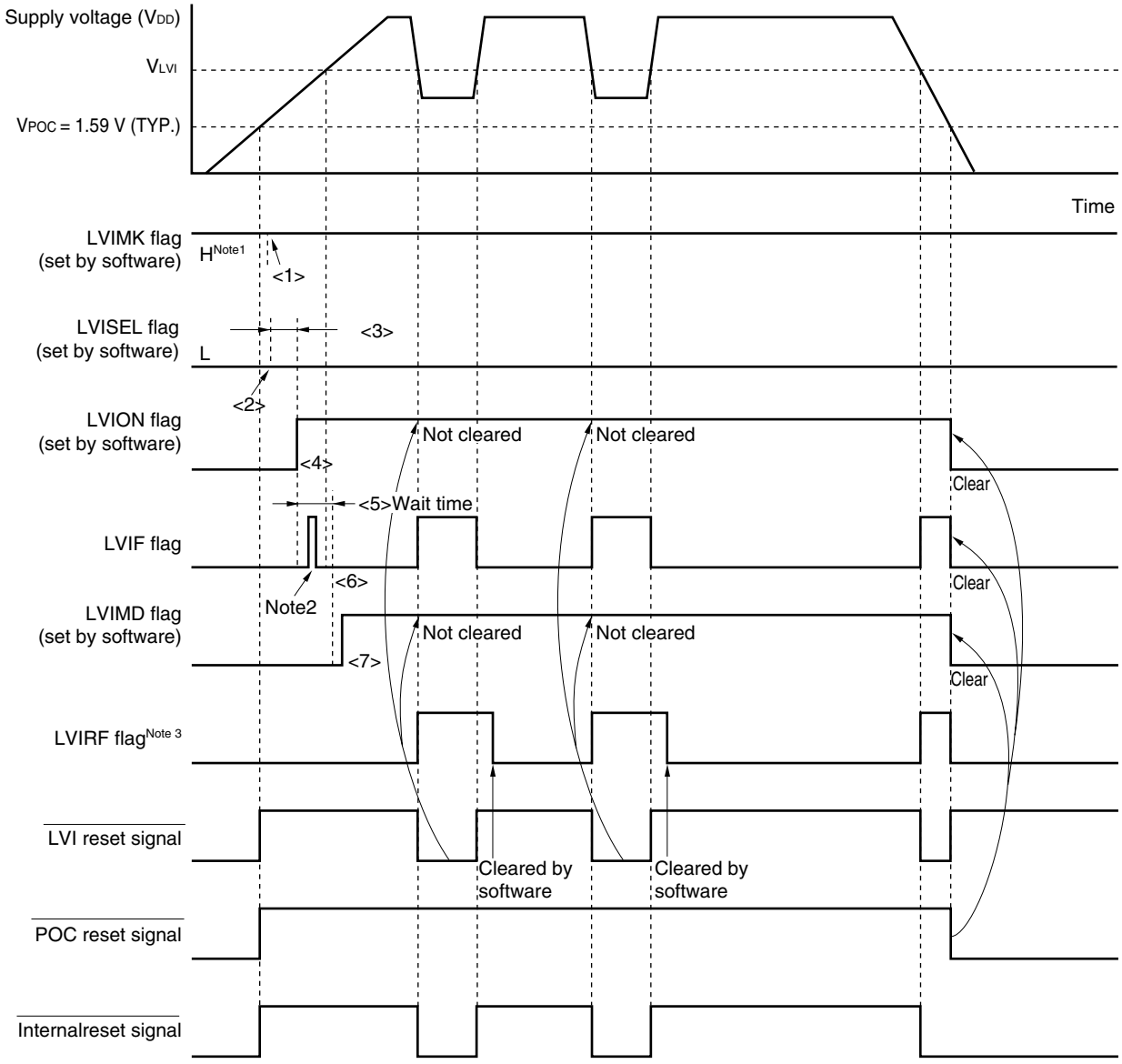
- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVIMD to 0 and then LVION to 0.

Figure 22-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode setup (option byte: LVISTART = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

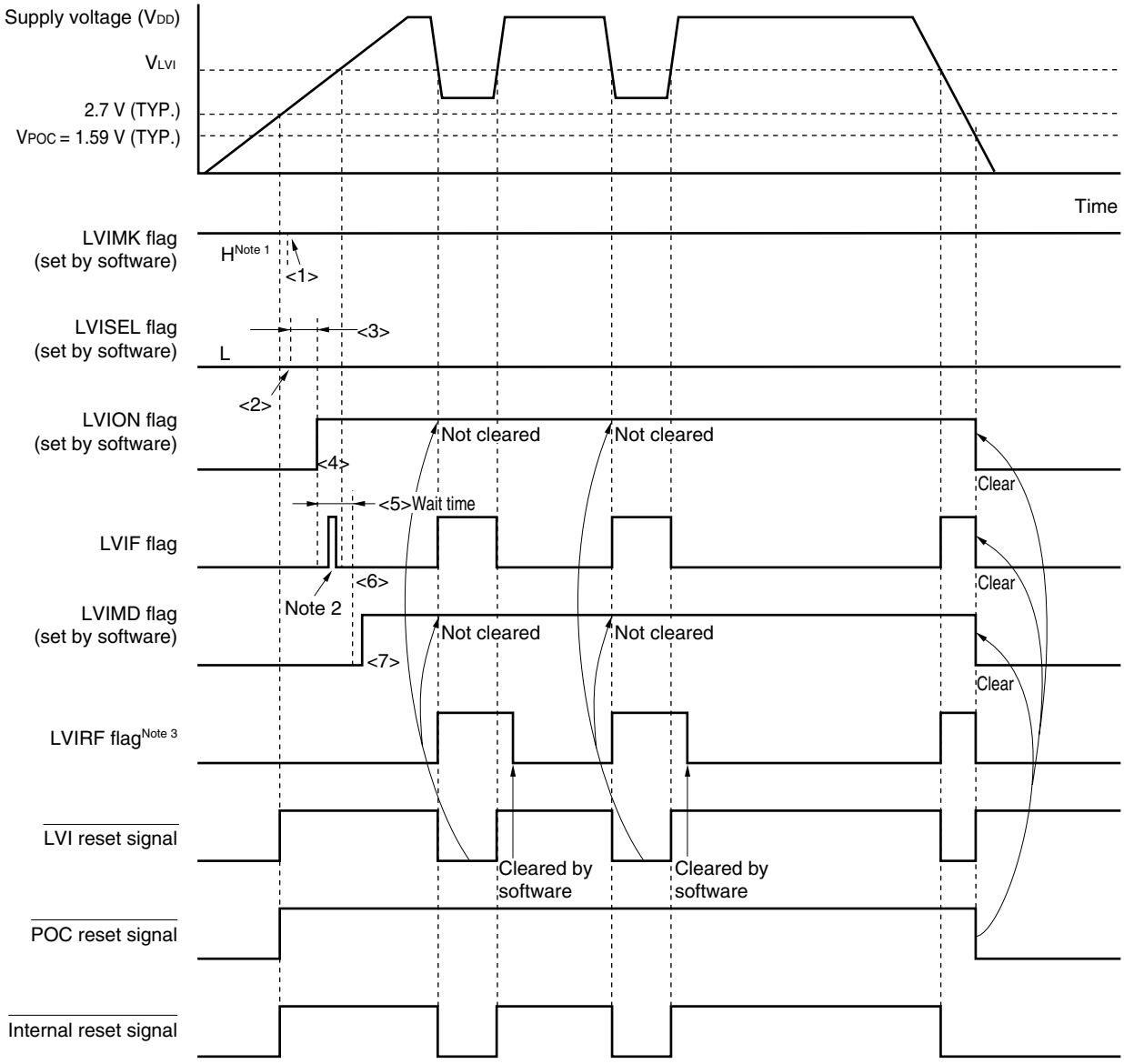
2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 19 RESET FUNCTION**.

Remark <1> to <7> in Figure 22-5 above correspond to <1> to <7> in the description of "When starting operation" in **22.4.1 (1) When detecting level of supply voltage (V_{DD})**.

Figure 22-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7/1.59 V POC mode setup (option byte: LVISTART = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 19 RESET FUNCTION**.

Remark <1> to <7> in Figure 22-5 above correspond to <1> to <7> in the description of "When starting operation" in **22.4.1 (1) When detecting level of supply voltage (V_{DD})**.

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for an operation stabilization time and minimum pulse width (10 μs (MAX.)).
  - <5> Wait until it is checked that (input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))) by bit 0 (LVIF) of LVIM.
  - <6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset signal when the level is detected).

  Figure 22-6 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

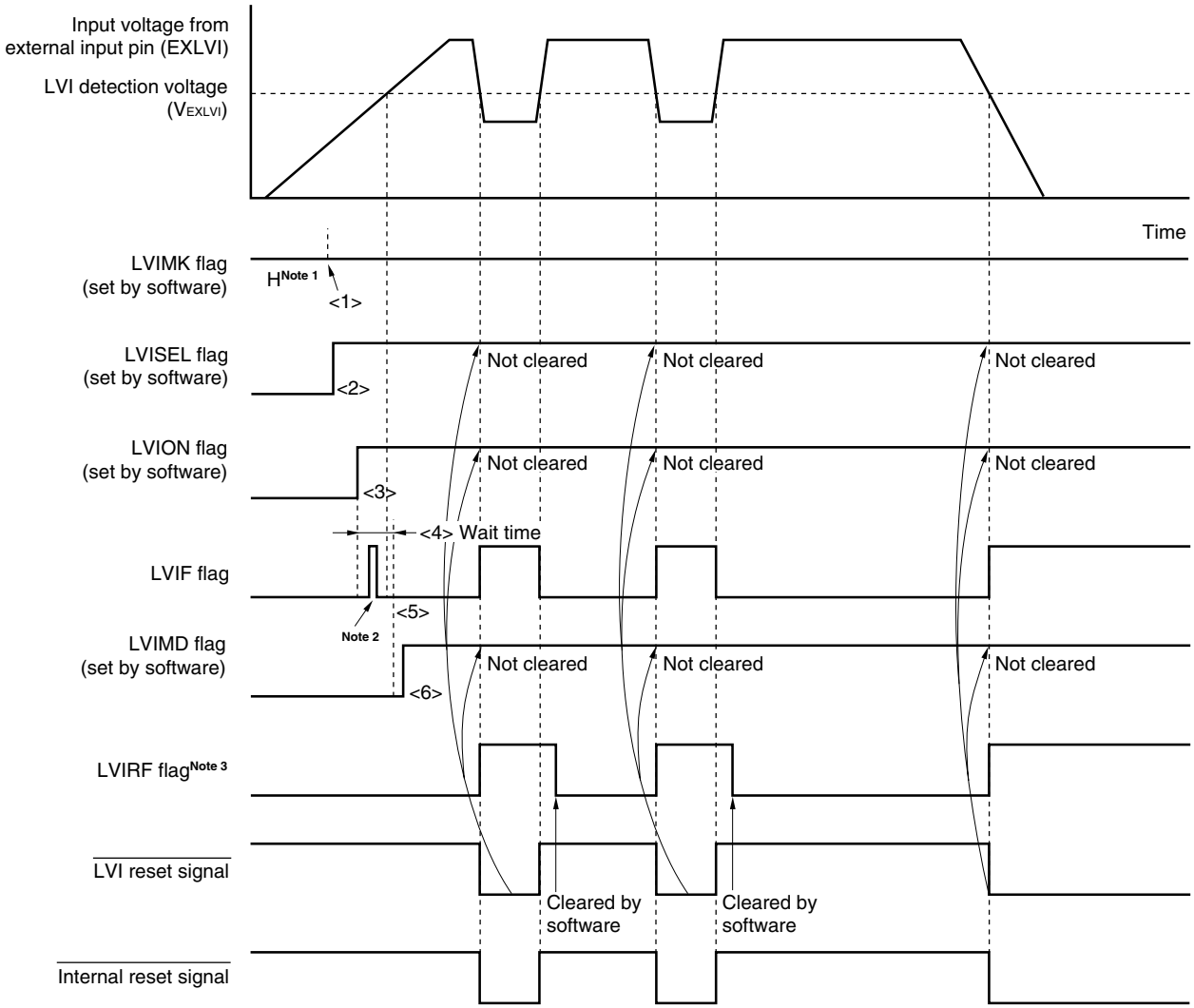
  Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.
  2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.
  3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVIMD to 0 and then LVION to 0.

Figure 22-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 19 RESET FUNCTION**.

Remark <1> to <6> in Figure 22-6 above correspond to <1> to <6> in the description of "When starting operation" in **22.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

22.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
  - <3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for an operation stabilization time and minimum pulse width (10 μs (MAX.)).
  - <6> Confirm that "supply voltage (V_{DD}) $\geq$ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, at bit 0 (LVIF) of LVIM.
  - <7> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <8> Release the interrupt mask flag of LVI (LVIMK).
  - <9> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <10> Execute the EI instruction (when vector interrupts are used).

  Figure 22-7 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <9> above.

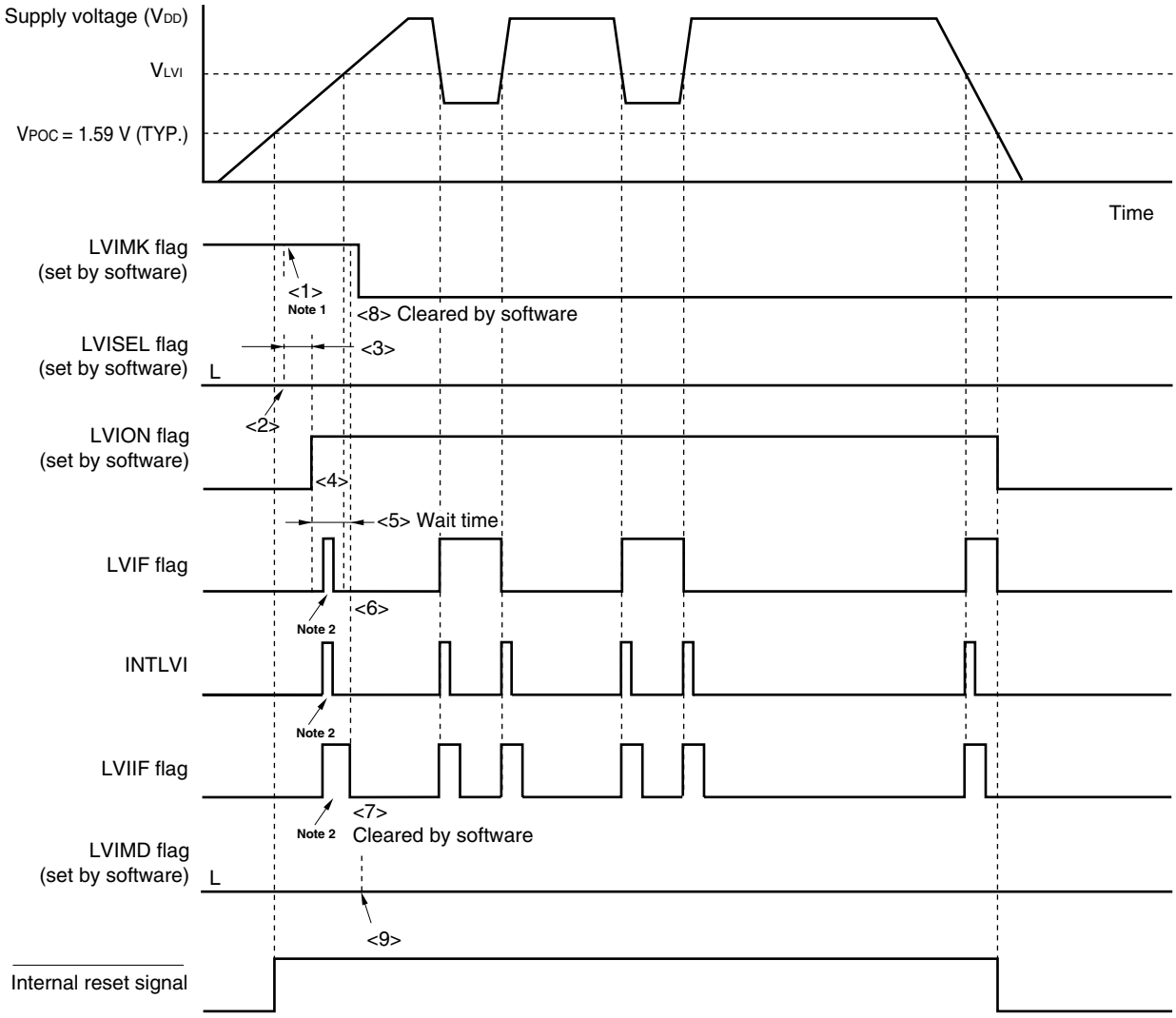
- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 22-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode setup (option byte: LVISTART = 0)



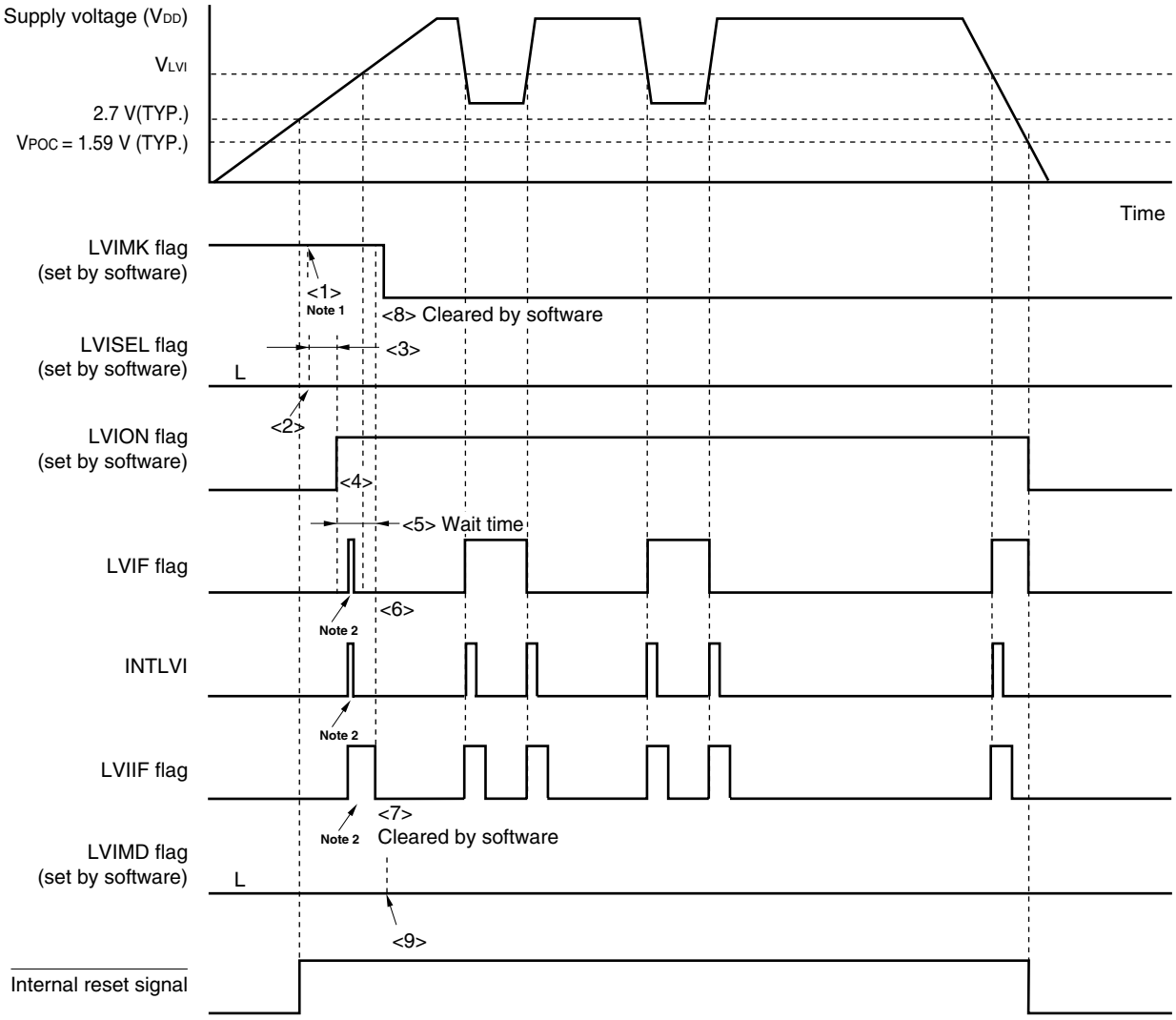
Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

Remark <1> to <9> in Figure 22-7 above correspond to <1> to <9> in the description of "When starting operation" in **22.4.2 (1) When detecting level of supply voltage (V_{DD})**.

Figure 22-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7/1.59 V POC mode setup (option byte: 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).

Remark <1> to <9> in Figure 22-7 above correspond to <1> to <9> in the description of "When starting operation" in **22.4.2 (1) When detecting level of supply voltage (V_{DD})**.

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for an operation stabilization time and minimum pulse width (10 μs (MAX.)).
  - <5> Confirm that "input voltage from external input pin (EXLVI) $\geq$ detection voltage (V_{EXLVI} = 1.21 V (TYP.)" when detecting the falling edge of EXLVI, or "input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.
  - <6> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <7> Release the interrupt mask flag of LVI (LVIMK).
  - <8> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <9> Execute the EI instruction (when vector interrupts are used).

  Figure 22-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

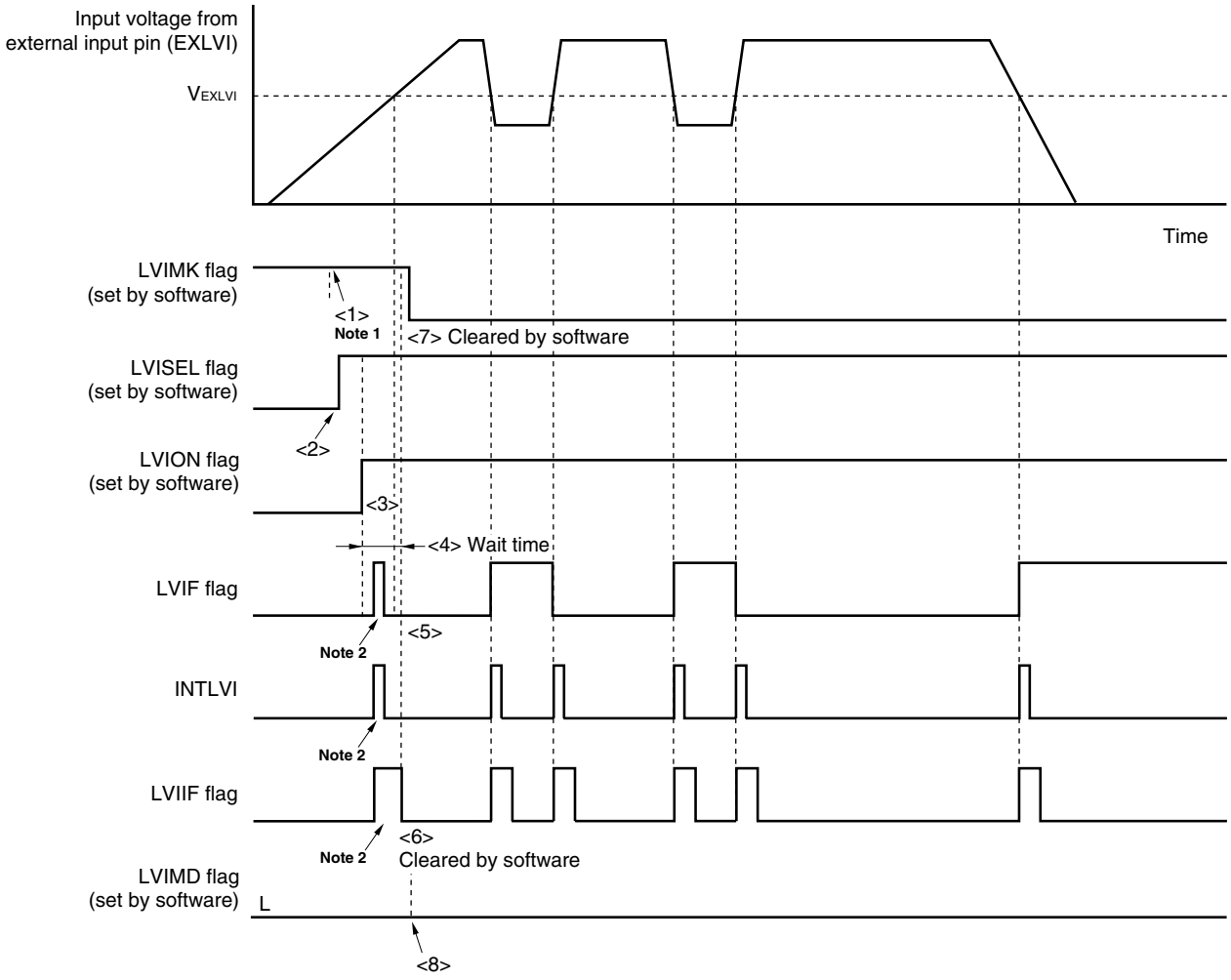
  Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 22-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

Remark <1> to <8> in Figure 22-8 above correspond to <1> to <8> in the description of "When starting operation" in **22.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

22.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

(1) When used as reset

The system may be repeatedly reset and released from the reset status.

In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take (b) of action (2) below.

<Action>

(1) When used as reset

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 22-9**).

(2) When used as interrupt

(a) Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 0 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

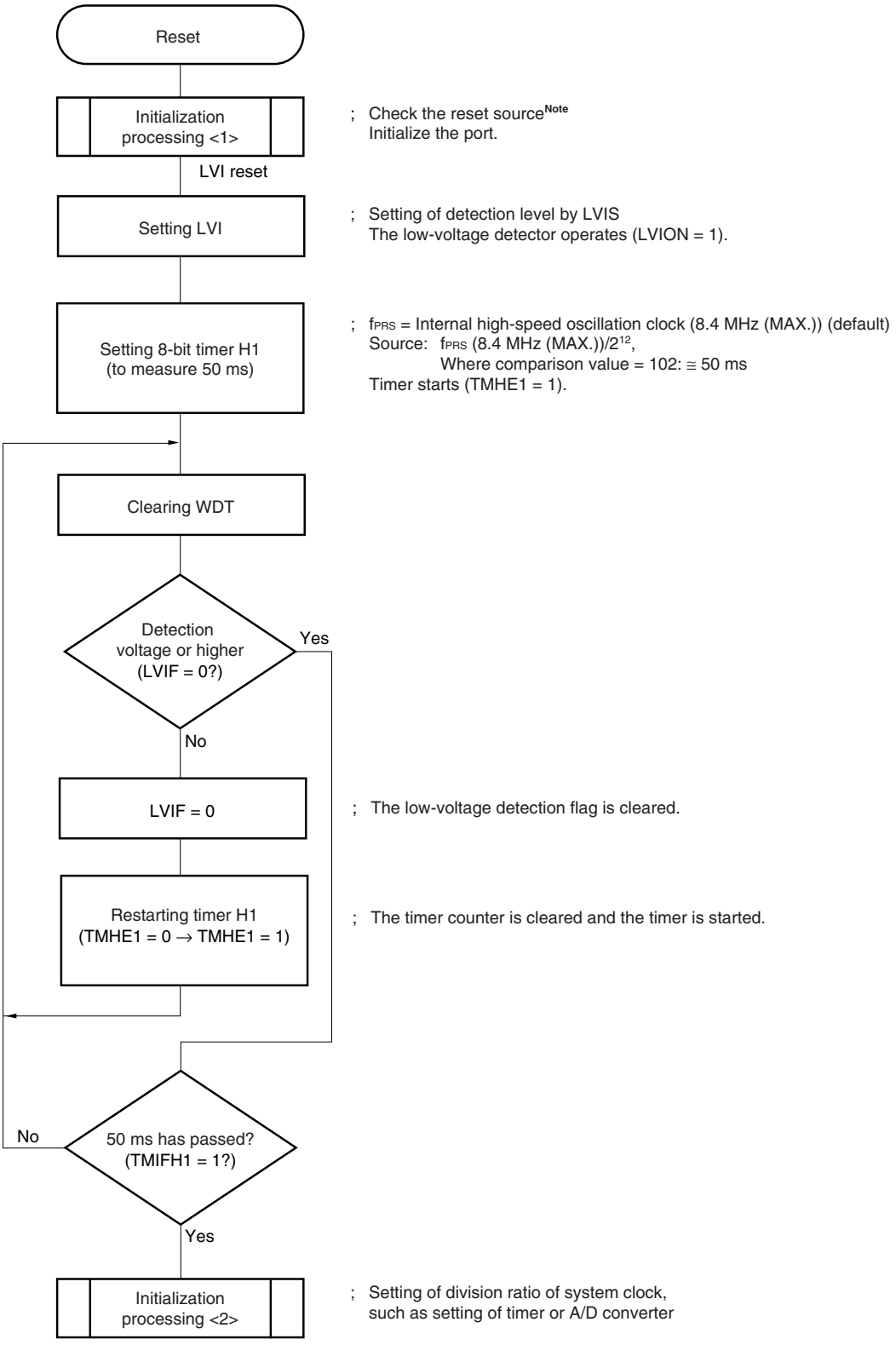
(b) In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, using the LVIF flag, and clear the LVIIF flag to 0.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 22-9. Example of Software Processing After Reset Release (1/2)

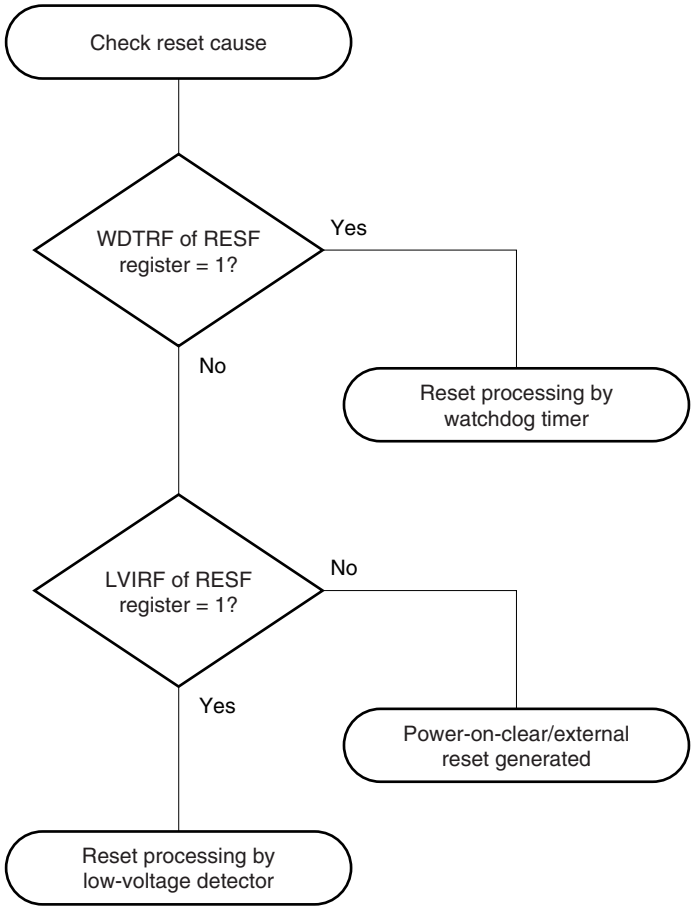
• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

Figure 22-9. Example of Software Processing After Reset Release (2/2)

• Checking reset cause



CHAPTER 23 OPTION BYTE

23.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the 78K0/FE2 is an option byte area. When power is turned on or when the device is restarted from the reset status, the device automatically references the option bytes and sets specified functions. When using the product, be sure to set the following functions by using the option bytes.

When the boot swap operation is used during self-programming, 0080H to 0084H are switched to 1080H to 1084H. Therefore, set values that are the same as those of 0080H to 0084H to 1080H to 1084H in advance.

Caution Be sure to set 00H to 0082H and 0083H (0082H/1082H and 0083H/1083H when the boot swap function is used).

(1) 0080H/1080H

- ○ Internal low-speed oscillator operation
  - Can be stopped by software
  - Cannot be stopped
- ○ Watchdog timer interval time setting
- ○ Watchdog timer counter operation
  - Enabled counter operation
  - Disabled counter operation
- ○ Watchdog timer window open period setting

Caution Set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.

(2) 0081H/1081H

- ○ Selecting POC mode
  - During 2.7 V/1.59 V POC mode operation (LVISTART = 1)
    The device is in the reset state upon power application and until the supply voltage reaches 2.7 V (TYP.). It is released from the reset state when the voltage exceeds 2.7 V (TYP.). After that, POC is not detected at 2.7 V but is detected at 1.59 V (TYP.).
    If the supply voltage rises to 1.8 V after power application at a pace slower than 0.5 V/ms (MIN.), use of the 2.7 V/1.59 V POC mode is recommended.
  - During 1.59 V POC mode operation (LVISTART = 0)
    The device is in the reset state upon power application and until the supply voltage reaches 1.59 V (TYP.). It is released from the reset state when the voltage exceeds 1.59 V (TYP.). After that, POC is detected at 1.59 V (TYP.), in the same manner as on power application.

Caution 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 (at this time, 1.59 V POC mode (default) is set). However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

(3) 0084H/1084H

○ On-chip debug operation control

- Disabling on-chip debug operation
- Enabling on-chip debug operation and erasing data of the flash memory in case authentication of the on-chip debug security ID fails
- Enabling on-chip debug operation and not erasing data of the flash memory even in case authentication of the on-chip debug security ID fails

Caution To use the on-chip debug function, set 02H or 03H to 0084H. Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot operation.

23.2 Format of Option Byte

The format of the option byte is shown below.

Figure 23-1. Format of Option Byte (1/2)

Address: 0080H/1080H[Note]

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC

WINDOW1	WINDOW0	Watchdog timer window open period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter/illegal access detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

LSROSC	Internal low-speed oscillator operation
0	Can be stopped by software (stopped when 1 is written to bit 0 (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 oscillator mode register (RCM).
When 8-bit timer H1 operates with the internal low-speed oscillation clock, the count clock is supplied to 8-bit timer H1 even in the HALT/STOP mode.

4. Be sure to clear bit 7 to 0.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

2. ( ): f_{RL} = 264 kHz (MAX.)

Figure 23-1. Format of Option Byte (2/2)

Address: 0081H/1081H[Notes 1, 2]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	LVISTART

LVISTART	POC mode selection
0	1.59 V POC mode (default)
1	2.7 V/1.59 V POC mode

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 (at this time, 1.59 V POC mode (default) is set). However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

2. To change the setting for the POC mode, set the value to 0081H again after batch erasure (chip erasure) of the flash memory. The setting cannot be changed after the memory of the specified block is erased.

Caution Be sure to clear bits 7 to 1 to "0".

Address: 0082H/1082H, 0083H/1083H[Note]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Note Be sure to set 00H to 0082H and 0083H, as these addresses are reserved areas. Also set 00H to 1082H and 1083H because 0082H and 0083H are switched with 1082H and 1083H when the boot swap operation is used.

Address: 0084H/1084H[Note]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	On-chip debug operation control
0	0	Operation disabled
0	1	Setting prohibited
1	0	Operation enabled. Does not erase data of the flash memory in case authentication of the on-chip debug security ID fails.
1	1	Operation enabled. Erases data of the flash memory in case authentication of the on-chip debug security ID fails.

Note To use the on-chip debug function, set 02H or 03H to 0084H. Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot swap operation.

Remark For the on-chip debug security ID, see **CHAPTER 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: 2^10/fRL,
                           ; Internal low-speed oscillator can be stopped by software.
         DB     00H        ; 1.59 V POC mode
         DB     00H        ; Reserved area
         DB     00H        ; Reserved area
         DB     00H        ; On-chip debug operation disabled
```

Remark Referencing of the option byte is performed during reset processing. For the reset processing timing, see **CHAPTER 19 RESET FUNCTION**.

CHAPTER 24 FLASH MEMORY

The 78K0/FE2 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

The internal memory capacity can be selected using the internal memory size switching register (IMS).
IMS is set by an 8-bit memory manipulation instruction.
Reset signal generation sets IMS to CFH.

Caution Be sure to set each product to the values shown in Table 24-1 after a reset release.

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
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
1	1	0	0	48 KB
1	1	1	1	60 KB
Other than above				Setting prohibited

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

Table 24-1. Internal Memory Size Switching Register Settings

Flash Memory Versions (78K0/FE2)	IMS Setting
μPD78F0887	CCH
μPD78F0888	CFH
μPD78F0889	CCH[Note]
μPD78F0890	CCH[Note]

Note The μPD78F0889 and μPD78F0890 have internal ROMs of 96 KB and 128 KB, respectively. However, the set values of the IMS of these devices is the same as those for the 48 KB product because banks are used. For how to set the banks, see **CHAPTER 4 MEMORY BANK SELECT FUNCTION (μPD78F0889, 78F0890 ONLY)**.

24.2 Internal Expansion RAM Size Switching Register

The internal expansion RAM capacity can be selected using the internal expansion RAM size switching register (IXS).

IXS is set by an 8-bit memory manipulation instruction.

Reset signal generation sets IXS to 0CH.

Caution Be sure to set each product to the values shown in Table 24-2 after a reset release.

Figure 24-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
0	1	0	0	0	2048 bytes
0	0	1	0	0	4096 bytes
0	0	0	0	0	6144 bytes
Other than above					Setting prohibited

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

Table 24-2. Internal Expansion RAM Size Switching Register Settings

Flash Memory Versions (78K0/FE2)	IXS Setting
μPD78F0887	08H
μPD78F0888	
μPD78F0889	04H
μPD78F0890	00H

24.3 Writing with Flash Memory Programmer

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the 78K0/FE2 has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0/FE2 is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 24-3. Wiring Between 78K0/FE2 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			With CSI10		With UART60	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SO10/P12	44	TxD60/P13	43
SO/TxD	Output	Transmit signal	SI10/RxD61/P11	45	RxD60/P14	42
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$/TxD61/P10	46	–	–
CLK	Output	Clock to 78K0/FE2	–Note 1	–	**Note 2**	**Note 2**
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	6	$\overline{\text{RESET}}$	6
FLMD0	Output	Mode signal	FLMD0	9	FLMD0	9
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	15	V_{DD}	15
			EV_{DD}	16	EV_{DD}	16
			AV_{REF}	47	AV_{REF}	47
GND	–	Ground	V_{SS}	13	V_{SS}	13
			EV_{SS}	14	EV_{SS}	14
			AV_{SS}	48	AV_{SS}	48

Notes 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

2. Only the X1 clock (f_X) or external main system clock (f_{EXCLK}) can be used when UART60 is used. When using the clock output of the dedicated flash memory programmer, pin connection varies depending on the type of the dedicated flash memory programmer used.

- PG-FP4, FL-PR4: Connect CLK of the programmer to EXCLK/X2/P122 (pin 10).
- PG-FPL3, FP-LITE3: Connect CLK of the programmer to X1/P121 (pin 11), and connect its inverted signal to X2/EXCLK/P122 (pin 10).

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 24-3. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode

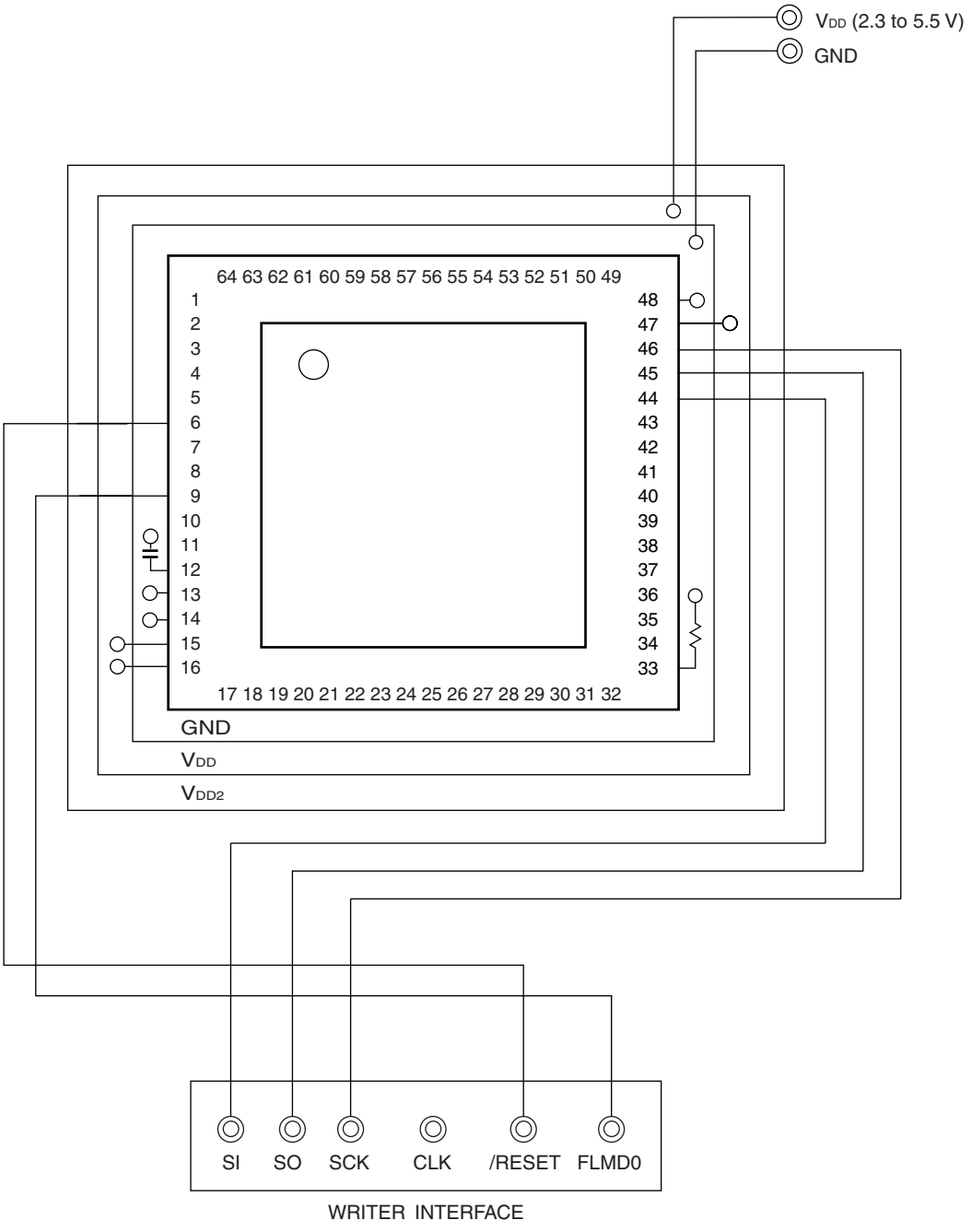
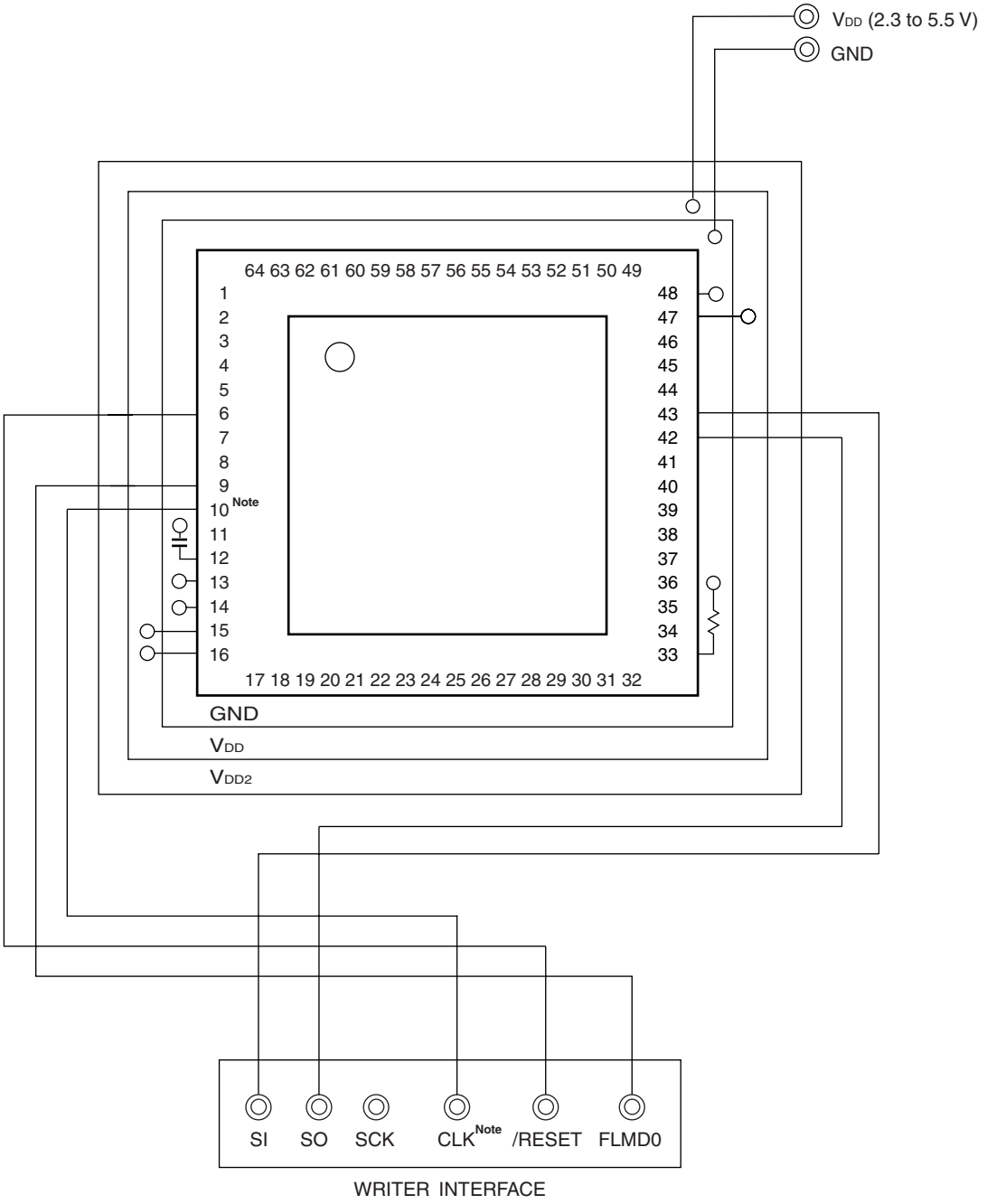


Figure 24-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART60) Mode

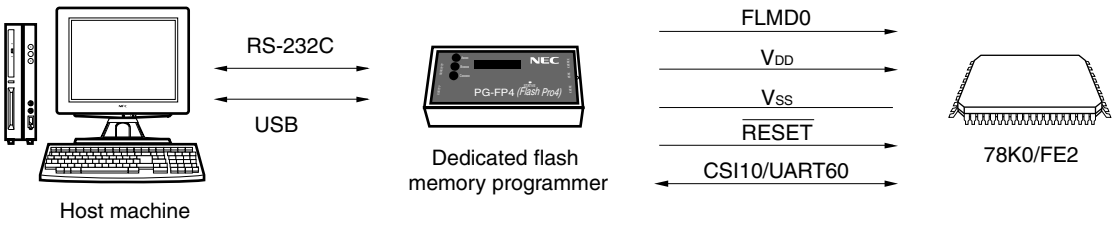


Note The above figure illustrates an example of wiring when using the clock output from the PG-FP4 or FL-PR4. When using the clock output from the PG-FPL3 or FP-LITE3, connect CLK to X1/P121 (pin 11), and connect its inverted signal to X2/EXCLK/P122 (pin 10).

24.4 Programming Environment

The environment required for writing a program to the flash memory of the 78K0/FE2 is illustrated below.

Figure 24-5. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

To interface between the dedicated flash memory programmer and the 78K0/FE2, CSI10 or UART60 is used for manipulation such as writing and erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

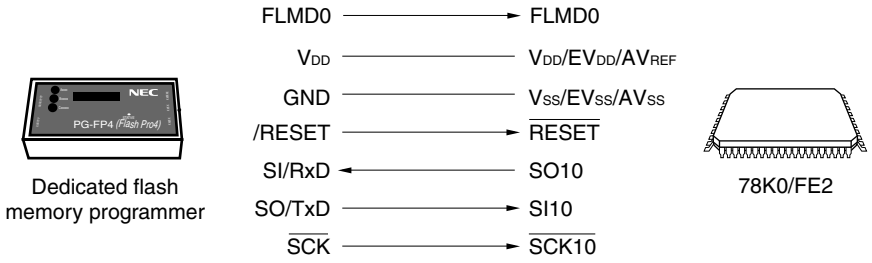
24.5 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0/FE2 is established by serial communication via CSI10 or UART60 of the 78K0/FE2.

(1) CSI10

Transfer rate: 2.4 kHz to 2.5 MHz

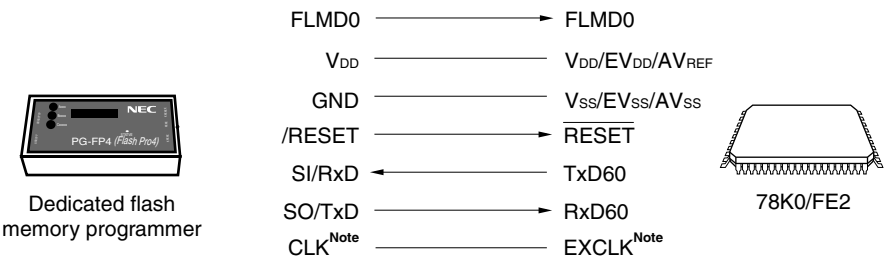
Figure 24-6. Communication with Dedicated Flash Memory Programmer (CSI10)



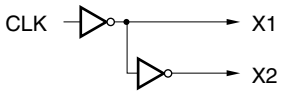
(2) UART60

Transfer rate: 115200 bps

Figure 24-7. Communication with Dedicated Flash Memory Programmer (UART60)



Note The above figure illustrates an example of wiring when using the clock output from the PG-FP4 or FL-PR4. When using the clock output from the PG-FPL3 or FP-LITE3, connect CLK to X1/P121 (pin 11), and connect its inverted signal to X2/EXCLK/P122 (pin 10).



The dedicated flash memory programmer generates the following signals for the 78K0/FE2. For details, refer to the user's manual for the PG-FP4, FL-PR4, PG-FPL3, or FP-LITE3.

Table 24-4. Pin Connection

Dedicated Flash memory programmer			78K0/FE2	Connection	
Signal Name	I/O	Pin Function	Pin Name	CSI10	UART60
FLMD0	Output	Mode signal	FLMD0	◎	◎
VDD	I/O	VDD voltage generation/power monitoring	VDD, EVDD, AVREF	◎	◎
GND	–	Ground	VSS, EVSS, AVSS	◎	◎
CLK	Output	Clock output to 78K0/FE2	**Note 1**	×Note 2	○Note 1
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI/RxD	Input	Receive signal	SO10/TxD60	◎	◎
SO/TxD	Output	Transmit signal	SI10/RxD60	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	◎	×

Notes 1. Only the X1 clock (fX) or external main system clock (fEXCLK) can be used when UART60 is used. When using the clock output of the dedicated flash memory programmer, pin connection varies depending on the type of the dedicated flash memory programmer used.

- PG-FP4, FL-PR4: Connect CLK of the programmer to EXCLK/X2/P122 (pin 10).
- PG-FPL3, FP-LITE3: Connect CLK of the programmer to X1/P121 (pin 11), and connect its inverted signal to X2/EXCLK/P122 (pin 10).

2. Only the internal high-speed oscillation clock (fRH) can be used when CSI10 is used.

Remark ◎: Be sure to connect the pin.
○: The pin does not have to be connected if the signal is generated on the target board.
×: The pin does not have to be connected.

24.6 Connection of Pins on Board

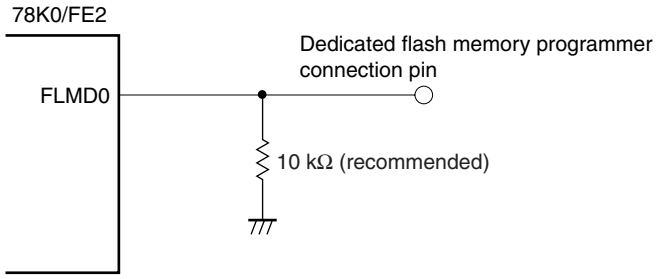
To write the flash memory on-board, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

24.6.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. An FLMD0 pin connection example is shown below.

Figure 24-8. FLMD0 Pin Connection Example



24.6.2 Serial interface pins

The pins used by each serial interface are listed below.

Table 24-5. Pins Used by Each Serial Interface

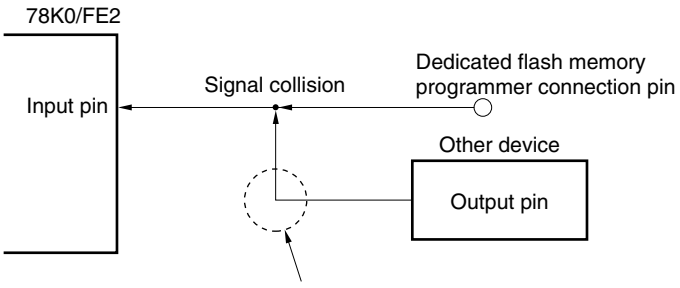
Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{SCK10}$
UART60	TxD60, RxD60

To connect the dedicated flash memory programmer to the pins of a serial interface that is connected to another device on the board, care must be exercised so that signals do not collide or that the other device does not malfunction.

(1) Signal collision

If the dedicated flash memory programmer (output) is connected to a pin (input) of a serial interface connected to another device (output), signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into an output high-impedance state.

Figure 24-9. Signal Collision (Input Pin of Serial Interface)

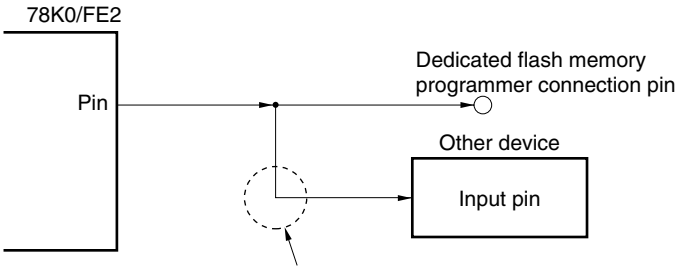


In the flash memory programming mode, the signal output by the device collides with the signal sent from the dedicated flash memory programmer. Therefore, isolate the signal of the other device.

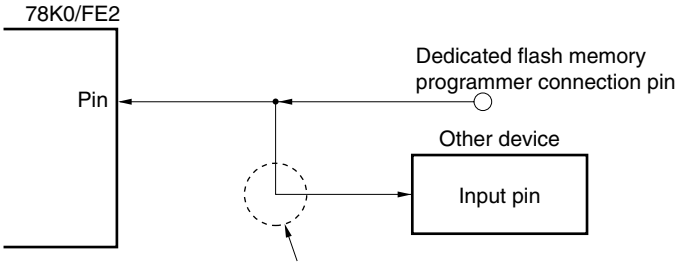
(2) Malfunction of other device

If the dedicated flash memory programmer (output or input) is connected to a pin (input or output) of a serial interface connected to another device (input), a signal may be output to the other device, causing the device to malfunction. To avoid this malfunction, isolate the connection with the other device.

Figure 24-10. Malfunction of Other Device



If the signal output by the 78K0/FE2 in the flash memory programming mode affects the other device, isolate the signal of the other device.



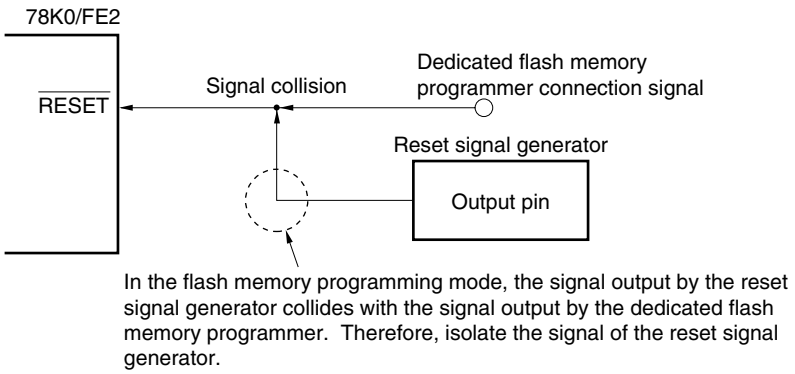
If the signal output by the dedicated flash memory programmer in the flash memory programming mode affects the other device, isolate the signal of the other device.

24.6.3 $\overline{\text{RESET}}$ pin

If the reset signal of the dedicated flash memory programmer is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 24-11. Signal Collision ($\overline{\text{RESET}}$ Pin)



24.6.4 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

24.6.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF: recommended) in the same manner as during normal operation.

24.6.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode when using the on-board clock.

To input the operating clock from the dedicated flash memory programmer, however, connect as follows.

- PG-FP4, FL-PR4: Connect CLK of the programmer to EXCLK/X2/P122.
- PG-FPL3, FP-LITE3: Connect CLK of the programmer and X1/P121, and connect its inverted signal to X2/EXCLK/P122.

Cautions 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

2. Only the X1 clock (f_X) or external main system clock (f_{EXCLK}) can be used when UART60 is used.

3. Connect P31/INTP2/TI002 and P121/X1 as follows when writing the flash memory with a flash memory programmer.

- **P31/INTP2/TI002: Connect to EV_{SS} via a resistor (10 kΩ: recommended).**
- **P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).**

The above connection is not necessary when writing the flash memory by means of self programming.

24.6.7 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

However, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer, even when using the on-board supply voltage.

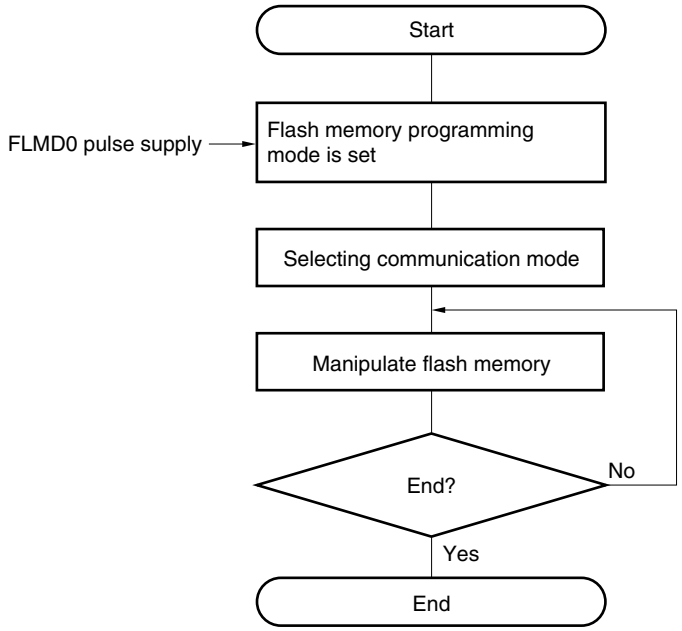
Supply the same other power supplies (EV_{DD}, EV_{SS}, AV_{REF}, and AV_{SS}) as those in the normal operation mode.

24.7 Programming Method

24.7.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 24-12. Flash Memory Manipulation Procedure



24.7.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0/FE2 in the flash memory programming mode. To set the mode, set the FLMD0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 24-13. Flash Memory Programming Mode

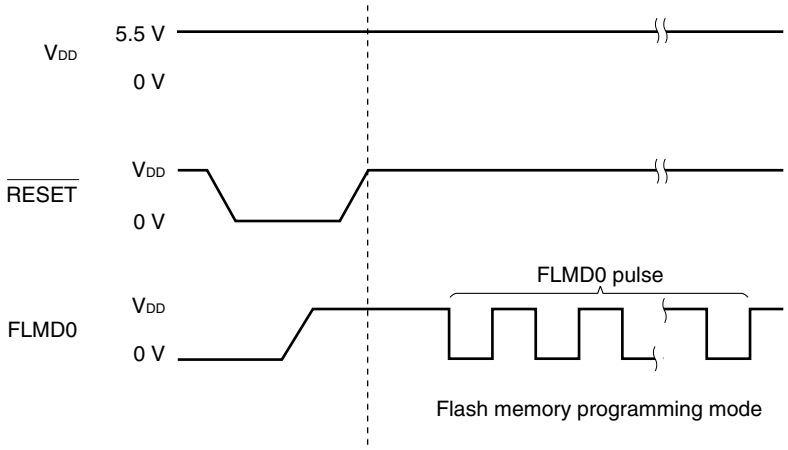


Table 24-6. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

24.7.3 Selecting communication mode

In the 78K0/FE2, a communication mode is selected by inputting pulses (up to 8 pulses) to the FLMD0 pin after the dedicated flash memory programming mode is entered. These FLMD0 pulses are generated by the flash memory programmer.

The following table shows the relationship between the number of pulses and communication modes.

Table 24-7. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used	Peripheral Clock	Number of FLMD0 Pulses
	Port	Speed	Frequency	Multiply Rate			
UART (UART60)	UART-Ext-Osc	115200 bps[Note 2]	2 to 20 MHz[Note 3]	1.0	TxD60, RxD60	f_X	0
	UART-Ext-FP4CK					f_{EXCLK}	3
3-wire serial I/O (CSI10)	CSI-Internal-Osc	2.4 kHz to 2.5 MHz	–		SO10, SI10, $\overline{SCK10}$	f_{RH}	8

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

3. The possible setting range differs depending on the voltage. For details, refer to the chapter of electrical specifications.

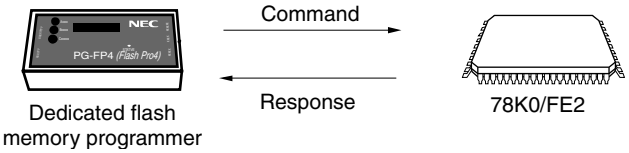
Caution When UART60 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash memory programmer after the FLMD0 pulse has been received.

Remark f_X: X1 clock
f_{EXCLK}: External main system clock
f_{RH}: Internal high-speed oscillation clock

24.7.4 Communication commands

The 78K0/FE2 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0/FE2 are called commands, and the signals sent from the 78K0/FE2 to the dedicated flash memory programmer are called response.

Figure 24-14. Communication Commands



The flash memory control commands of the 78K0/FE2 are listed in the table below. All these commands are issued from the programmer and the 78K0/FE2 perform processing corresponding to the respective commands.

Table 24-8. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Chip Erase	Erases the entire flash memory.
	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Status	Gets the current operating status (status data).
	Silicon Signature	Gets 78K0/Fx2 information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0/Fx2 version and firmware version.
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
Others	Reset	Used to detect synchronization status of communication.
	Oscillating Frequency Set	Specifies an oscillation frequency.

The 78K0/FE2 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0/FE2 are listed below.

Table 24-9. Response Names

Command Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

24.8 Security Settings

The 78K0/FE2 supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the security set command. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0
  Execution of the batch erase (chip erase) command, block erase command, and write command on boot cluster 0 (0000H to 0FFFH) in the flash memory is prohibited by this setting.

  Caution If a security setting that rewrites boot cluster 0 has been applied, boot cluster 0 of that device will not be rewritten.

The batch erase (chip erase), block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by on-board/off-board programming and self programming. Each security setting can be used in combination.

Prohibition of erasing blocks and writing is cleared by executing the batch erase (chip erase) command.

Table 24-10 shows the relationship between the erase and write commands when the 78K0/FE2 security function is enabled.

Table 24-10. 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 24-11 shows how to perform security settings in each programming mode.

Table 24-11. Setting Security in Each Programming Mode

(1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

(2) Self programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set by using information library.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

24.9 Processing Time for Each Command When PG-FP4 Is Used (Reference)

The following table shows the processing time for each command (reference) when the PG-FP4 is used as a dedicated flash memory programmer.

<R> **Table 24-12. Processing Time for Each Command When PG-FP4 Is Used (Reference)**

- **μPD78F0890 (internal ROM capacity: 128 KB)**

Command of PG-FP4	Port: CSI-Internal-OSC (Internal high-speed oscillation clock (f_{RH})), Speed: 2.5 MHz	Port: UART-Ext-FP4CK (External main system clock (f_{EXCLK})), Speed: 115,200 bps	
		Frequency: 2.0 MHz	Frequency: 20 MHz
Signature	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)
Blankcheck	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Erase	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)
Program	9.5 s (TYP.)	18 s (TYP.)	18 s (TYP.)
Verify	4.5 s (TYP.)	13.5 s (TYP.)	13.5 s (TYP.)
E.P.V	11 s (TYP.)	19.5 s (TYP.)	19.5 s (TYP.)
Checksum	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Security	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)

24.10 Flash Memory Programming by Self-Programming

The 78K0/FE2 supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the 78K0/FE2 self-programming library, it can be used to upgrade the program in the field.

If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. To execute interrupt servicing, restore the normal operation mode after self-programming has been stopped, and execute the EI instruction. After the self-programming mode is later restored, self-programming can be resumed.

Remark For details of the self-programming function and the 78K0/FE2 self-programming library, refer to a separate document to be published (document name: 78K0/Fx2 Application Note, release schedule: Pending).

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. Input a high level to the FLMD0 pin during self-programming.

3. Be sure to execute the DI instruction before starting self-programming.
The self-programming function checks the interrupt request flags (IF0L, IF0H, IF1L, and IF1H). If an interrupt request is generated, self-programming is stopped.

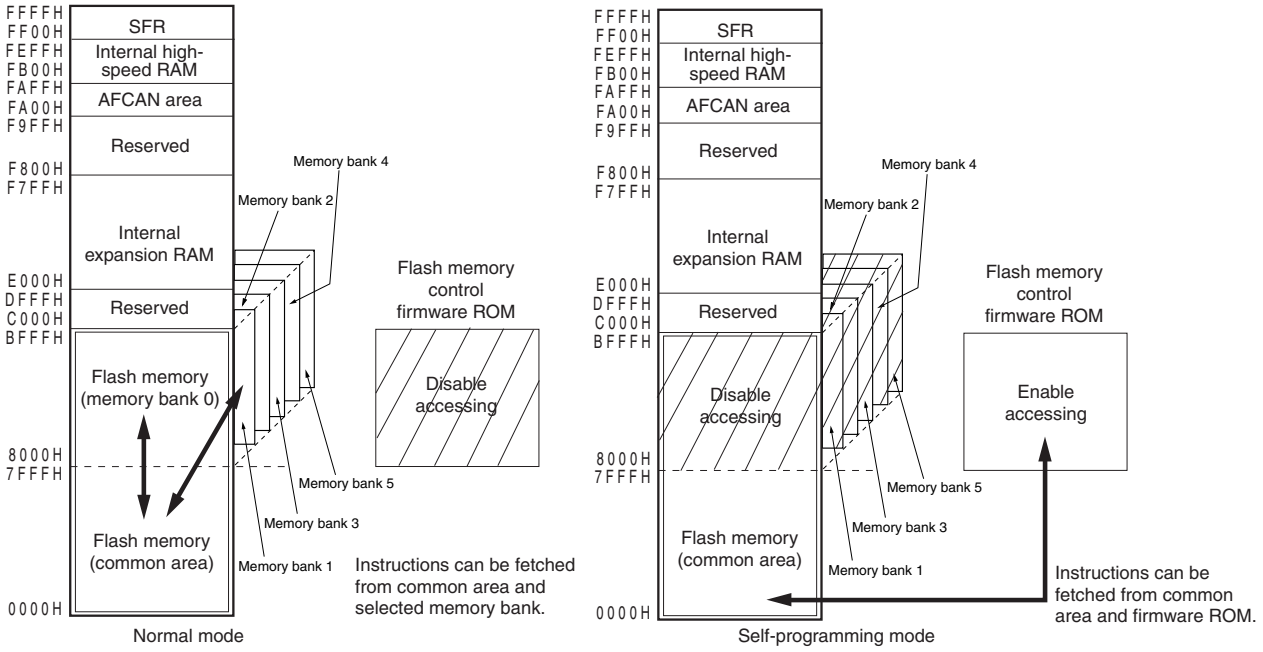
4. Self-programming is also stopped by an interrupt request that is not masked even in the DI status. To prevent this, mask the interrupt by using the interrupt mask flag registers (MK0L, MK0H, MK1L, and MK1H).

5. Self-programming is executed with the internal high-speed oscillation clock. If the CPU operates with the X1 clock or external main system clock, the oscillation stabilization wait time of the internal high-speed oscillation clock elapses during self-programming.

(Cautions 6 and 7 are listed on the next page.)

Cautions 6. Allocate the entry program for self-programming in the common area of 0000H to 7FFFH.

Figure 24-15. Operation Mode and Memory Map for Self-Programming (μPD78F0890)



7. If the flash memory size is 96 KB or 128 KB, specify a flash real address, instead of a CPU address, as a flash write/erase address.

Table 24-13. Correspondence Among Bank Numbers, CPU Addresses, and Flash Real Addresses

(a) μPD78F0889

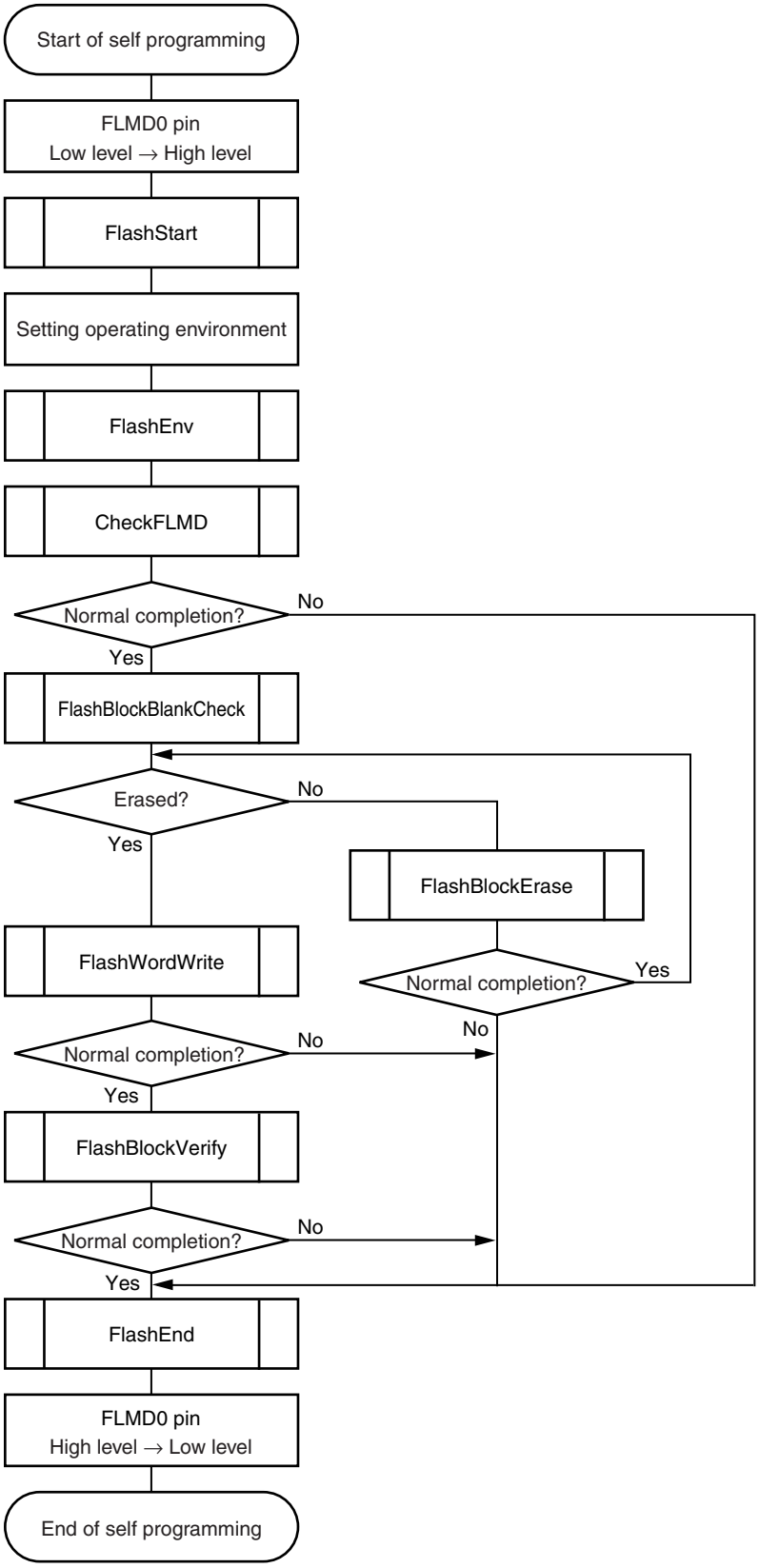
Bank No.	CPU Address	Real Address of Flash Memory
–	0000H to 7FFFH (common area)	00000H to 07FFFH
0	8000H to BFFFH	08000H to 0BFFFH
1		0C000H to 0FFFFH
2		10000H to 13FFFH
3		14000H to 17FFFH
4 or more	Setting prohibited	

(b) μPD78F0890

Bank No.	CPU Address	Real Address of Flash Memory
–	0000H to 7FFFH (common area)	00000H to 07FFFH
0	8000H to BFFFH	08000H to 0BFFFH
1		0C000H to 0FFFFH
2		10000H to 13FFFH
3		14000H to 17FFFH
4		18000H to 1BFFFH
5		1C000H to 1FFFFH
6 or more	Setting prohibited	

The following figure illustrates a flow of rewriting the flash memory by using a self programming sample library.

Figure 24-16. Flow of Self Programming (Rewriting Flash Memory)



The following table shows the processing time and interrupt response time for the self programming sample library.

Table 24-14. Processing Time and Interrupt Response Time for Self Programming Sample Library (1/4)

(1) When internal high-speed oscillation clock is used and entry RAM is located outside short direct addressing range

Library Name	Processing Time (μs)				Interrupt Response Time (μs)	
	Normal Model of C Compiler		Static Model of C Compiler/Assembler			
	Min.	Max.	Min.	Max.	Min.	Max.
Self programming start library	4.25				–	–
Initialize library	977.75				–	–
Mode check library	753.875		753.125		–	–
Block blank check library	12770.875		12765.875		391.25	1300.5
Block erase library	36909.5	356318	36904.5	356296.25	389.25	1393.5
Word write library	1214 (1214.375)	2409 (2409.375)	1207 (1207.375)	2402 (2402.375)	394.75	1289.5
Program verify library	25618.875		25613.875		390.25	1324.5
Self programming end library	4.25				–	–
Get information library (option value: 03H)	871.25 (871.375)		866 (866.125)		–	–
Get information library (option value: 04H)	863.375 (863.5)		858.125 (858.25)		–	–
Get information library (option value: 05H)	1024.75 (1043.625)		1037.5 (1038.375)		–	–
Set information library	105524.75	790809.375	105523.75	790808.375	387	852.5
EEPROM write library	1496.5 (1496.875)	2691.5 (2691.875)	1489.5 (1489.875)	2684.5 (2684.875)	399.75	1395.5

Remark The value in the parentheses indicates the value when a write start address structure is located at a place other than the internal high-speed RAM.

Table 24-14. Processing Time and Interrupt Response Time for Self Programming Sample Library (2/4)

(2) When internal high-speed oscillation clock is used and entry RAM is located in short direct addressing range (FE20H)

Library Name	Processing Time (μs)				Interrupt Response Time (μs)	
	Normal Model of C Compiler		Static Model of C Compiler/Assembler			
	Min.	Max.	Min.	Max.	Min.	Max.
Self programming start library	4.25				–	–
Initialize library	443.5				–	–
Mode check library	219.625		218.875		–	–
Block blank check library	12236.625		12231.625		81.25	727.5
Block erase library	36363.25	355771.75	36358.25	355750	79.25	820.5
Word write library	679.75 (680.125)	1874.75 (1875.125)	672.75 (673.125)	1867.75 (1868.125)	84.75	716.5
Program verify library	25072.625		25067.625		80.25	751.5
Self programming end library	4.25				–	–
Get information library (option value: 03H)	337 (337.125)		331.75 (331.875)		–	–
Get information library (option value: 04H)	329.125 (239.25)		323.875 (324)		–	–
Get information library (option value: 05H)	502.25 (503.125)		497 (497.875)		–	–
Set information library	104978.5	541143.125	104977.5	541142.125	77	279.5
EEPROM write library	962.25 (962.625)	2157.25 (2157.625)	955.25 (955.625)	2150.25 (2150.625)	89.75	822.5

Remark The value in the parentheses indicates the value when a write start address structure is located at a place other than the internal high-speed RAM.

Table 24-14. Processing Time and Interrupt Response Time for Self Programming Sample Library (3/4)

(3) When high-speed system clock (X1 oscillation or external clock input) is used and entry RAM is located outside short direct addressing range

Library Name	Processing Time (μs)				Interrupt Response Time (μs)	
	Normal Model of C Compiler		Static Model of C Compiler/Assembler			
	Min.	Max.	Min.	Max.	Min.	Max.
Self programming start library	$34/f_{XH}$				–	–
Initialize library	$49/f_{XH} + 485.8125$				–	–
Mode check library	$35/f_{XH} + 374.75$		$29/f_{XH} + 374.75$		–	–
Block blank check library	$174/f_{XH} + 6382.0625$		$134/f_{XH} + 6382.0625$		$18/f_{XH} + 192$	$28/f_{XH} + 698$
Block erase library	$174/f_{XH} + 31093.875$	$174/f_{XH} + 298948.125$	$134/f_{XH} + 31093.875$	$134/f_{XH} + 298948.125$	$18/f_{XH} + 186$	$28/f_{XH} + 745$
Word write library	$318\ (321)/f_{XH} + 644.125$	$318\ (321)/f_{XH} + 1491.625$	$262\ (265)/f_{XH} + 644.125$	$262\ (265)/f_{XH} + 1491.625$	$22/f_{XH} + 189$	$28/f_{XH} + 693$
Program verify library	$174/f_{XH} + 13448.5625$		$134/f_{XH} + 13448.5625$		$18/f_{XH} + 192$	$28/f_{XH} + 709$
Self programming end library	$34/f_{XH}$				–	–
Get information library (option value: 03H)	$171\ (172)/f_{XH} + 432.4375$		$129\ (130)/f_{XH} + 432.4375$		–	–
Get information library (option value: 04H)	$181\ (182)/f_{XH} + 427.875$		$139\ (140)/f_{XH} + 427.875$		–	–
Get information library (option value: 05H)	$404\ (411)/f_{XH} + 496.125$		$362\ (369)/f_{XH} + 496.125$		–	–
Set information library	$75/f_{XH} + 79157.6875$	$75/f_{XH} + 652400$	$67/f_{XH} + 79157.6875$	$67/f_{XH} + 652400$	$16/f_{XH} + 190$	$28/f_{XH} + 454$
EEPROM write library	$318\ (321)/f_{XH} + 799.875$	$318\ (321)/f_{XH} + 1647.375$	$262\ (265)/f_{XH} + 799.875$	$262\ (265)/f_{XH} + 1647.375$	$22/f_{XH} + 191$	$28/f_{XH} + 783$

Remarks 1. The value in the parentheses indicates the value when a write start address structure is located at a place other than the internal high-speed RAM.

2. f_{XH}: High-speed system clock frequency

Table 24-14. Processing Time and Interrupt Response Time for Self Programming Sample Library (4/4)

(4) When high-speed system clock (X1 oscillation or external clock input) is used and entry RAM is located in short direct addressing range (FE20H)

Library Name	Processing Time (μs)				Interrupt Response Time (μs)	
	Normal Model of C Compiler		Static Model of C Compiler/Assembler			
	Min.	Max.	Min.	Max.	Min.	Max.
Self programming start library	34/f_{XH}				–	–
Initialize library	49/f_{XH} + 224.6875				–	–
Mode check library	35/f_{XH} + 113.625		29/f_{XH} + 113.625		–	–
Block blank check library	174/f_{XH} + 6120.9375		134/f_{XH} + 6120.9375		18/f_{XH} + 55	28/f_{XH} + 462
Block erase library	174/f_{XH} + 30820.75	174/f_{XH} + 298675	134/f_{XH} + 30820.75	134/f_{XH} + 298675	18/f_{XH} + 49	28/f_{XH} + 509
Word write library	318 (321)/f_{XH} + 383	318 (321)/f_{XH} + 1230.5	262 (265)/f_{XH} + 383	262 (265)/f_{XH} + 1230.5	22/f_{XH} + 52	28/f_{XH} + 457
Program verify library	174/f_{XH} + 13175.4375		134/f_{XH} + 13175.4375		18/f_{XH} + 55	28/f_{XH} + 473
Self programming end library	34/f_{XH}				–	–
Get information library (option value: 03H)	171 (172)/f_{XH} + 171.3125		129 (130)/f_{XH} + 171.3125		–	–
Get information library (option value: 04H)	181 (182)/f_{XH} + 166.75		139 (140)/f_{XH} + 166.75		–	–
Get information library (option value: 05H)	404 (411)/f_{XH} + 231.875		362 (369)/f_{XH} + 231.875		–	–
Set information library	75/f_{XH} + 78884.5625	75/f_{XH} + 527566.875	67/f_{XH} + 78884.5625	67/f_{XH} + 527566.875	16/f_{XH} +53	28/f_{XH} +218
EEPROM write library	318 (321)/f_{XH} + 538.75	318 (321)/f_{XH} + 1386.25	262 (265)/f_{XH} + 538.75	262 (265)/f_{XH} + 1386.25	22/f_{XH} +54	28/f_{XH} +547

Remarks 1. The value in the parentheses indicates the value when a write start address structure is located at a place other than the internal high-speed RAM.

2. f_{XH}: High-speed system clock frequency

24.10.1 Registers used for self-programming function

The following three registers are used for the self-programming function.

- Flash-programming mode control register (FLPMC)
- Flash protect command register (PFCMD)
- Flash status register (PFS)

(1) Flash-programming mode control register (FLPMC)

This register is used to enable or disable writing or erasing of the flash memory and to set the operation mode during self-programming.

The FLPMC can be written only in a specific sequence (see **24.10.1 (2) Flash protect command register (PFCMD)**) so that the application system does not stop inadvertently due to malfunction caused by noise or program hang-up.

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

Reset signal generation sets this register to 0xH[Note].

Note Differs depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

Figure 24-17. Format of Flash-Programming Mode Control Register (FLPMC)

Address: FFC4H After reset: 0×H[Note 1] R/W[Note 2]

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	0	0	FWEDIS	FWEPR	FLSPM1	FLSPM0

FWEDIS	Control of flash memory writing/erasing
0	Writing/erasing enabled[Note 3]
1	Writing/erasing disabled

FWEPR	Status of FLMD0 pin
0	Low level
1	High level[Note 3]

FLSPM1[Note 4]	FLSPM0[Note 4]	Selection of operation mode during self-programming
0	0	Normal mode Access (fetch of a command, lead of data) is possible to all the address domains of a flash memory.
0	1	Self-programming mode Execution"CALL #8100 H" of firmware is possible. Access (lead of an instruction fetch and data) is possible to a flash memory .
1	1	Setting prohibited
1	0	

Notes 1. Differs depending on the operation mode.
- User mode: 08H
- On-board mode: 0CH

2. Bit 2 (FWEPR) is read-only.

3. For actual writing/erasing, the FLMD0 pin must be high (FWEPR = 1), as well as FWEDIS = 0.

FWEDIS	FWEPR	Enable or disable of flash memory writing/erasing
0	1	Writing/erasing enabled
Other than above		Writing/erasing disabled

4. The user ROM (flash memory) or firmware ROM can be selected by FLSPM1 and FLSPM0, and the operation mode set on the application system by the mode pin or the self-programming mode can be selected.

Cautions 1. Be sure to keep FWEDIS at 0 until writing or erasing of the flash memory is completed.

2. Make sure that FWEDIS = 1 in the normal mode.

3. Manipulate FLSPM1 and FLSPM0 after execution branches to the internal RAM. The address of the flash memory is specified by an address signal from the CPU when FLSPM1 = 0 or the set value of the firmware written when FLSPM1 = 1. In the on-board mode, the specifications of FLSPM1 and FLSPM0 are ignored.

(2) Flash protect command register (PFCMD)

If the application system stops inadvertently due to malfunction caused by noise or program hang-up, an operation to write the flash programming mode control register (FLPMC) may have a serious effect on the system. PFCMD is used to protect FLPMC from being written, so that the application system does not stop inadvertently. Writing FLPMC is enabled only when a write operation is performed in the following specific sequence.

<1> Write a specific value to PFCMD (PFCMD = A5H)
<2> Write the value to be set to FLPMC (writing in this step is invalid)
<3> Write the inverted value of the value to be set to FLPMC
<4> Write the value to be set to FLPMC (writing in this step is valid)

This rewrites the value of the register, so that the register cannot be written illegally.
Occurrence of an illegal store operation can be checked by bit 0 (FPRERR) of the flash status register (PFS).
A5H must be written to PFCMD each time the value of FLPMC is changed.
PFCMD can be set by an 8-bit memory manipulation instruction.
Reset signal generation makes this register undefined.

Figure 24-18. Format of Flash Protect Command Register (PFCMD)

Address: FFC0H After reset: Undefined W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) Flash status register (PFS)

If data is not written to the flash programming mode control register (FLPMC), which is protected, in the correct sequence (writing the flash protect command register (PFCMD)), FLPMC is not written and a protection error occurs. If this happens, bit 0 of PFS (FPRERR) is set to 1.
This bit is a cumulative flag. After checking FPRERR, clear it by writing 0 to it.
PFS can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 24-19. Format of Flash Status Register (PFS)

Address: FFC2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR

The operating conditions of the FPRERR flag are as follows.

<Setting conditions>

- If PFCMD is written when the store instruction operation recently performed on a peripheral register is not to write a specific value (A5H) to PFCMD
- If the first store instruction operation after <1> is on a peripheral register other than FLPMC
- If the first store instruction operation after <2> is on a peripheral register other than FLPMC
- If a value other than the inverted value of the value to be set to FLPMC is written by the first store instruction after <2>
- If the first store instruction operation after <3> is on a peripheral register other than FLPMC
- If a value other than the value to be set to FLPMC (value written in <2>) is written by the first store instruction after <3>

Remark The numbers in angle brackets above correspond to the those in **(2) Flash protect command register (PFCMD)**.

<Reset conditions>

- If 0 is written to the FPRERR flag
- If reset signal is generated

<Example of description in specific sequence>

To write 05H to FLPMC

```
MOV     PFCMD, #0A5H      ; Writes A5H to PFCMD.
MOV     FLPMC, #05H       ; Writes 05H to FLPMC.
MOV     FLPMC, #0FAH      ; Writes 0FAH (inverted value of 05H) to FLPMC.
MOV     FLPMC, #05H       ; Writes 05H to FLPMC.
```

24.11 Boot Swap Function

If rewriting the boot area has failed during self-programming due to a power failure or some other cause, the data in the boot area may be lost and the program may not be restarted by resetting.

The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the 78K0/FE2, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0.

As a result, even if a power failure occurs while the boot programming area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

If the program has been correctly written to boot cluster 0, restore the original boot area by using the set information function of the firmware of the 78K0/FE2.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Boot cluster 0 (0000H to 0FFFH): Original boot program area

Boot cluster 1 (1000H to 1FFFH): Area subject to boot swap function

Figure 24-20. Boot Swap Function

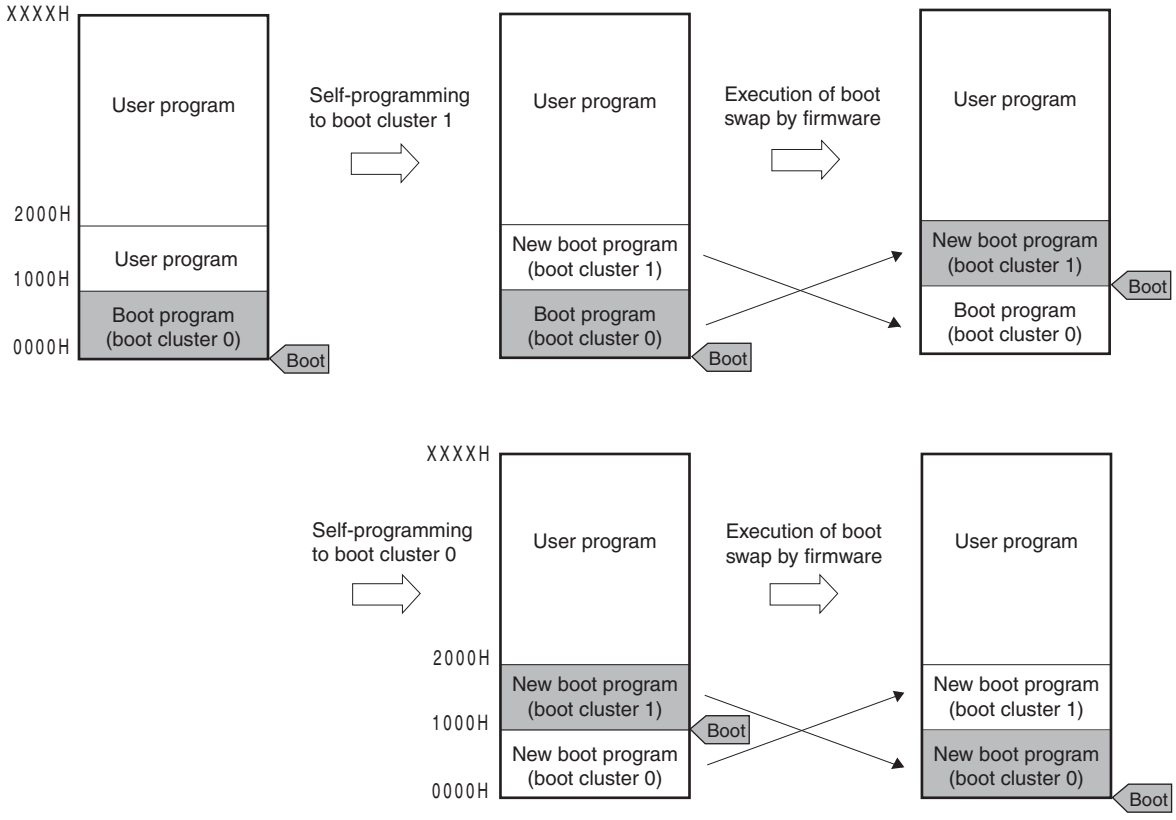
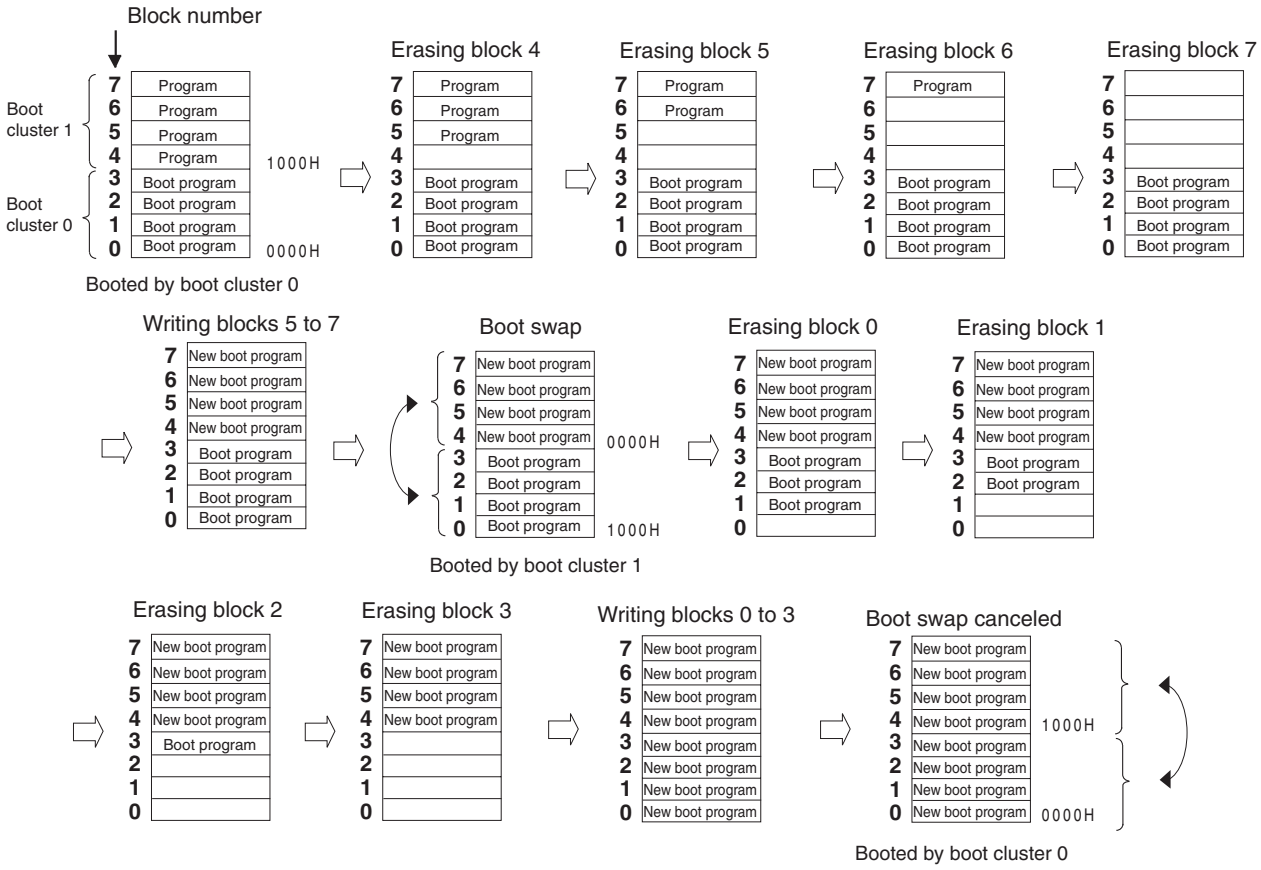


Figure 24-21. Example of Executing Boot Swapping



CHAPTER 25 ON-CHIP DEBUG FUNCTION

25.1 Outline of Functions

The 78K0/FE2 uses the V_{DD}, FLMD0, $\overline{RESET}$, X1 (or P31), X2 (or P32), and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-78K0MINI). Whether X1 and P31, or X2 and P32 are used can be selected.

Caution Do not use this product for mass production after the on-chip debug function has been used because its reliability cannot be guaranteed, due to issues with respect to the number of times the flash memory can be rewritten. NEC Electronics does not accept complaints concerning when use this product for mass production after the on-chip debug function has been used.

25.2 Connection with MINICUBE

In order to connect QB-78K0MINI, it is necessary to mount the connector for emulator connection, and the circuit for connection on a target system.

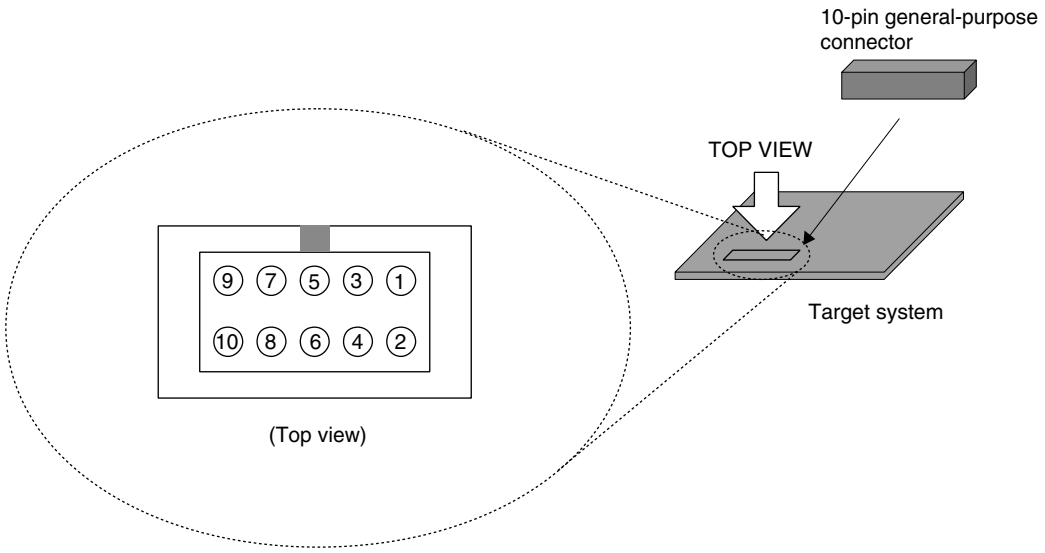
The connector for OCD (a two-row 2.54 pitch type connector, with reverse-insertion blocker) is described below.

- Recommended connectors: (straight) HIF3FC-10PA-2.54DSA (manufactured by Hirose Electric Co., Ltd.)
  (right angle) HIF3FC-10PA-2.54DS (manufactured by Hirose Electric Co., Ltd.))

Pin No.	Name	IN/OUT	Remark
1	RESET_IN	IN	Target reset input signal
2	$\overline{\text{RESET\_OUT}}$	OUT	Reset signal output to target device
3	FLMD0	OUT	Output signal[Note] used to control on-chip debugging functions
4	V_{DD}_IN	IN	This signal is used to generate an interface output signal when the target system's V_{DD} is detected.
5	X2	IN/OUT	Bidirectional signal used for data communications
6	GND	–	Connected to GND.
7	X1	OUT	Output signal used for clock supply
8	GND	–	Connected to GND.
9	RESERVED	–	Open
10	RESERVED	–	Open

Note FLMD0 is at high level during on-chip debugging.

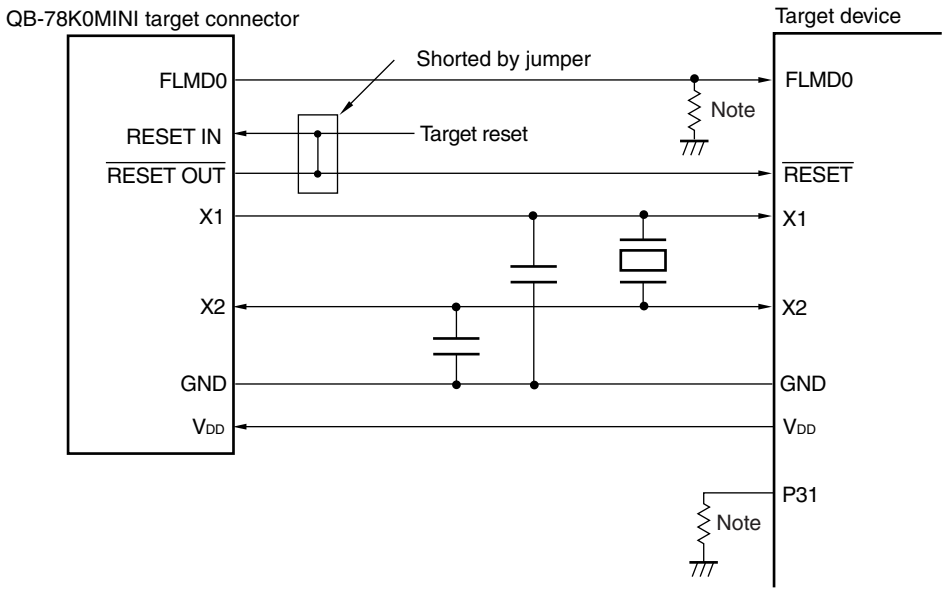
Figure 25-1. Connector Pin Layout



25.3 Connection Circuit Examples

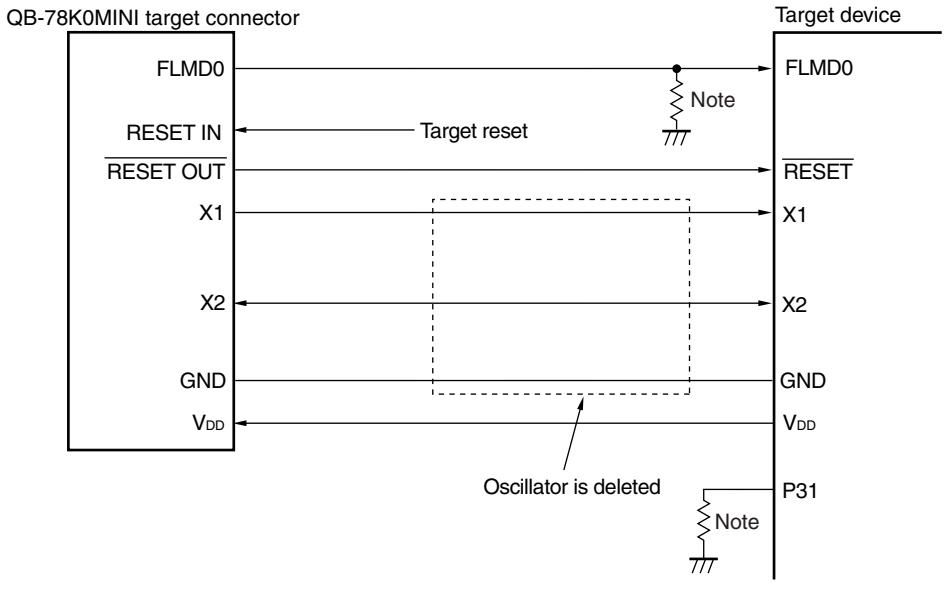
The following are examples of circuits required when connecting the QB-78K0MINI to the target system.

Figure 25-2. Connection Circuit Example (When QB-78K0MINI Is Not Used)



Note Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Figure 25-3. Connection Circuit Example (When Using QB-78K0MINI: X1 and X2 Are Used)

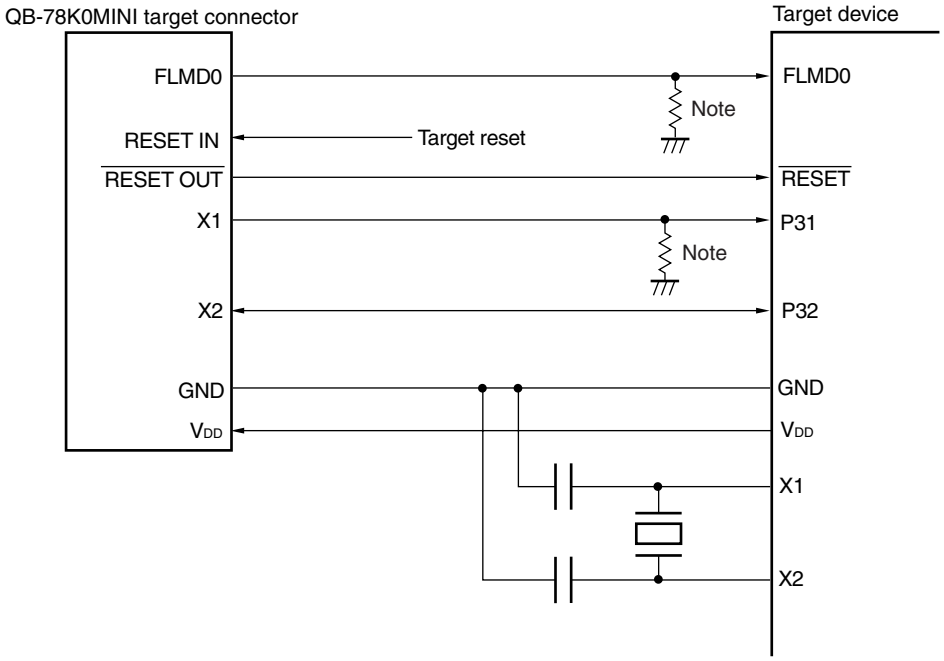


Note Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Cautions 1. Input the clock from the X1 pin during on-chip debugging.

2. Control the X1 and X2 pins by externally pulling down the P31 pin or by using an external circuit using the P130 pin (that outputs a low level when the device is reset).

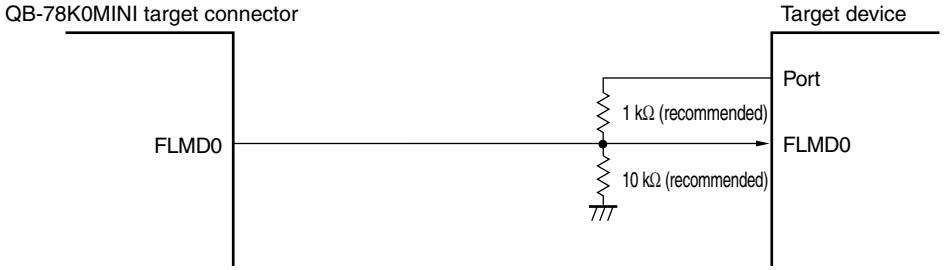
Figure 25-4. Connection Circuit Example (When Using QB-78K0MINI: Ports 31 and 32 Are Used)



Note Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Connect the FLMD0 pin as follows when performing self programming by means of on-chip debugging.

Figure 25-5. Connection of FLMD0 Pin for Self Programming by Means of On-Chip Debugging



25.4 On-Chip Debug Security ID

The 78K0/FE2 has an on-chip debug operation control flag in the flash memory at 0084H (see **CHAPTER 23 OPTION BYTE**) and an on-chip debug security ID setting area at 0085H to 008EH.

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-78K0MINI User's Manual (U17029E)**.

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	

25.5 Restrictions and Cautions on On-Chip Debug Function

When setting to on-chip debugging mode via the normal port, without using pins X1 and X2, two of the user ports will be unavailable for use.

A high-level signal is always output from to the FLMD0 pin during emulation when self-writing. Be sure to connect a pull-down resistor to the FLMD0 pin, and manipulate this pin based on high/high impedance levels, rather than on high/low levels, when using ports for manipulation.

In order to realize on-chip debug function, use the following user resource.

(a) Flash memory area

- Addresses 0x02 and 0x03
- Addresses 0x7E and 0x7F (when using a software break)
- Address 0x84
- Addresses 0x85 to 0x8E
- Addresses 0x8F to 0x18F: Standard value of program
  (+256 bytes when using pseudo real-time RAM monitor function)
  (when using a device with 10 or more SFRs the can be accessed in 16-bit units: +n (the number of exceeding registers x 6 bytes))

(b) Internal extended RAM area

- Addresses 0xF7F0 to 0xF7FF
  (when using pseudo real-time RAM monitor function)

(c) Internal high-speed RAM area

- 7 bytes as stack area: Standard value of stack
  (+2 bytes when using software breaks)
  (+7 bytes when using pseudo real-time RAM monitor function)

For details, refer to the QB-78K0MINI User's Manual (U17029E).

CHAPTER 26 INSTRUCTION SET

This chapter lists each instruction set of 78K0/FE2 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. Upper case 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, see **Table 3-8. 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
$\wedge$: Logical product (AND)
$\vee$: Logical sum (OR)
$\veebar$: Exclusive logical sum (exclusive OR)
$\overline{\ \ }$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

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) + C	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ [HL]	×		
		A, [HL + byte]	2	8	9	A ← A ∧ [HL + byte]	×		
		A, [HL + B]	2	8	9	A ← A ∧ [HL + B]	×		
		A, [HL + C]	2	8	9	A ← A ∧ [HL + C]	×		

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×
		A, r Note 3	2	4	–	A ← A ∨ r	×
		r, A	2	4	–	r ← r ∨ A	×
		A, saddr	2	4	5	A ← A ∨ (saddr)	×
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×
		A, [HL]	1	4	5	A ← A ∨ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ∨ (HL + C)	×
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×
		A, r Note 3	2	4	–	A ← A ⊻ r	×
		r, A	2	4	–	r ← r ⊻ A	×
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ⊻ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ⊻ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ⊻ (HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9	A – (addr16)	× × ×
		A, [HL]	1	4	5	A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit	×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit	×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit	×
		CY, PSW. bit	3	–	7	CY ← CY ⊻ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit	×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1	
		sfr.bit	3	–	8	sfr.bit ← 1	
		A.bit	2	4	–	A.bit ← 1	
		PSW.bit	2	–	6	PSW.bit ← 1	× × ×
		[HL].bit	2	6	8	(HL).bit ← 1	
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0	
		sfr.bit	3	–	8	sfr.bit ← 0	
		A.bit	2	4	–	A.bit ← 0	
		PSW.bit	2	–	6	PSW.bit ← 0	× × ×
		[HL].bit	2	6	8	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15 – 11 ← 00001, PC10 – 0 ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (00000000, addr5 + 1), PCL ← (00000000, 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)

27.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			−0.5 to +6.5	V
	EV_{DD}			−0.5 to +6.5	V
	V_{SS}			−0.5 to +0.3	V
	EV_{SS}			−0.5 to +0.3	V
	AV_{REF}			−0.5 to V_{DD} +0.3[Note]	V
	AV_{SS}			−0.5 to +0.3	V
REGC pin Input voltage	V_{REGC}			−0.5 to +3.6 and ≤ V_{DD}	V
Input voltage	V_{I1}	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P80 to P87, P90 to P93, P120, P131, P132, X1, X2, XT1, XT2, $\overline{\text{RESET}}$, FLMD0		−0.3 to V_{DD} +0.3[Note]	V
	V_{I2}	P60 to P63	N-ch open drain	−0.3 to +6.5	V
Output voltage	V_O			−0.3 to V_{DD} +0.3[Note]	V
Analog input voltage	V_{AN}	ANI0 to ANI11		−0.3 to AV_{REF} +0.3[Note] and −0.3 to V_{DD} +0.3[Note]	V
Output current, high	I_{OH}	Per pin	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130, P131, P132	−10	mA
		Total of all pins −80 mA	P05, P06, P10 to P17, P30 to P33, P50 to P53, P70 to P76, P130	−55	mA
			P00, P01, P40 to P43, P120, P131, P132	−25	mA
	I_{OH2}	Per pin	P80 to P87, P90 to P93	−0.5	mA
		Total of all pins		−2	
	I_{OH3}	Per pin	P121 to P124	−1	mA
		Total of all pins		−4	

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions		Ratings	Unit
Output current, low	I_{OL}	Per pin	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P130, P131, P132	30	mA
		Total of all pins 200 mA	P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P76, P130	140	mA
			P00, P01, P40 to P43, P120, P131, P132	60	mA
	I_{OL2}	Per pin	P80 to P87, P90 to P93	1	mA
		All pins		5	
	I_{OL3}	Per pin	P121 to P124	4	mA
		All pins		10	
Operating ambient temperature	T_A	In normal operation mode		−40 to +85	°C
		In flash memory programming mode		−40 to +85	
Storage temperature	T_{stg}			−65 to +150	°C

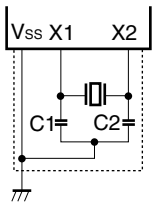
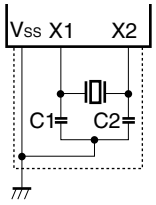
Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure 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.

27.2 Oscillator Characteristics

(1) Main System Clock (Crystal/Ceramic) Oscillator Characteristics

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)Note	4.0 V ≤ V_{DD} ≤ 5.5 V	4.0		20	MHz
			2.7 V ≤ V_{DD} < 4.0 V	4.0		10	
			1.8 V ≤ V_{DD} < 2.7 V	4.0		5.0	
Crystal resonator		X1 clock oscillation frequency (f_X)Note	4.0 V ≤ V_{DD} ≤ 5.5 V	4.0		20	MHz
			2.7 V ≤ V_{DD} < 4.0 V	4.0		10	
			1.8 V ≤ V_{DD} < 2.7 V	4.0		5.0	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the 8 MHz internal oscillator after reset, check the oscillation stabilization time of the main system clock using the oscillation stabilization time counter status register (OSTC). 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.

(2) On-chip Internal Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

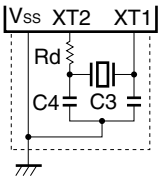
Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
8 MHz internal oscillator	Internal high-speed oscillation clock frequency (f_{RH})[Note]	RSTS = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6	8	8.4	MHz
			1.8 V ≤ V_{DD} < 2.7 V	7.6	8	10.4	MHz
		RSTS = 0		2.48	5.6	9.86	MHz
240 kHz internal oscillator	Internal low-speed oscillation clock frequency (f_{RL})	2.7 V ≤ V_{DD} ≤ 5.5 V		216	240	264	kHz
		1.8 V ≤ V_{DD} < 2.7 V		192	240	264	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Remark RSTS: Bit 7 of the internal oscillator mode register (RCM)

(3) Subsystem Clock Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})[Note]		32	32.768	35	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the high-speed system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

27.3 DC Characteristics

DC Characteristics (1/7)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output current, high[Note1]	I_{OH1}	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132			–3.0	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P70 to P76, P130			–18.0	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			–12.0	mA
		Total of pins[Note2]			–23.0	mA
	I_{OH2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			–100	μA
		Per pin for P121 to P124				
Output current, low[Note3]	I_{OL1}[Note3]	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132			8.5	mA
		Per pin for P60 to P63			15	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P76, P130			45	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			20	mA
		Total of pins[Note2]			65	mA
	I_{OL2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			400	μA
		Per pin for P121 to P124				

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7) / (n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = 20.0 mA
Total output current of pins = (20.0 × 0.7) / (50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins. High level output current and low level current are the spec in Duty = 70% conditions.

DC Characteristics (2/7)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output current, high[Note1]	I_{OH1}	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132			–2.5	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P70 to P76, P130			–15.0	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			–7.0	mA
		Total of pins[Note2]			–18.0	mA
	I_{OH2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			–100	μA
		Per pin for P121 to P124				
Output current, low	I_{OL1}[Note3]	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P130 to P132			5.0	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P76, P130			35	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			15	mA
		Total of pins[Note2]			50	mA
	I_{OL2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			400	μA
		Per pin for P121 to P124				

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7) / (n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = 20.0 mA
Total output current of pins = (20.0 × 0.7) / (50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins. High level output current and low level current are the spec in Duty = 70% conditions.

DC Characteristics (3/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} < 2.7 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132			–1.0	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P70 to P76, P130			–10	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			–5.0	mA
		Total of pins[Note2]			–15	mA
	I_{OH2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			–100	μA
		Per pin for P121 to P124				
Output current, low	I_{OL1}[Note3]	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P130 to P132			2.0	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P76, P130			20	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			9	mA
		Total of pins[Note2]			29	mA
	I_{OL2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			400	μA
		Per pin for P121 to P124				

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7) / (n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = 20.0 mA
Total output current of pins = (20.0 × 0.7) / (50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins. High level output current and low level current are the spec in Duty = 70% conditions.

DC Characteristics (4/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P12, P13, P15, P40 to P43, P50 to P53, P70, P74, P121 to P124			0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P05, P06, P10, P11, P14, P16, P17, P30 to P33, P71 to P73, P75, P76, P120, P131, P132, $\overline{\text{RESET}}$, EXCLK, EXCLKS			0.8V_{DD}		V_{DD}	V
	V_{IH3}	P80 to P87, P90 to P93		AV_{REF} = V_{DD}	0.7AV_{REF}		AV_{REF}	V
	V_{IH4}	P60 to P63			0.7V_{DD}		6.0	V
Input voltage, low	V_{IL1}	P12, P13, P15, P40 to P43, P50 to P53, P60 to P63, P70, P74, P121 to P124			0		0.3V_{DD}	V
	V_{IL2}	P00, P01, P05, P06, P10, P11, P14, P16, P17, P30 to P33, P71, P72, P73, P75, P76, P120, P131, P132, $\overline{\text{RESET}}$, EXCLK, EXCLKS			0		0.2V_{DD}	V
	V_{IL3}	P80 to P87, P90 to P93		AV_{REF} = V_{DD}	0		0.3AV_{REF}	V
Output voltage, high	V_{OH1}	I_{OH} = –3.0 mA	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132	4.0 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.7			V
		I_{OH} = –2.5 mA		2.7 V ≤ V_{DD} ≤ 4.0 V	V_{DD} – 0.5			V
		I_{OH} = –1.0 mA		1.8 V ≤ V_{DD} ≤ 2.7 V	V_{DD} – 0.5			V
	V_{OH2}	I_{OH} = –100 μA	P80 to P87, P90 to P93	AV_{REF} = V_{DD}	V_{DD} – 0.5			V
			P121 to P124					
Output voltage, low	V_{OL1}	I_{OL} = 8.5 mA	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132	4.0 V ≤ V_{DD} ≤ 5.5 V			0.7	V
		I_{OL} = 5.0 mA		2.7 V ≤ V_{DD} ≤ 4.0 V			0.7	V
		I_{OL} = 2.0 mA		1.8 V ≤ V_{DD} ≤ 2.7 V			0.5	V
		I_{OL} = 1.0 mA		1.8 V ≤ V_{DD} ≤ 2.7 V			0.5	V
		I_{OL} = 0.5 mA		1.8 V ≤ V_{DD} ≤ 2.7 V			0.4	V
	V_{OL2}	I_{OL} = 400 μA	P80 to P87, P90 to P93	AV_{REF} = V_{DD}			0.4	V
			P121 to P124					
	V_{OL3}	I_{OL} = 15 mA	P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V			2.0	V
		I_{OL} = 5.0 mA					0.4	V
		I_{OL} = 5.0 mA		2.7 V ≤ V_{DD} ≤ 4.0 V			0.6	V
		I_{OL} = 3.0 mA		2.7 V ≤ V_{DD} ≤ 4.0 V			0.4	V
		I_{OL} = 2.0 mA		1.8 V ≤ V_{DD} ≤ 2.7 V			0.4	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (5/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_I = V_{DD}$	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P131, P132, $\overline{\text{RESET}}$, FLMD0				1	μA
	I_{LIH2}	$V_I = AV_{REF}$	P80 to P87, P90 to P93	$AV_{REF} = V_{DD}$			1	μA
	I_{LIH3}	$V_I = V_{DD}$	P121 to P124	I/O port mode			1	μA
			(X1, X2, XT1, XT2)	OSC port mode			20	μA
Input leakage current, low	I_{LIL1}	$V_I = V_{SS}$	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P131, P132, $\overline{\text{RESET}}$, FLMD0				–1	μA
	I_{LIL2}		P80 to P87, P90 to P93	$AV_{REF} = V_{DD}$			–1	μA
	I_{LIL3}		P121 to P124	I/O port mode			–1	μA
			(X1, X2, XT1, XT2)	OSC port mode			–20	μA
Pull-up resistor	R_U	$V_I = V_{SS}$			10	20	100	kΩ
FLMD0 supply voltage	V_{IL}	In normal operation mode			0		$0.2V_{DD}$	V
	V_{IH}	In self programming mode			$0.8\ V_{DD}$		V_{DD}	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (6/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.3 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note1]	I_{DD1}	Operating mode	f_{XH} = 20 MHz[Note2], V_{DD} = 5.0 V	Square wave input		3.4	6.8	mA
				Resonator connection		4.7	8.2	
			f_{XH} = 10 MHz[Notes2, 3], V_{DD} = 5.0 V	Square wave input		1.8	3.6	mA
				Resonator connection		2.5	4.7	
			f_{XH} = 10 MHz[Notes2, 3], V_{DD} = 3.0 V	Square wave input		1.7	3.5	mA
				Resonator connection		2.4	4.0	
			f_{XH} = 5 MHz[Notes2, 3], V_{DD} = 3.0 V	Square wave input		1.0	2.0	mA
				Resonator connection		1.4	2.4	
			f_{XH} = 5 MHz[Notes2, 3], V_{DD} = 2.0 V	Square wave input		0.8	1.7	mA
				Resonator connection		1.1	1.9	
			f_{RH} = 8 MHz[Note4], V_{DD} = 5.0 V			1.5	2.7	mA
			f_{SUB} = 32.768 kHz[Note5], V_{DD} = 5.0 V	Square wave input		6	30	μA
				Resonator connection		15	35	
	I_{DD2}	HALT mode	f_{XH} = 20 MHz[Note2], V_{DD} = 5.0 V	Square wave input		1.0	3.9	mA
				Resonator connection		2.2	5.7	
			f_{XH} = 10 MHz[Notes2, 3], V_{DD} = 5.0 V	Square wave input		0.6	2.0	mA
				Resonator connection		1.2	3.1	
			f_{XH} = 5 MHz[Notes2, 3], V_{DD} = 3.0 V	Square wave input		0.3	1.0	mA
				Resonator connection		0.6	1.5	
			f_{RH} = 8 MHz[Note4], V_{DD} = 5.0 V			0.5	1.4	mA
			f_{SUB} = 32.768 kHz[Note5], V_{DD} = 5.0 V	Square wave input		3.0	27	μA
				Resonator connection		12	32	
	I_{DD3}[Note6]	STOP mode	V_{DD} = 5.0 V			1	20	μA

Notes 1. Total current flowing into the internal power supply (V_{DD}, EV_{DD}), including the peripheral operation current and the input leakage current flowing when the level of the input pin are fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors and the output current of the port is not included.

2. Not including the operating current of the 8 MHz internal oscillator, XT1 oscillation, 240 kHz internal oscillator and the current flowing into the A/D converter, watchdog timer and LVI circuit.

3. When AMPH (bit 0 of clock operation mode select register (OSCCTL)) = 0.

4. Not including the operating current of the X1 oscillation, XT1 oscillation and 240 kHz internal oscillator. Not including the current flowing into the A/D converter, watchdog timer, LVI circuit and CAN controller.

5. Not including the operating current of the X1 oscillation, 8 MHz internal oscillator and 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

6. Not including the operating current of the 240 kHz internal oscillator and XT1 oscillation, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

Remarks 1. f_{XH}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency or external subsystem clock frequency)

DC Characteristics (7/7)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.3 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
A/D converter operating current	I_{ADC} Note1	ADCE = 1		0.86	1.9	mA
Watchdog timer operating current	I_{WDT} Note2	During 240 kHz internal low-speed oscillation clock operation		5	10	μA
LVI operating current	I_{LVI} Note3			9	18	μA

Notes 1. Current flowing only to the A/D converter (AVREF-pin). The current value of the 78K0/FE2 is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

2. Current flowing only to the watchdog timer (V_{DD}-pin) (including the operating current of the 240 kHz internal oscillator). The current value of the 78K0/FE2 is the sum of I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates in the HALT or STOP mode.

3. Current flowing only to the LVI circuit (V_{DD}-pin). The current value of the 78K0/FE2 is the sum of I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates in the HALT or STOP mode.

27.4 AC Characteristics

(1) Basic operation

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

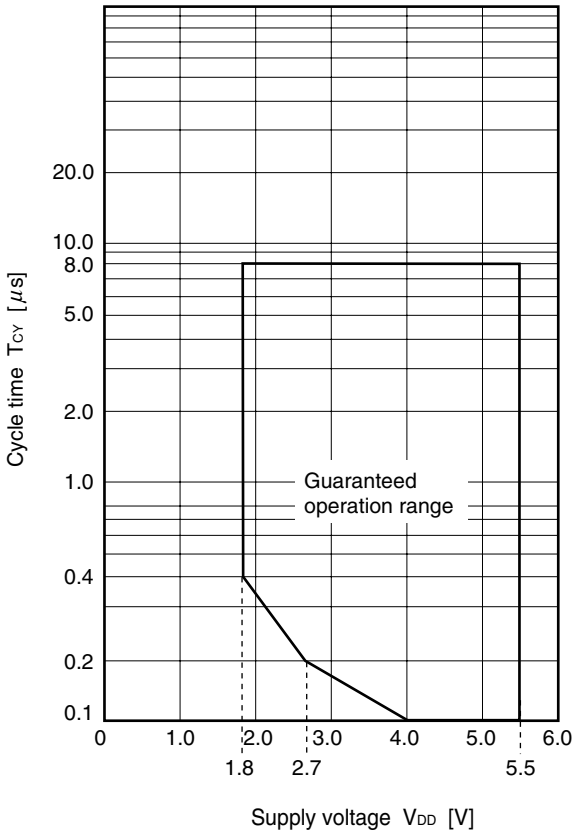
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	4.0 V ≤ V_{DD} ≤ 5.5 V	0.1		8	μs
			2.7 V ≤ V_{DD} < 4.0 V	0.2		8	μs
			1.8 V ≤ V_{DD} < 2.7 V	0.4 Note1		8	μs
		Subsystem clock (f_{SUB}) operation		114	122	125	μs
Peripheral hardware clock frequency	f_{PRS}	f_{PRS} = f_{XH}	4.0 V ≤ V_{DD} ≤ 5.5 V			20	MHz
			2.7 V ≤ V_{DD} < 4.0 V			10	
			1.8 V ≤ V_{DD} < 2.7 V			5	
		f_{PRS} = f_{RH}	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6		8.4	MHz
			1.8 V ≤ V_{DD} < 2.7 V Note2	7.6		10.4	
External main system clock frequency	f_{EXT}	4.0 V ≤ V_{DD} ≤ 5.5 V		4.0		20	MHz
		2.7 V ≤ V_{DD} < 4.0 V		4.0		10	MHz
		1.8 V ≤ V_{DD} < 2.7 V		4.0		5	MHz
External clock input high level width, low level width	f_{EXTH}, f_{EXTL}	4.0 V ≤ V_{DD} ≤ 5.5 V		24			ns
		2.7 V ≤ V_{DD} < 4.0 V		48			
		1.8 V ≤ V_{DD} < 2.7 V		96			
External subsystem clock frequency	f_{EXTS}			32	32.768	35	kHz
External sub clock input high level width, low level width	f_{EXTSH}, f_{EXTSL}			12			μs
TI000, TI001, TI002, TI003, TI010, TI011, TI012, TI013 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} ≤ 5.5 V		$2/f_{sam}$ + 0.1 Note3			μs
		2.7 V ≤ V_{DD} < 4.0 V		$2/f_{sam}$ + 0.2 Note3			μs
		1.8 V ≤ V_{DD} < 2.7 V		$2/f_{sam}$ + 0.5 Note3			μs
TI50, TI51 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				10	MHz
		1.8 V ≤ V_{DD} < 2.7 V				5	MHz
TI50, TI51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	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_{INIH}, t_{INIL}			1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			μs

Notes 1. 0.38 μs when operating with the 8 MHz internal oscillator.

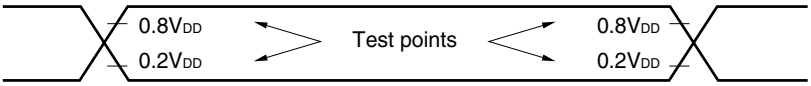
2. This spec is a definition of the main system clock. Therefore, peripheral hardware must use the clock of f_{RH}/2 or less. (V_{DD} = 1.8 V or less)

3. IT sampling with selection count clock (f_{PRS}, f_{PRS}/4, f_{PRS}/256) using bits 0 and 1 (PRM0n0, PRM0n1) of prescaler mode registers 00 (PRM0n). Note that when selecting the TI0n0 valid edge as the count clock, f_{sam} = f_{PRS}. (n = 0, 1, 2, 3)

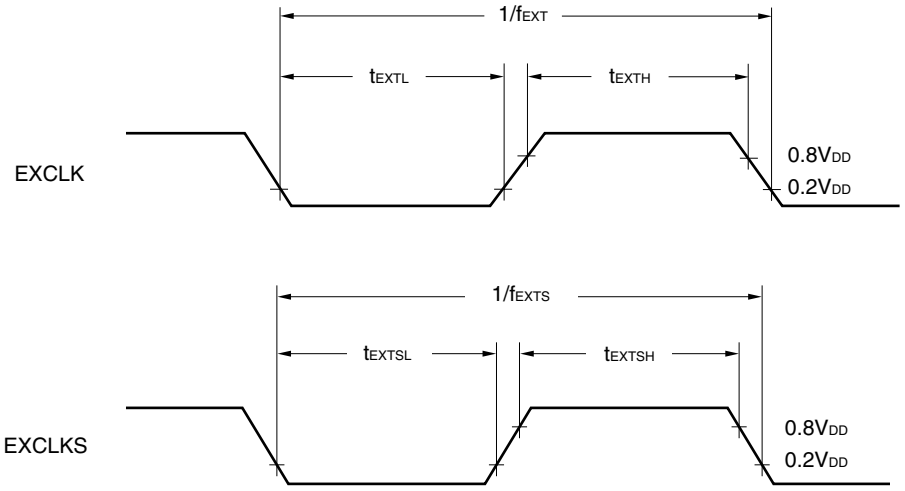
T_{CY} vs. V_{DD} (Main System Clock Operation)



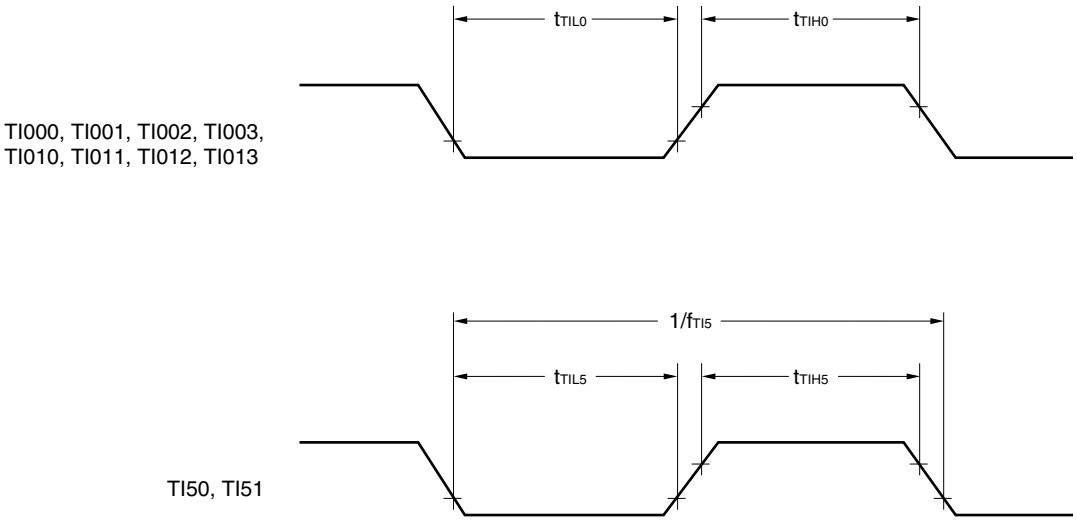
AC Timing Test Points (Excluding X1, XT1)



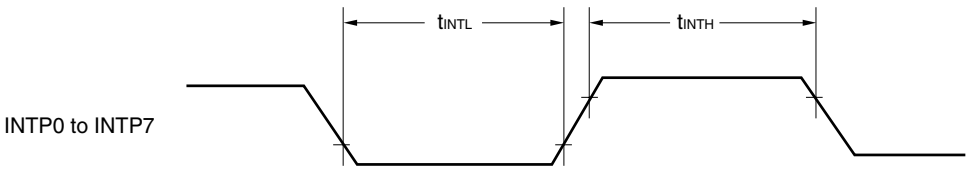
External clock input timing



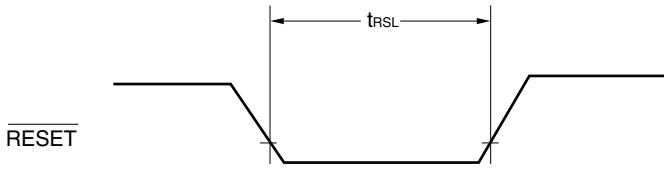
TI Timing



Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



(2) Serial interface

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) UART mode (UART6n, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

(b) 3-wire serial I/O mode (master mode, $\overline{\text{SCK1n}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ 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
$\overline{\text{SCK1n}}$ high-/low-level width[Note1]	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V	t_{KCY1}/2 – 30			
		1.8 V ≤ V_{DD} < 2.7 V	t_{KCY1}/2 – 60			
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} < 4.0 V	100			
		1.8 V ≤ V_{DD} < 2.7 V	190			
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO1}	C = 50 pF[Note2]			40	ns

Notes 1. It is value at the time of fx use. Keep in mind that spec different at the time of f_{OSC8} use.

2. C is the load capacitance of the $\overline{\text{SCK1n}}$ and SO1n output lines.

(c) 3-wire serial I/O mode (slave mode, $\overline{\text{SCK1n}}$... external clock input)

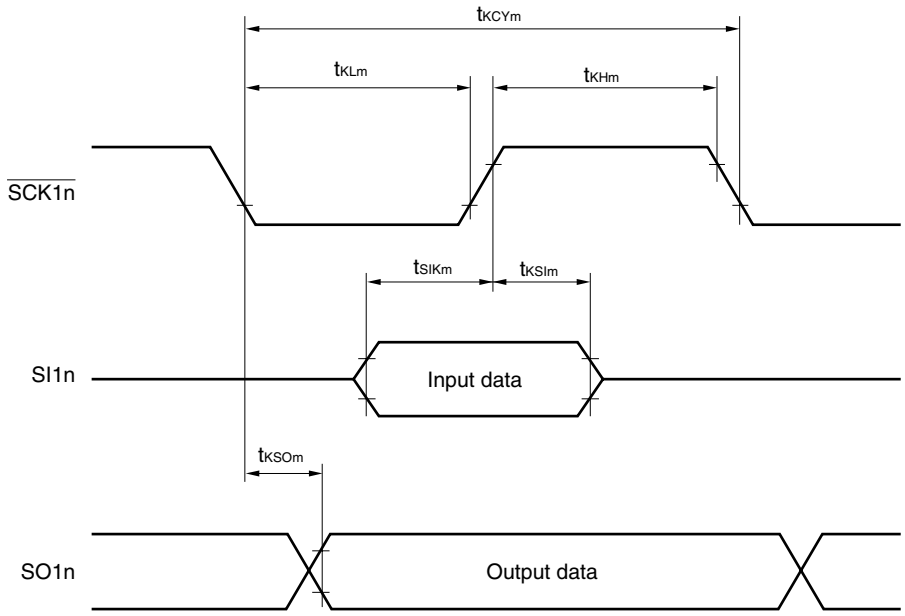
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ cycle time	t_{KCY2}			400			ns
$\overline{\text{SCK1n}}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK2}			80			ns
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI2}			50			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO2}	C = 50 pF[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V			120	ns
			2.7 V ≤ V_{DD} < 4.0 V			120	
			1.8 V ≤ V_{DD} < 2.7 V			180	

Note C is the load capacitance of the SO1n output line.

Remark n = 0, 1

Serial Transfer Timing

3-wire serial I/O mode:

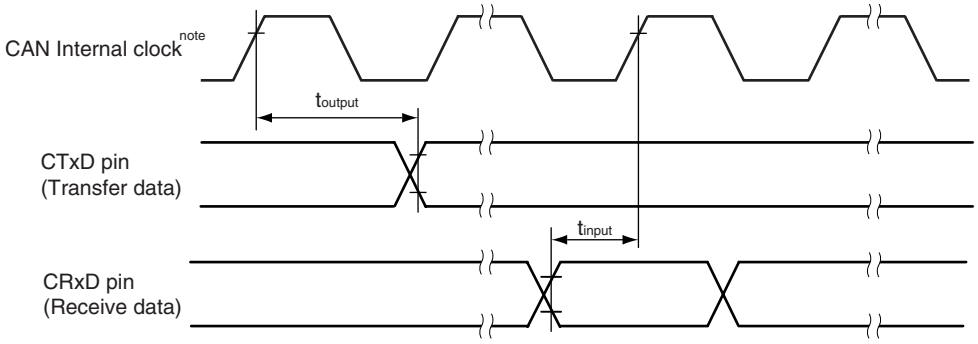


Remark m = 1, 2
n = 0, 1

(3) CAN controller

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					1	Mbps
Internal delay time	t_{NODE}				100	ns



Internal delay time (t_{NODE}) = Internal Transfer Delay (t_{output}) + Internal Receive Delay (t_{input})

Note CAN Internal clock (f_{CAN}): CAN baud rate clock

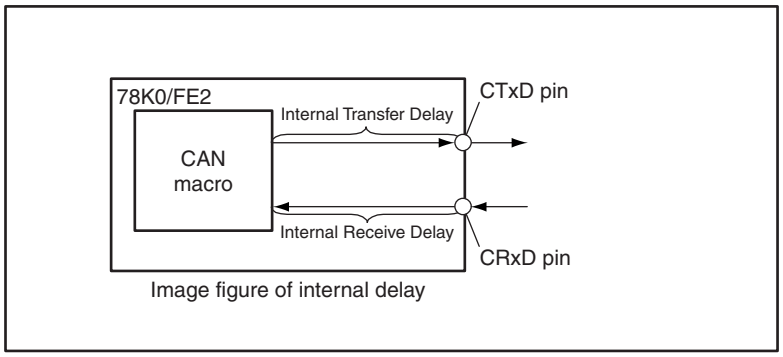


Image figure of internal delay

(4) A/D Converter Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.3 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				10	bit
Overall error[Notes1, 2]	A_{INL}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	
		2.3 V ≤ AV_{REF} < 2.7 V			±1.2	
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	6.1		36.7	μs
		2.7 V ≤ AV_{REF} < 4.0 V	12.2		36.7	
		2.3 V ≤ AV_{REF} < 2.7 V	27		66.6	
Zero-scale error[Notes1, 2]	E_{ZS}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	
		2.3 V ≤ AV_{REF} < 2.7 V			±0.6	
Full-scale error[Notes1, 2]	E_{FS}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	
		2.3 V ≤ AV_{REF} < 2.7 V			±0.6	
Integral non-linearity error[Note1]	I_{LE}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	
		2.3 V ≤ AV_{REF} < 2.7 V			±6.5	
Differential non-linearity error[Note1]	D_{LE}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.0	
		2.3 V ≤ AV_{REF} < 2.7 V			±2.0	
Analog input voltage	V_{AIN}	2.3 V ≤ AV_{REF} ≤ 5.5 V	AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

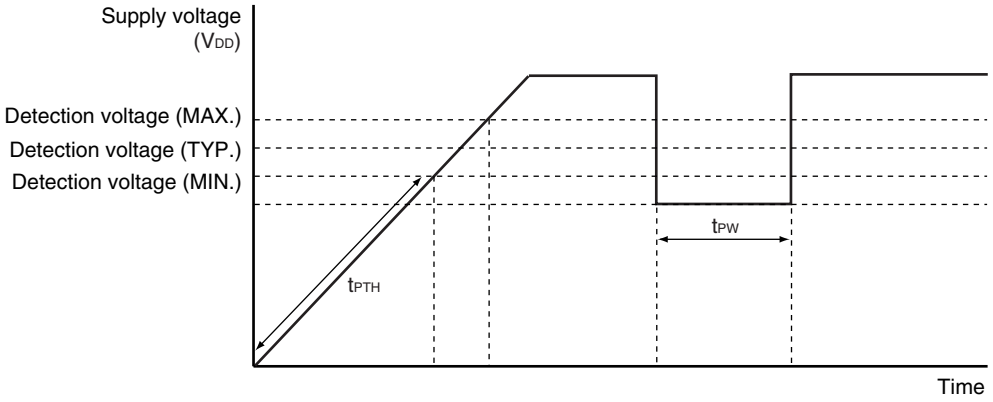
2. This value is indicated as a ratio (%FSR) to the full-scale value.

(5) POC Circuit Characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		1.44	1.59	1.74	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → V_{POC0}	0.5			V/ms
Minimum pulse width	t_{PW}		200			μs

POC Circuit Timing



Caution Spec may change after device evaluation.

(6) LVI Circuit Characteristics

(T_A = –40 to +85°C, $V_{POC} \le V_{DD} = EV_{DD} = 0$ V ≤ 5.5 V, $AV_{REF} \le V_{DD}$, $V_{SS} = EV_{SS} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.14	4.24	4.34	V
		V_{LVI1}		3.99	4.09	4.19	V
		V_{LVI2}		3.83	3.93	4.03	V
		V_{LVI3}		3.68	3.78	3.88	V
		V_{LVI4}		3.52	3.62	3.72	V
		V_{LVI5}		3.37	3.47	3.57	V
		V_{LVI6}		3.22	3.32	3.42	V
		V_{LVI7}		3.06	3.16	3.26	V
		V_{LVI8}		2.91	3.01	3.11	V
		V_{LVI9}		2.75	2.85	2.95	V
		V_{LVI10}		2.60	2.70	2.80	V
		V_{LVI11}		2.45	2.55	2.65	V
		V_{LVI12}		2.29	2.39	2.49	V
		V_{LVI13}		2.14	2.24	2.34	V
		V_{LVI14}		1.98	2.08	2.18	V
		V_{LVI15}		1.83	1.93	2.03	V
	External input pin[Note 1]	EX_{LVI}	$EX_{LVI} < V_{DD}$, 1.8 V $\le V_{DD} \le 5.5$ V	1.11	1.21	1.31	V
	Detection voltage on application of supply voltage	VDD_{LVI}	LVISTART (option bye) = 1	2.50	2.70	2.90	V
Minimum pulse width		t_{LW}		200			μs
Operation stabilization wait time[Note 2]		t_{LWAIT1}				10	μs

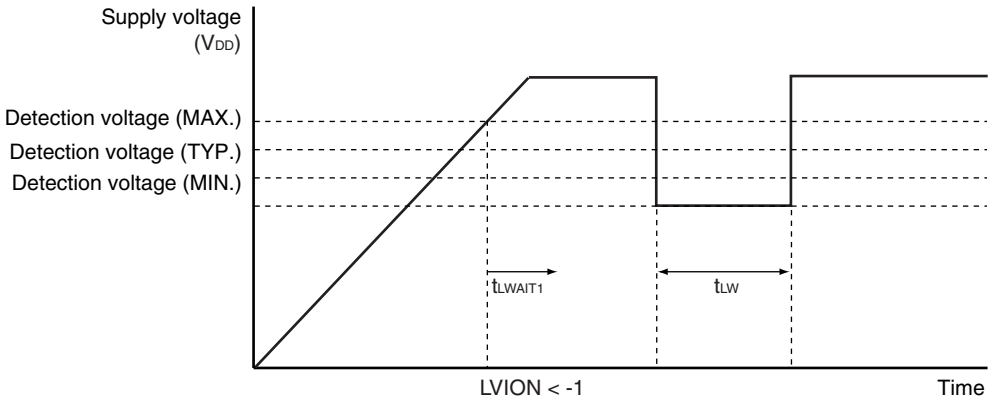
Notes 1. External input pin is alternate P120/INTP pin.

2. Time required from setting LVION to 1 to operation stabilization.

Remarks 1. $V_{LVIn-1} > V_{LVIn}$ (n = 1 to 15)

2. $V_{POC} < V_{LVIm}$ (V_{POC} : Power-on clear detection voltage, m = 0 to 15)

LVI Circuit Timing

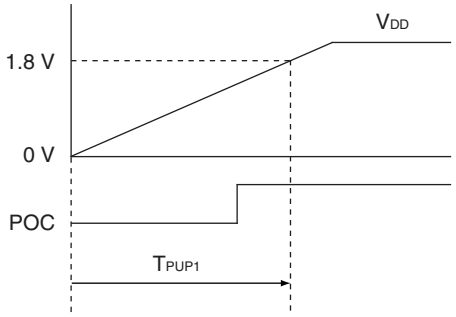


(7) Power Supply Starting Time

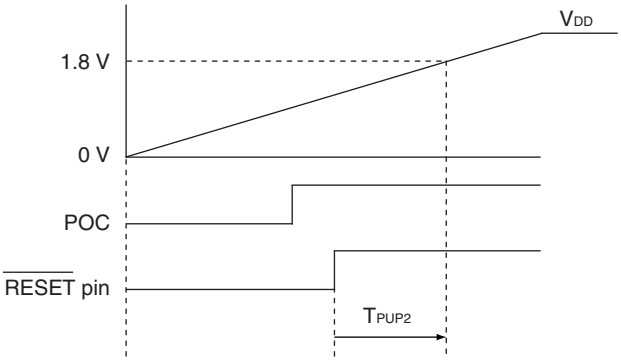
(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Starting maximum time to V_{DD} min (1.8 V)[Note] (V_{DD}: 0 V→1.8 V)	t_{PUP1}	LVI starting option invalid When pin RESET intact			3.6	ms
Starting maximum time to V_{DD} min (1.8 V)[Note] (pin RESET release→V_{DD}: 1.8 V)	t_{PUP2}	LVI starting option invalid When pin RESET use			1.9	ms

Note Start a power supply in time shorter than this when LVI staring option invalid.



Pin RESET intact



Pin RESET use

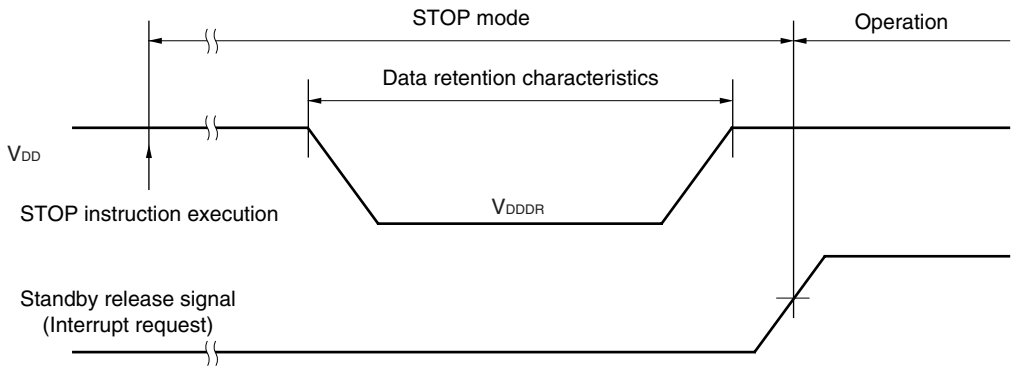
27.5 Data Retention Characteristics

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.44[Note]		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.

Data Retention Timing



27.6 Flash EEPROM Programming Characteristics

(1) Basic characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD}= EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}	f_{XP} = 10 MHz (TYP.), 20 MHz (MAX.)		4.5	11.0	mA
Erase time[Notes1, 2]	All block	T_{eraca}			20	200	ms
	Block unit	T_{erasa}			20	200	ms
Write time (in 8-bit units)[Note 1]		T_{wrwa}			10	100	μs
Number of rewrites per chip		C_{erwr}	Retention: 15 years 1 erase + 1 write after erase = 1 rewrite[Note 3]	100			Times

Notes 1. Characteristic of the flash memory. For the characteristic when a dedicated flash memory programmer, PG-FP4, is used and the rewrite time during self programming,

2. The prewrite time before erasure and the erase verify time (writeback time) are not included.

3. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

Remark SPEC may change after device evaluation.

CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)

28.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			−0.5 to +6.5	V
	EV_{DD}			−0.5 to +6.5	V
	V_{SS}			−0.5 to +0.3	V
	EV_{SS}			−0.5 to +0.3	V
	AV_{REF}			−0.5 to V_{DD} +0.3[Note]	V
	AV_{SS}			−0.5 to +0.3	V
REGC pin Input voltage	V_{REGC}			−0.5 to +3.6 and ≤ V_{DD}	V
Input voltage	V_{I1}	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P80 to P87, P90 to P93, P120, P131, P132, X1, X2, XT1, XT2, $\overline{\text{RESET}}$, FLMD0		−0.3 to V_{DD} +0.3[Note]	V
	V_{I2}	P60 to P63	N-ch open drain	−0.3 to +6.5	V
Output voltage	V_O			−0.3 to V_{DD} +0.3[Note]	V
Analog input voltage	V_{AN}	ANI0 to ANI11		−0.3 to AV_{REF} +0.3[Note] and −0.3 to V_{DD} +0.3[Note]	V
Output current, high	I_{OH}	Per pin	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130, P131, P132	−10	mA
		Total of all pins −80 mA	P05, P06, P10 to P17, P30 to P33, P50 to P53, P70 to P76, P130	−55	mA
			P00, P01, P40 to P43, P120, P131, P132	−25	mA
	I_{OH2}	Per pin	P80 to P87, P90 to P93	−0.5	mA
		Total of all pins		−2	
	I_{OH3}	Per pin	P121 to P124	−1	mA
		Total of all pins		−4	

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions		Ratings	Unit
Output current, low	I_{OL}	Per pin	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P130, P131, P132	30	mA
		Total of all pins 200 mA	P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P76, P130	140	mA
			P00, P01, P40 to P43, P120, P131, P132	60	mA
	I_{OL2}	Per pin	P80 to P87, P90 to P93	1	mA
		All pins		5	
	I_{OL3}	Per pin	P121 to P124	4	mA
		All pins		10	
Operating ambient temperature	T_A	In normal operation mode		−40 to +125	°C
		In flash memory programming mode		−40 to +125	
Storage temperature	T_{stg}			−65 to +150	°C

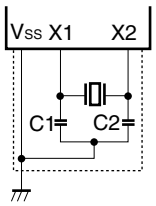
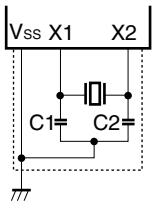
Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

28.2 Oscillator Characteristics

(1) Main System Clock (Crystal/Ceramic) Oscillator Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)Note	4.0 V ≤ V_{DD} ≤ 5.5 V	4.0		20	MHz
			2.7 V ≤ V_{DD} < 4.0 V	4.0		10	
Crystal resonator		X1 clock oscillation frequency (f_X)Note	4.0 V ≤ V_{DD} ≤ 5.5 V	4.0		20	MHz
			2.7 V ≤ V_{DD} < 4.0 V	4.0		10	

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 8 MHz internal oscillator after reset, check the oscillation stabilization time of the main system clock using the oscillation stabilization time counter status register (OSTC). 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.

(2) On-chip Internal Oscillator Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

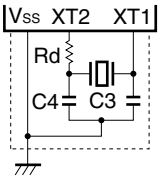
Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
8 MHz internal oscillator	Internal high-speed oscillation clock frequency (f_{RH})[Note]	RSTS = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6	8	8.46	MHz
		RSTS = 0		2.48	5.6	9.86	MHz
240 kHz internal oscillator	Internal low-speed oscillation clock frequency (f_{RL})	2.7 V ≤ V_{DD} ≤ 5.5 V		216	240	264	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Remark RSTS: Bit 7 of the internal oscillator mode register (RCM)

(3) Subsystem Clock Oscillator Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})[Note]		32	32.768	35	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the high-speed system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

28.3 DC Characteristics

DC Characteristics (1/6)

(T_A = −40 to +125°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output current, high[Note1]	I_{OH1}	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132			−1.5	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P70 to P76, P130			−10.0	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			−6.0	mA
		Total of pins[Note2]			−14.0	mA
	I_{OH2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			−100	μA
		Per pin for P121 to P124				
Output current, low[Note3]	I_{OL1}[Note3]	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132			4.0	mA
		Per pin for P60 to P63			8.0	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P76, P130			20	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			10	mA
		Total of pins[Note2]			30	mA
	I_{OL2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			400	μA
		Per pin for P121 to P124				

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7) / (n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = 20.0 mA

Total output current of pins = (20.0 × 0.7) / (50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins. High level output current and low level current are the spec in Duty = 70% conditions.

DC Characteristics (2/6)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output current, high[Note1]	I_{OH1}	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132			–1.0	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P70 to P76, P130			–8.0	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			–4.0	mA
		Total of pins[Note2]			–12.0	mA
	I_{OH2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			–100	μA
		Per pin for P121 to P124				
Output current, low	I_{OL1}[Note3]	Per pin for P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P130 to P132			2.0	mA
		Total of pins[Note2] P05, P06, P10 to P17, P30 to P33, P50 to P53, P60 to P63, P70 to P76, P130			16.0	mA
		Total of pins[Note2] P00, P01, P40 to P43, P120, P131, P132			8.0	mA
		Total of pins[Note2]			24.0	mA
	I_{OL2}	Per pin for P80 to P87, P90 to P93 $AV_{REF} = V_{DD}$			400	μA
		Per pin for P121 to P124				

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7) / (n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = 20.0 mA
Total output current of pins = (20.0 × 0.7) / (50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins. High level output current and low level current are the spec in Duty = 70% conditions.

DC Characteristics (3/6)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P12, P13, P15, P40 to P43, P50 to P53, P70, P74, P121 to P124			0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P05, P06, P10, P11, P14, P16, P17, P30 to P33, P71 to P73, P75, P76, P120, P131, P132, $\overline{\text{RESET}}$, EXCLK, EXCLKS			0.8V_{DD}		V_{DD}	V
	V_{IH3}	P80 to P87, P90 to P93		AV_{REF} = V_{DD}	0.7AV_{REF}		AV_{REF}	V
	V_{IH4}	P60 to P63			0.7V_{DD}		6.0	V
Input voltage, low	V_{IL1}	P12, P13, P15, P40 to P43, P50 to P53, P60 to P63, P70, P74, P121 to P124			0		0.3V_{DD}	V
	V_{IL2}	P00, P01, P05, P06, P10, P11, P14, P16, P17, P30 to P33, P71, P72, P73, P75, P76, P120, P131, P132, $\overline{\text{RESET}}$, EXCLK, EXCLKS			0		0.2V_{DD}	V
	V_{IL3}	P80 to P87, P90 to P93		AV_{REF} = V_{DD}	0		0.3AV_{REF}	V
Output voltage, high	V_{OH1}	I_{OH} = –1.5 mA	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132	4.0 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.7			V
		I_{OH} = –1.0 mA		2.7 V ≤ V_{DD} ≤ 4.0 V	V_{DD} – 0.7			V
	V_{OH2}	I_{OH} = –100 μA	P80 to P87, P90 to P93	AV_{REF} = V_{DD}	V_{DD} – 0.5			V
			P121 to P124					
Output voltage, low	V_{OL1}	I_{OL} = 4.0mA	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P70 to P76, P120, P130 to P132	4.0 V ≤ V_{DD} ≤ 5.5 V			0.7	V
		I_{OL} = 2.0 mA		2.7 V ≤ V_{DD} ≤ 4.0 V			0.7	V
	V_{OL2}	I_{OL} = 400 μA	P80 to P87, P90 to P93	AV_{REF} = V_{DD}			0.4	V
			P121 to P124					
	V_{OL3}	I_{OL} = 8 mA	P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V			2.0	V
		I_{OL} = 2.0 mA					0.6	V
		I_{OL} = 2.0 mA		2.7 V ≤ V_{DD} ≤ 4.0 V			0.6	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (4/6)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_I = V_{DD}$	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P131, P132, $\overline{\text{RESET}}$, FLMD0				5	μA
	I_{LIH2}	$V_I = AV_{REF}$	P80 to P87, P90 to P93	$AV_{REF} = V_{DD}$			5	μA
	I_{LIH3}	$V_I = V_{DD}$	P121 to P124	I/O port mode			5	μA
			(X1, X2, XT1, XT2)	OSC port mode			20	μA
Input leakage current, low	I_{LIL1}	$V_I = V_{SS}$	P00, P01, P05, P06, P10 to P17, P30 to P33, P40 to P43, P50 to P53, P60 to P63, P70 to P76, P120, P131, P132, $\overline{\text{RESET}}$, FLMD0				–5	μA
	I_{LIL2}		P80 to P87, P90 to P93	$AV_{REF} = V_{DD}$			–5	μA
	I_{LIL3}		P121 to P124	I/O port mode			–5	μA
			(X1, X2, XT1, XT2)	OSC port mode			–20	μA
Pull-up resistor	R_U	$V_I = V_{SS}$			10	20	100	kΩ
FLMD0 supply voltage	V_{IL}	In normal operation mode			0		$0.2V_{DD}$	V
	V_{IH}	In self programming mode			$0.8\ V_{DD}$		V_{DD}	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (5/6)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note1]	I_{DD1}	Operating mode	f_{XH} = 20 MHz[Note2], V_{DD} = 5.0 V	Square wave input		3.4	10.3	mA
				Resonator connection		4.7	12.5	
			f_{XH} = 10 MHz[Notes2, 3], V_{DD} = 5.0 V	Square wave input		1.8	5.4	mA
				Resonator connection		2.5	7.2	
			f_{XH} = 10 MHz[Notes2, 3], V_{DD} = 3.0 V	Square wave input		1.7	5.4	mA
				Resonator connection		2.4	6.0	
			f_{XH} = 5 MHz[Notes2, 3], V_{DD} = 3.0 V	Square wave input		1.0	3.0	mA
				Resonator connection		1.4	3.6	
			f_{RH} = 8 MHz[Note4], V_{DD} = 5.0 V			1.5	4.2	mA
			f_{SUB} = 32.768 kHz[Note5], V_{DD} = 5.0 V	Square wave input		6	138	μA
				Resonator connection		15	145	
	I_{DD2}	HALT mode	f_{XH} = 20 MHz[Note2], V_{DD} = 5.0 V	Square wave input		1.0	5.9	mA
				Resonator connection		2.2	8.6	
			f_{XH} = 10 MHz[Notes2, 3], V_{DD} = 5.0 V	Square wave input		0.6	3.1	mA
				Resonator connection		1.2	4.7	
			f_{XH} = 5 MHz[Notes2, 3], V_{DD} = 3.0 V	Square wave input		0.3	1.6	mA
				Resonator connection		0.6	2.4	
			f_{RH} = 8 MHz[Note4], V_{DD} = 5.0 V			0.5	2.1	mA
			f_{SUB} = 32.768 kHz[Note5], V_{DD} = 5.0 V	Square wave input		3.0	133	μA
				Resonator connection		12	138	
	I_{DD3}[Note 6]	STOP mode	V_{DD} = 5.0 V			1	100	μA

Notes 1. Total current flowing into the internal power supply (V_{DD},EV_{DD}), including the peripheral operation current and the input leakage current flowing when the level of the input pin are fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors and the output current of the port is not included.

2. Not including the operating current of the 8 MHz internal oscillator, XT1 oscillation, 240 kHz internal oscillator and the current flowing into the A/D converter, watchdog timer and LVI circuit.

3. When AMPH (bit 0 of clock operation mode select register (OSCCTL)) = 0.

4. Not including the operating current of the X1 oscillation, XT1 oscillation and 240 kHz internal oscillator. Not including the current flowing into the A/D converter, watchdog timer, LVI circuit and CAN controller.

5. Not including the operating current of the X1 oscillation, 8 MHz internal oscillator and 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

6. Not including the operating current of the 240 kHz internal oscillator and XT1 oscillation, and the current flowing into the A/D converter, watchdog timer and LVI circuit.

Remarks 1. f_{XH}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency or external subsystem clock frequency)

DC Characteristics (6/6)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
A/D converter operating current	I_{ADC} Note1	ADCE = 1		0.86	2.9	mA
Watchdog timer operating current	I_{WDT} Note2	During 240 kHz internal low-speed oscillation clock operation		5	15	μA
LVI operating current	I_{LVI} Note3			9	27	μA

Notes 1. Current flowing only to the A/D converter (AVREF-pin). The current value of the 78K0/FE2 is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

2. Current flowing only to the watchdog timer (V_{DD}-pin) (including the operating current of the 240 kHz internal oscillator). The current value of the 78K0/FE2 is the sum of I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates in the HALT or STOP mode.

3. Current flowing only to the LVI circuit (V_{DD}-pin). The current value of the 78K0/FE2 is the sum of I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates in the HALT or STOP mode.

28.4 AC Characteristics

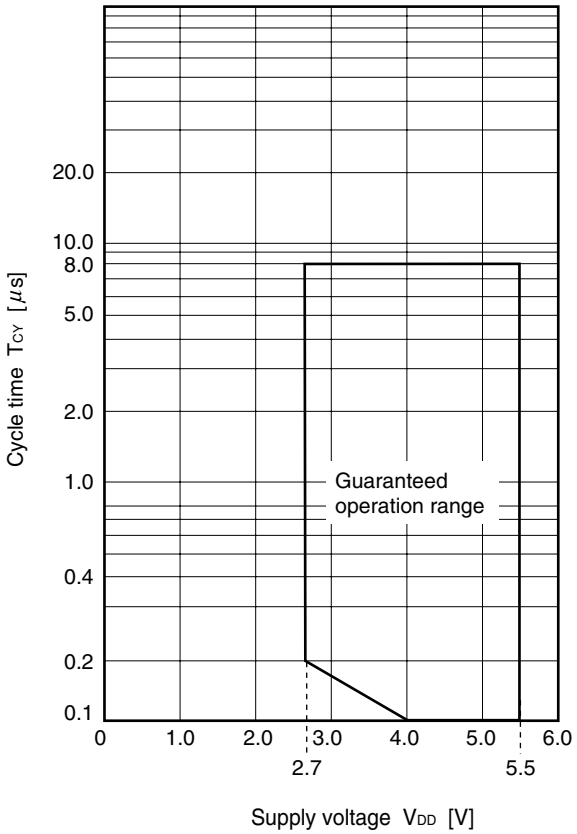
(1) Basic operation

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

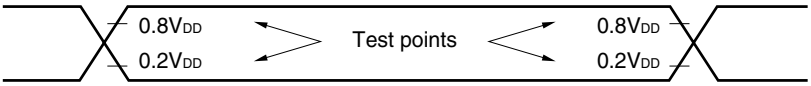
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	4.0 V ≤ V_{DD} ≤ 5.5 V	0.1		8	μs
			2.7 V ≤ V_{DD} < 4.0 V	0.2		8	μs
		Subsystem clock (f_{SUB}) operation		114	122	125	μs
Peripheral hardware clock frequency	f_{PRS}	f_{PRS} = f_{XH}	4.0 V ≤ V_{DD} ≤ 5.5 V			20	MHz
			2.7 V ≤ V_{DD} < 4.0 V			10	
		f_{PRS} = f_{RH}	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6		8.46	MHz
External main system clock frequency	f_{EXT}	4.0 V ≤ V_{DD} ≤ 5.5 V		4.0		20	MHz
		2.7 V ≤ V_{DD} < 4.0 V		4.0		10	MHz
External clock input high level width, low level width	f_{EXTH}, f_{EXTL}	4.0 V ≤ V_{DD} ≤ 5.5 V		24			ns
		2.7 V ≤ V_{DD} < 4.0 V		48			
External subsystem clock frequency	f_{EXTS}			32	32.768	35	kHz
External sub clock input high level width, low level width	f_{EXTSH}, f_{EXTSL}			12			μs
TI000, TI001, TI002, TI003, TI010, TI011, TI012, TI013 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} ≤ 5.5 V		$2/f_{sam}$ + 0.1[Note]			μs
		2.7 V ≤ V_{DD} < 4.0 V		$2/f_{sam}$ + 0.2[Note]			μs
TI50, TI51 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				10	MHz
TI50, TI51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	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_{INIH}, t_{INIL}			1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			μs

Note IT sampling with selection count clock (f_{PRS}, $f_{PRS}/4$, $f_{PRS}/256$) using bits 0 and 1 (PRM0n0, PRM0n1) of prescaler mode registers 00 (PRM0n). Note that when selecting the TI0n0 valid edge as the count clock, f_{sam} = f_{PRS}. (n = 0, 1, 2, 3)

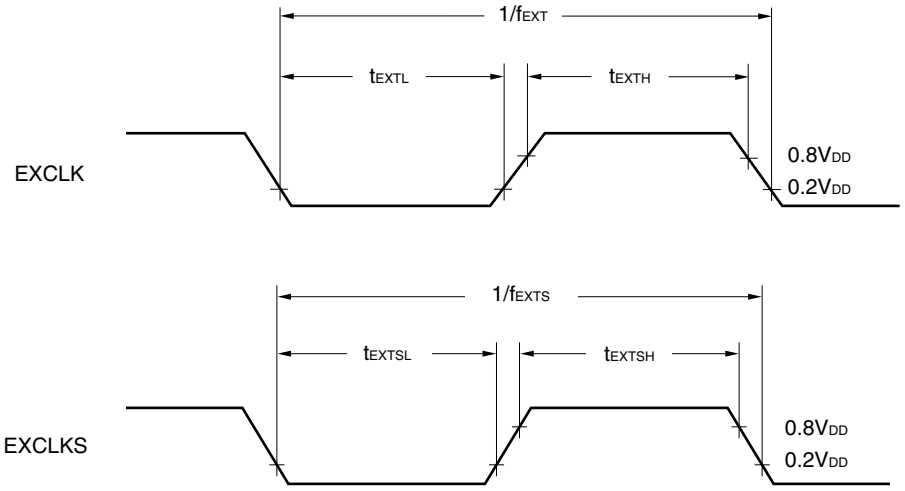
T_{CY} vs. V_{DD} (Main System Clock Operation)



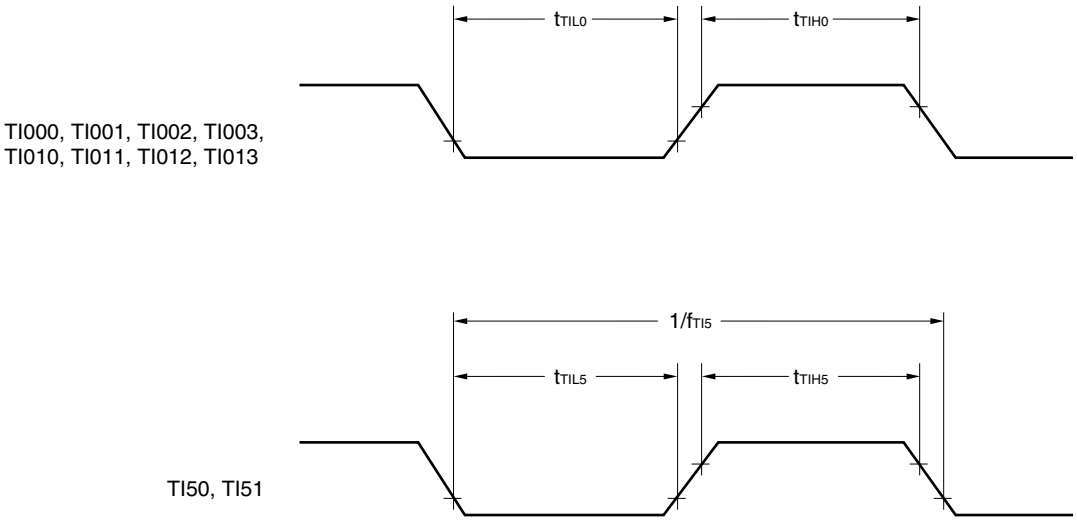
AC Timing Test Points (Excluding X1, XT1)



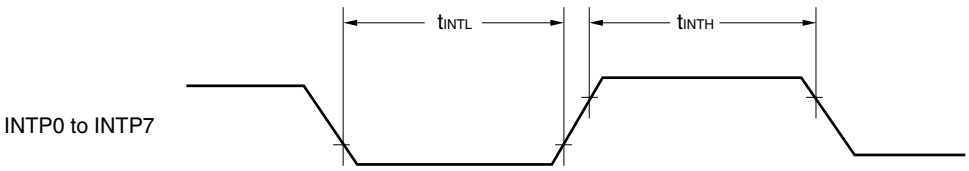
External clock input timing



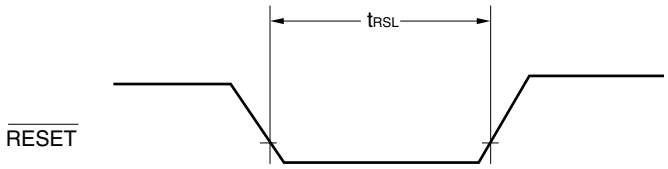
TI Timing



Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



(2) Serial interface

(T_A = –40 to +125°C, 2.7V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) UART mode (UART6n, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

(b) 3-wire serial I/O mode (master mode, $\overline{\text{SCK1n}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.0 V	400			ns
$\overline{\text{SCK1n}}$ high-/low-level width[Note1]	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V	t_{KCY1}/2 – 30			
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} < 4.0 V	100			
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO1}	C = 50 pF[Note2]			40	ns

Notes 1. It is value at the time of fx use. Keep in mind that spec different at the time of f_{OSC8} use.

2. C is the load capacitance of the $\overline{\text{SCK1n}}$ and SO1n output lines.

(c) 3-wire serial I/O mode (slave mode, $\overline{\text{SCK1n}}$... external clock input)

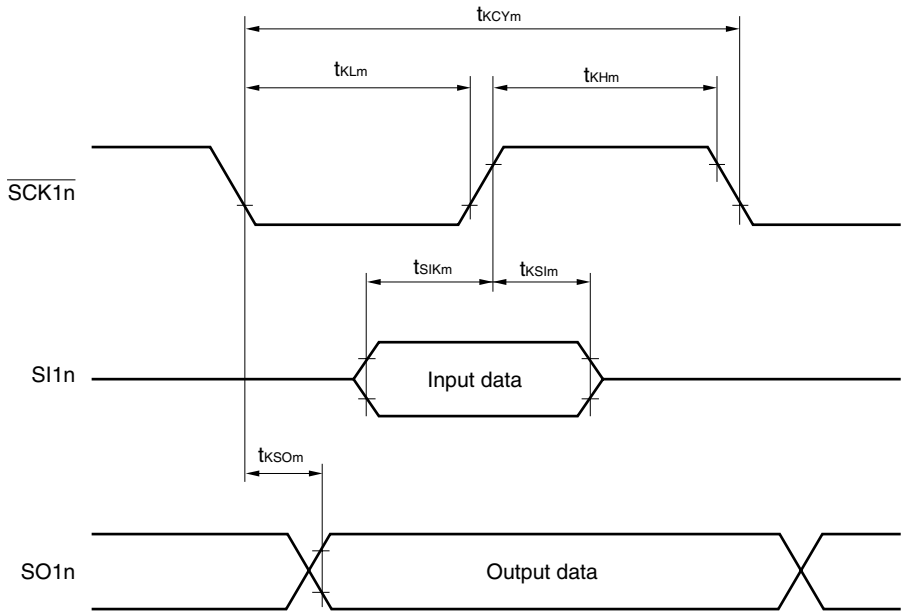
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ cycle time	t_{KCY2}			400			ns
$\overline{\text{SCK1n}}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK2}			80			ns
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI2}			50			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO2}	C = 50 pF[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V			120	ns
			2.7 V ≤ V_{DD} < 4.0 V			120	

Note C is the load capacitance of the SO1n output line.

Remark n = 0, 1

Serial Transfer Timing

3-wire serial I/O mode:

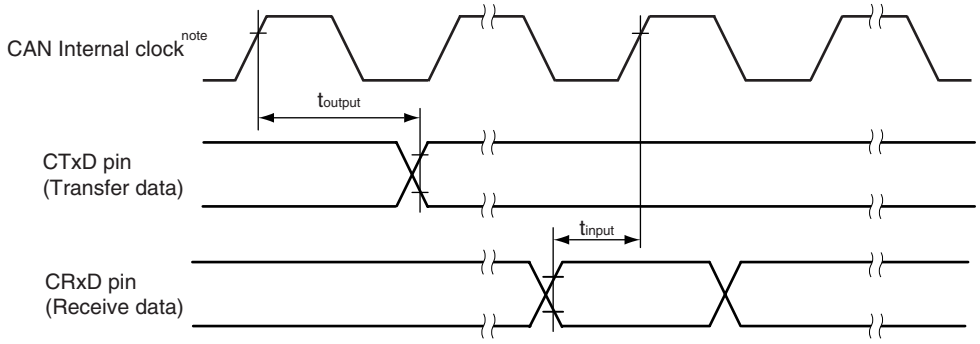


Remark m = 1, 2
n = 0, 1

(3) CAN controller

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					1	Mbps
Internal delay time	t_{NODE}				100	ns



Internal delay time (t_{NODE}) = Internal Transfer Delay (t_{output}) + Internal Receive Delay (t_{input})

Note CAN Internal clock (f_{CAN}): CAN baud rate clock

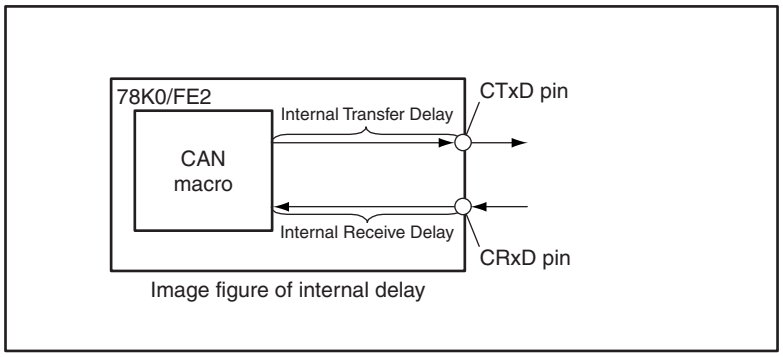


Image figure of internal delay

(4) A/D Converter Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				10	bit
Overall error[Notes1, 2]	A_{INL}	4.0 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} ≤ 4.0 V			±0.6	
Conversion time	t_{CONV}	4.0 V ≤ V_{DD} ≤ 5.5 V	6.1		36.7	μs
		2.7 V ≤ V_{DD} ≤ 4.0 V	12.2		36.7	
Zero-scale error[Notes1, 2]	E_{ZS}	4.0 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} ≤ 4.0 V			±0.6	
Full-scale error[Notes1, 2]	E_{FS}	4.0 V ≤ V_{DD} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ V_{DD} ≤ 4.0 V			±0.6	
Integral non-linearity error[Note1]	I_{LE}	4.0 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ V_{DD} ≤ 4.0 V			±4.5	
Differential non-linearity error[Note1]	D_{LE}	4.0 V ≤ V_{DD} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ V_{DD} ≤ 4.0 V			±2.0	
Analog input voltage	V_{AIN}		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

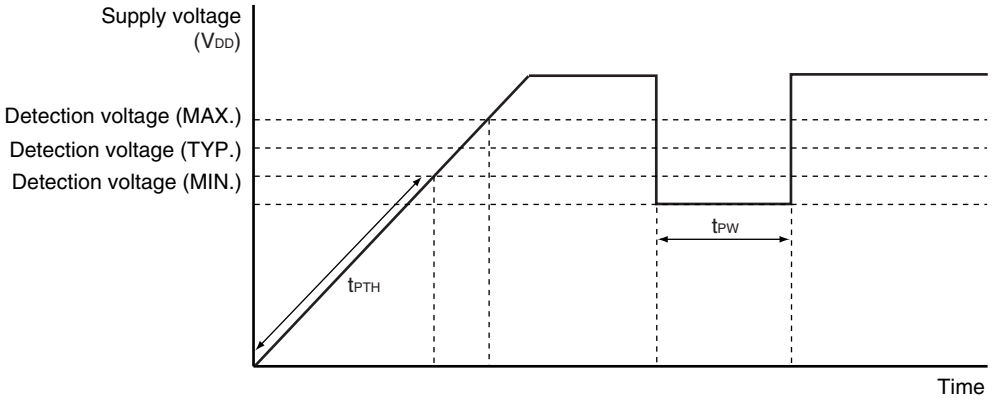
2. This value is indicated as a ratio (%FSR) to the full-scale value.

(5) POC Circuit Characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		1.44	1.59	1.74	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → V_{POC0}	0.5			V/ms
Minimum pulse width	t_{PW}		200			μs

POC Circuit Timing



Caution Spec may change after device evaluation.

(6) LVI Circuit Characteristics

(T_A = –40 to +125°C, $V_{POC} \le V_{DD} = EV_{DD} \le 5.5$ V, $AV_{REF} \le V_{DD}$, $V_{SS} = EV_{SS}$ = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.14	4.24	4.34	V
		V_{LVI1}		3.99	4.09	4.19	V
		V_{LVI2}		3.83	3.93	4.03	V
		V_{LVI3}		3.68	3.78	3.88	V
		V_{LVI4}		3.52	3.62	3.72	V
		V_{LVI5}		3.37	3.47	3.57	V
		V_{LVI6}		3.22	3.32	3.42	V
		V_{LVI7}		3.06	3.16	3.26	V
		V_{LVI8}		2.91	3.01	3.11	V
		V_{LVI9}		2.75	2.85	2.95	V
	External input pin[Note1]	EX_{LVI}	$EX_{LVI} < V_{DD}$, 2.7 V ≤ V_{DD} ≤ 5.5 V	1.11	1.21	1.31	V
	Detection voltage on application of supply voltage	VDD_{LVI}	LVISTART (option bye) = 1	2.50	2.70	2.90	V
Minimum pulse width		t_{LW}		200			μs
Operation stabilization wait time[Note2]		t_{LWAIT1}				10	μs

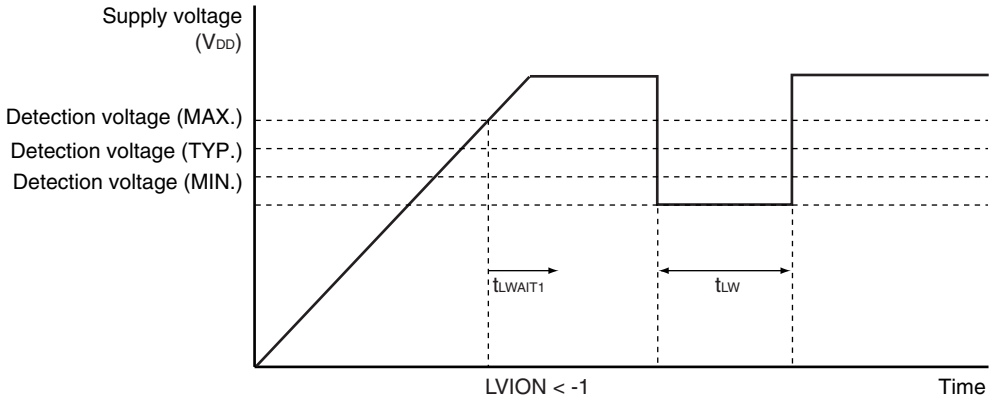
Notes 1. External input pin is alternate P120/INTP pin.

2. Time required from setting LVION to 1 to operation stabilization.

Remarks 1. $V_{LVIn-1} > V_{LVIn}$ (n = 1 to 15)

2. $V_{POC} < V_{LVIm}$ (V_{POC} : Power-on clear detection voltage, m = 0 to 15)

LVI Circuit Timing

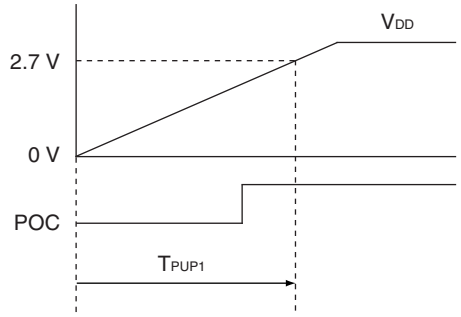


(7) Power Supply Starting Time

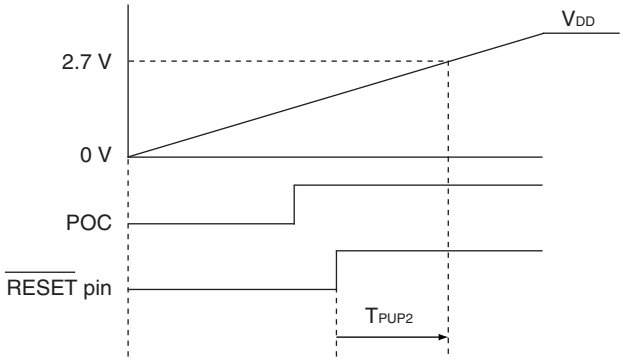
(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Starting maximum time to V_{DD} min (2.7 V)[Note] (V_{DD}: 0 V→2.7 V)	t_{PUP1}	LVI starting option invalid When pin RESET intact			3.6	ms
Starting maximum time to V_{DD} min (2.7 V)[Note] (pin RESET release→V_{DD}: 2.7 V)	t_{PUP2}	LVI starting option invalid When pin RESET use			1.9	ms

Note Start a power supply in time shorter than this when LVI staring option invalid.



Pin RESET intact



Pin RESET use

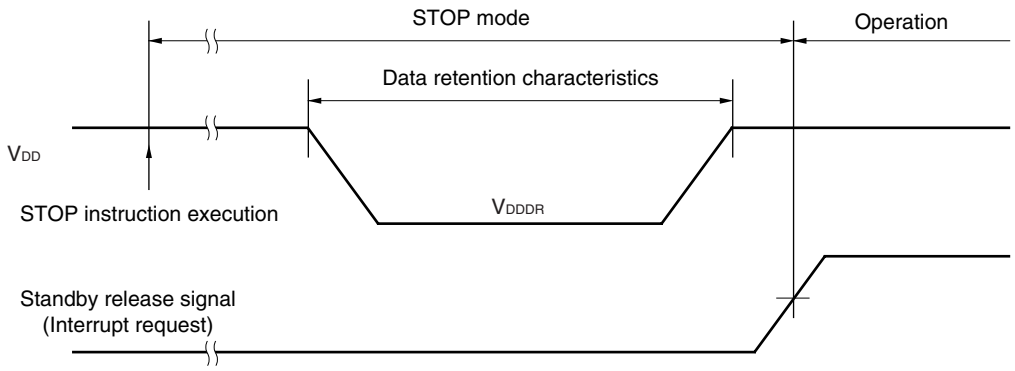
28.5 Data Retention Characteristics

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}		1.44[Note]		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.

Data Retention Timing



28.6 Flash EEPROM Programming Characteristics

(1) Basic characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}	f_{XP} = 10 MHz (TYP.), 20 MHz (MAX.)		4.5	16	mA
Erase time[Notes 1, 2]	All block	T_{eraca}			20	200	ms
	Block unit	T_{erasa}			20	200	ms
Write time (in 8-bit units)[Note 1]		T_{wrwa}			10	100	μs
Number of rewrites per chip		C_{erwr}	Retention: 15 years 1 erase + 1 write after erase = 1 rewrite[Note 3]		100		Times

Notes 1. Characteristic of the flash memory. For the characteristic when a dedicated flash memory programmer, PG-FP4, is used and the rewrite time during self programming,

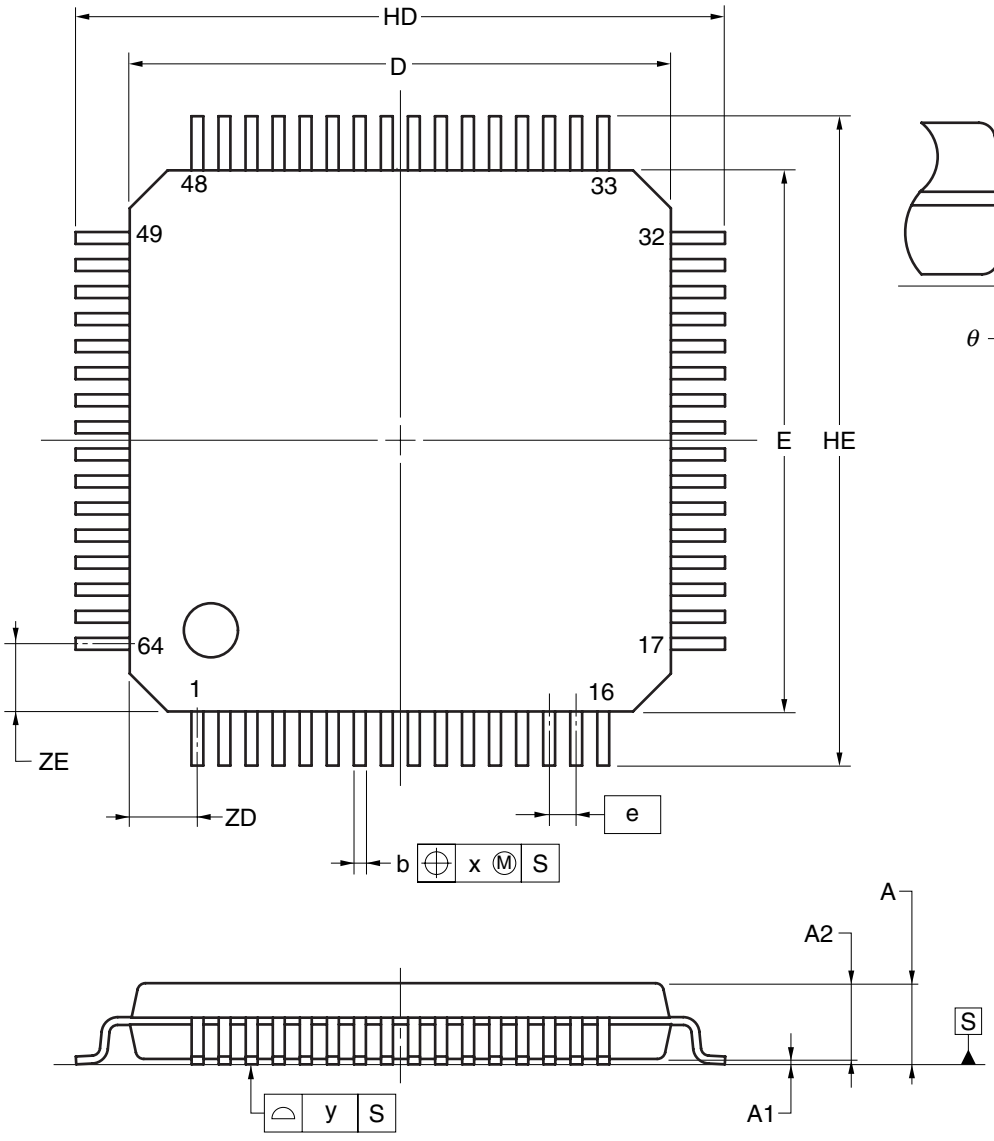
2. The prewrite time before erasure and the erase verify time (writeback time) are not included.

3. When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

Remark SPEC may change after device evaluation.

- **μPD78F0887GB(A)-GAH-AX, 78F0887GB(A2)-GAH-AX, 78F0888GB(A)-GAH-AX, 78F0888GB(A2)-GAH-AX, 78F0889GB(A)-GAH-AX, 78F0889GB(A2)-GAH-AX, 78F0890GB(A)-GAH-AX, 78F0890GB(A2)-GAH-AX**

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)



(UNIT:mm)

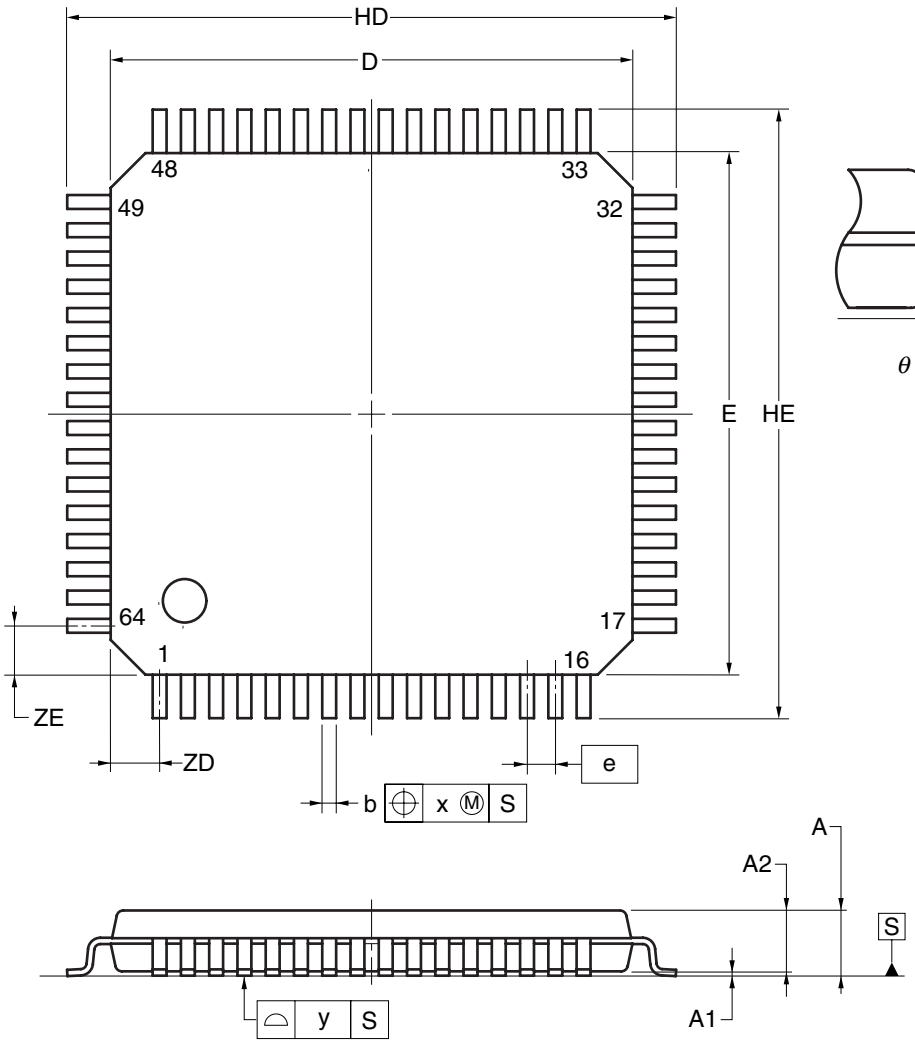
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

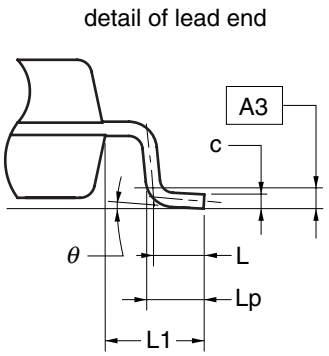
P64GB-50-GAH

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

- **μPD78F0887GK(A)-GAJ-AX, 78F0887GK(A2)-GAJ-AX, 78F0888GK(A)-GAJ-AX, 78F0888GK(A2)-GAJ-AX, 78F0889GK(A)-GAJ-AX, 78F0889GK(A2)-GAJ-AX, 78F0890GK(A)-GAJ-AX, 78F0890GK(A2)-GAJ-AX**

64-PIN PLASTIC LQFP (12x12)





(UNIT:mm)

ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.30^{+0.08}_{-0.04}$
c	$0.125^{+0.75}_{-0.25}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.65
x	0.13
y	0.10
ZD	1.125
ZE	1.125

NOTE
Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

P64GK-65-GAJ

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 NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 30-1. Surface Mounting Type Soldering Conditions

- **64-pin plastic LQFP (10 × 10)**
  μPD78F0887GB(A)-GAH-AX, 78F0887GB(A2)-GAH-AX, 78F0888GB(A)-GAH-AX, 78F0888GB(A2)-GAH-AX, 78F0889GB(A)-GAH-AX, 78F0889GB(A2)-GAH-AX, 78F0890GB(A)-GAH-AX, 78F0890GB(A2)-GAH-AX
- **64-pin plastic LQFP (12 × 12)**
  μPD78F0887GK(A)-GAJ-AX, 78F0887GK(A2)-GAJ-AX, 78F0888GK(A)-GAJ-AX, 78F0888GK(A2)-GAJ-AX, 78F0889GK(A)-GAJ-AX, 78F0889GK(A2)-GAJ-AX, 78F0890GK(A)-GAJ-AX, 78F0890GK(A2)-GAJ-AX

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 20 to 72 hours)	IR60-207-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

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, see **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

Peripheral Hardware	Register	Access	Number of Wait Clocks
Serial interface UART60	ASIS60	Read	1 clock (fixed)
Serial interface UART61	ASIS61	Read	1 clock (fixed)
A/D converter	ADM	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	
	ADPC	Write	
	ADCR	Read	
	The above number of clocks is when the same source clock is selected for f_{CPU} and f_{PRS}. 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. f_{AD}: A/D conversion clock frequency ($f_{PRS}/2$ to $f_{PRS}/12$) f_{CPU}: CPU clock frequency f_{PRS}: Peripheral hardware clock frequency f_{XP}: Main system clock frequency <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 ($f_{SUB}/2$), highest speed of A/D conversion clock ($f_{PRS}/2$)		

Caution When the CPU is operating on the subsystem clock and the peripheral hardware clock 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}).

Table 31-2 RAM Access That Generate Wait and Number of CPU Wait Clocks

Peripheral Hardware	Register	Access	number of wait clocks		Cause
			MIN.	MAX.	
CAN	Global Reg.	Read/Write	1	1	synchronizaition of NPB signals with VPCLK <Calculating number of wait clocks> MIN. ROUNDUP[(1/F_{VPCLK}) × 1/(1/F_{VPSTB})] MAX. ROUNDUP[(1/F_{VPCLK}) × 2/(1/F_{VPSTB})]
	CANmodule Reg.				
	C0RGPT	Read	2	14	Synchronization of NPB signals with VPCLK RAM access delay (1 RAM - RD access) <Calculating number of wait clocks> MIN. ROUNDUP[(1/F_{CANCLK}) × 3/(1/F_{VPSTB})] MAX. ROUNDUP[(1/F_{CANCLK}) × 4/(1/F_{VPSTB})]
	C0LIPT				
	C0TGPT				
	C0LOPT				
	Message Buf.				
	Message Buf.	Write(8 bit)	2	17	synchronization of NPB signals with VPCLK RAM access delay (1RAM - RD + 1RAM - WR access) <Calculating number of wait clocks> MIN. ROUNDUP[(1/F_{CANCLK}) × 4/(1/F_{VPSTB})] MAX. ROUNDUP[(1/F_{CANCLK}) × 5/(1/F_{VPSTB})]
	Message Buf.	Write(16 bit)	1	11	synchronization of NPB signals with VPCLK RAM access delay (1 RAM - WR access) <Calculating number of wait clocks> MIN. ROUNDUP[(1/F_{CANCLK}) × 2/(1/F_{VPSTB})] MAX. ROUNDUP[(1/F_{CANCLK}) × 3/(1/F_{VPSTB})]

Caution When Value is Φ_{CANMOD}(CAN module system clock) ≥ 2 MHz.

Remark F_{VPCLK}: VPCLK frequency
F_{VPSTB}: VPSTB frequency
F_{CANCLK}: AFCAN macro frequency

31.3 Example of Wait Occurrence

- Serial interface UART61

  <On execution of MOV A, ASIS61>

  Number of execution clocks: 6

  (5 clocks when data is read from a register that does not issue a wait (MOV A, sfr).)

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0/FE2.
Figure A-1 shows the development tool configuration.

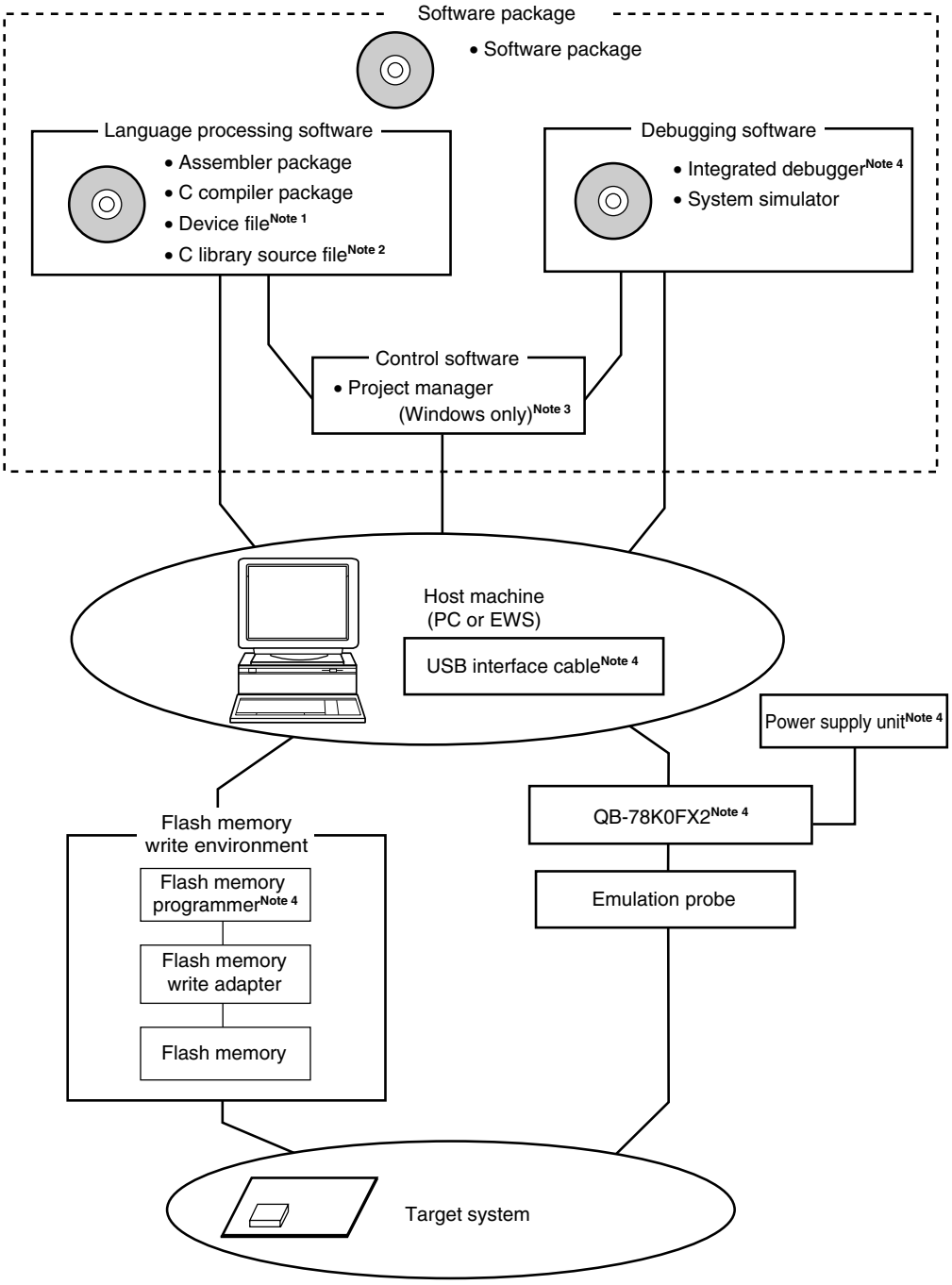
- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows™**
  Unless otherwise specified, “Windows” means the following OSs.
  - Windows 98
  - Windows NT™
  - Windows 2000
  - Windows XP™

Caution For the development tools of the 78K0/FE2, contact an NEC Electronics sales representative.

Figure A-1. Development Tool Configuration (1/3)

(1) When using the in-circuit emulator QB-78K0FX2



Notes 1. Download the device file for 78K0/FE2 (DF780893) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

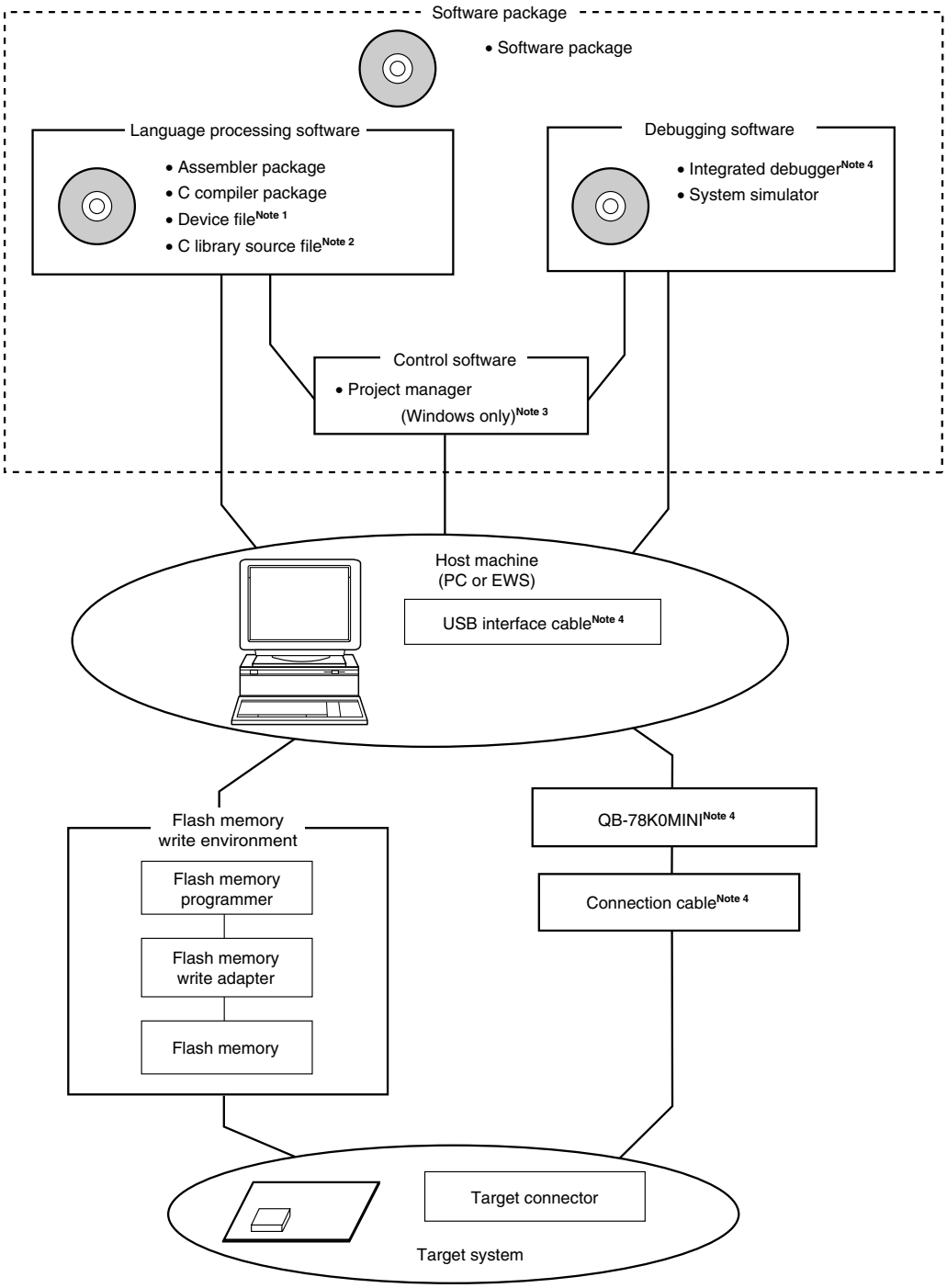
2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
PM+ is only used for Windows.

4. In-circuit emulator QB-78K0FX2 is supplied with integrated debugger ID78K0-QB, simple flash memory programmer PG-FPL3, power supply unit, and USB interface cable. Any other products are sold separately.

Figure A-1. Development Tool Configuration (2/3)

(2) When using the on-chip debug emulator QB-78K0MINI



Notes 1. Download the device file for 78K0/FE2 (DF780893) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

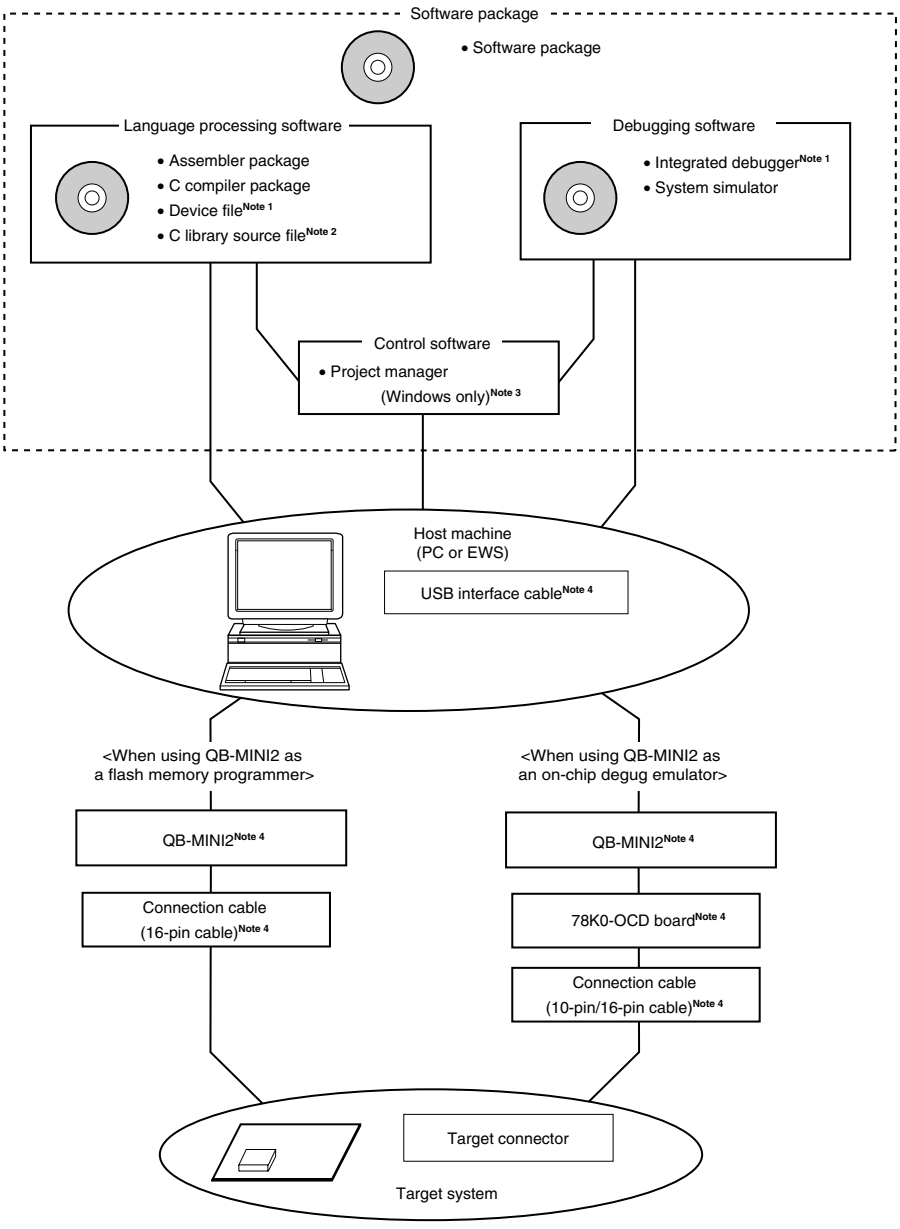
2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
PM+ is only used for Windows.

4. On-chip debug emulator QB-78K0MINI is supplied with integrated debugger ID78K0-QB, USB interface cable, and connection cable. Any other products are sold separately.

Figure A-1. Development Tool Configuration (3/3)

(3) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file for 78K0/FE2 (DF780893) and the integrated debugger (ID78K0-QB) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

4. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

A.1 Software Package

SP78K0 78K/0 Series software package	Development tools (software) common to the 78K/0 Series are combined in this package.
	Part number: μS××××SP78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

RA78K0 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 (DF780893). **<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 (included in assembler package) on Windows.
	Part number: μS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Precaution when using 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 (included in assembler package) on Windows.
	Part number: μS××××CC78K0
DF780893[Note 1] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0, CC78K0, and ID78K0-QB) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used (all sold separately).
	Part number: μS××××DF780893
CC78K/0-L[Note 2] C library source file	This is a source file of the functions that configure the object library included in the C compiler package (CC78K0). This file is required to match the object library included in the C compiler package to the user's specifications.
	Part number: μS××××CC78K0-L

Notes 1. The DF780893 can be used in common with the RA78K0, CC78K0, and ID78K0-QB. Download the DF780893 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The CC78K0-L is not included in the software package (SP78K0).

Remark ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF780893

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	

A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from PM+. **<Caution>** PM+ is included in the assembler package (RA78K0). It can only be used in Windows.

A.4 Flash Memory Programming Tools

A.4.1 When using flash memory programmer FG-FP4, FL-PR4, PG-FPL3, and FP-LITE3

PG-FP4, FL-PR4 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
PG-FPL3, FP-LITE3 Simple flash memory programmer	Simple flash memory programmer dedicated to microcontrollers with on-chip flash memory.

Remark FL-PR4, FP-LITE3 are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0/Fx2. 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)

Remarks 1. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. A connection cable (10-pin cable) and the 78K0-OCD board are used only when using the on-chip debug function.

2. Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

A.5 Debugging Tools (Hardware)

A.5.1 When using in-circuit emulator QB-78K0FX2

QB-78K0FX2[Note] In-circuit emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0/Fx2. It supports 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.
QB-144-CA-01 Check pin adapter	This adapter is used in waveform monitoring using the oscilloscope, etc.
QB-80-EP-01T Emulation probe	This emulation probe is flexible type and used to connect the in-circuit emulator and target system.
<R> QB-64GB-EA-03T <R> QB-64GK-EA-03T <R> Exchange adapter	This adapter is used to perform the pin conversion from the in-circuit emulator to the target connector. • QB-64GB-EA-03T: For 64-pin plastic LQFP (GB-UEU, GB-GAH type) • QB-64GK-EA-03T: For 64-pin plastic LQFP (GK-UET, GK-GAJ type)
QB-64GB-YS-01T QB-64GK-YS-01T Space adapter	This space adapter is used to adjust the height between the target system and in-circuit emulator. • QB-64GB-YS-01T: For 64-pin plastic LQFP (GB-UEU, GB-GAH type) • QB-64GK-YS-01T: For 64-pin plastic LQFP (GK-UET, GK-GAJ type)
QB-64GB-YQ-01T QB-64GK-YQ-01T YQ connector	This YQ connector is used to connect the target connector and exchange adapter. • QB-64GB-YQ-01T: For 64-pin plastic LQFP (GB-UEU, GB-GAH type) • QB-64GK-YQ-01T: For 64-pin plastic LQFP (GK-UET, GK-GAJ type)
QB-64GB-HQ-01T QB-64GK-HQ-01T Mount adapter	This mount adapter is used to mount the target device with socket. • QB-64GB-HQ-01T: For 64-pin plastic LQFP (GB-UEU, GB-GAH type) • QB-64GK-HQ-01T: For 64-pin plastic LQFP (GK-UET, GK-GAJ type)
QB-64GB-NQ-01T QB-64GK-NQ-01T Target connector	This target connector is used to mount on the target system. • QB-64GB-NQ-01T: For 64-pin plastic LQFP (GB-UEU, GB-GAH type) • QB-64GK-NQ-01T: For 64-pin plastic LQFP (GK-UET, GK-GAJ type)

Note The QB-78K0FX2 is supplied with a power supply unit, USB interface cable, and flash memory programmer PG-FPL3. It is also supplied with integrated debugger ID78K0-QB as control software.

Remark The package contents differ depending on the part number.

Package Contents / Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	YQ Connector	Target Connector
QB-78K0FX2-ZZZ (-EE)	QB-78K0FX2	Not included			
QB-78K0FX2-T64GB		QB-80-EP-01T	QB-64GB-EA-01T	QB-64GB-YQ-01T	QB-64GB-NQ-01T
QB-78K0FX2-T64GK			QB-64GK-EA-01T	QB-64GK-YQ-01T	QB-64GK-NQ-01T

A.5.2 When using on-chip debug emulator QB-78K0MINI

QB-78K0MINI On-chip debug emulator	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0/Fx2. It supports the integrated debugger (ID78K0-QB). This emulator should be used in connection cable and a USB interface cable that is used to connect the host machine.
Target connector specifications	10-pin general-purpose connector (2.54 mm pitch)

Remark The QB-78K0MINI is supplied with a USB interface cable and a connection cable. As control software, the integrated debugger ID78K0-QB is supplied.

A.5.3 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. 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 (10-pin cable or 16-pin cable), a USB interface cable that is used to connect the host machine, and the 78K0-OCD board.
Target connector specifications	10-pin general-purpose connector (2.54 mm pitch) or 16-pin general-purpose connector (2.54 mm pitch)

Remarks 1. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. A connection cable (10-pin cable) and the 78K0-OCD board are used only when using the on-chip debug function.

2. Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

A.6 Debugging Tools (Software)

SM+ for 78K0/Fx2[Note] System simulator	SM+ for 78K0/Fx2 is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of SM+ for 78K0/Fx2 allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. SM+ for 78K0/FX2 should be used in combination with the device file (DF780893).
ID78K0-QB Integrated debugger	This debugger supports the in-circuit emulators for the 78K/0 Series. 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.
	Part number: μS××××ID78K0-QB

Note This product is under development

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID78K0-QB

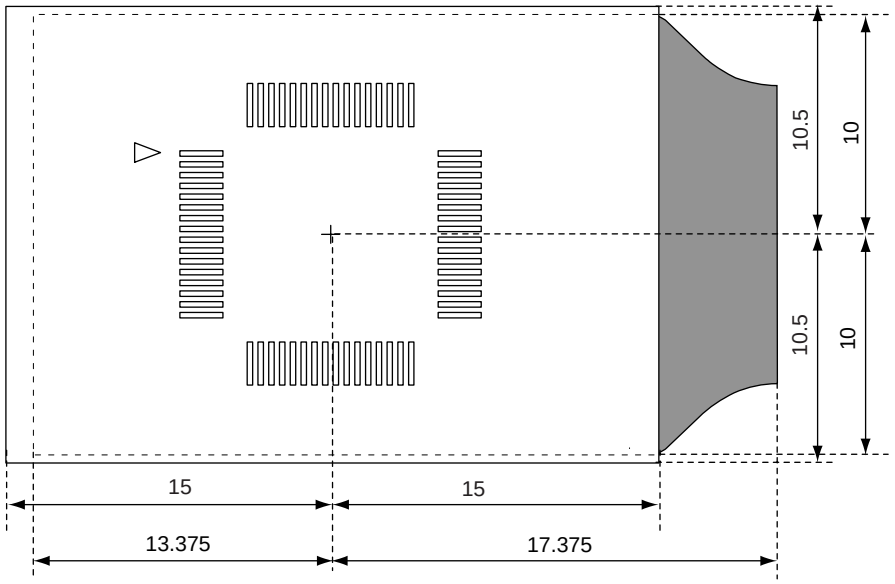
××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX B NOTES ON TARGET SYSTEM DESIGN

This chapter shows areas on the target system where component mounting is prohibited and areas where there are component mounting height restrictions when the QB-78K0FX2 is used.

(a) Case of 64-pin GB package

Figure B-1. The Restriction Domain on a Target System (Case of 64-pin GB Package)



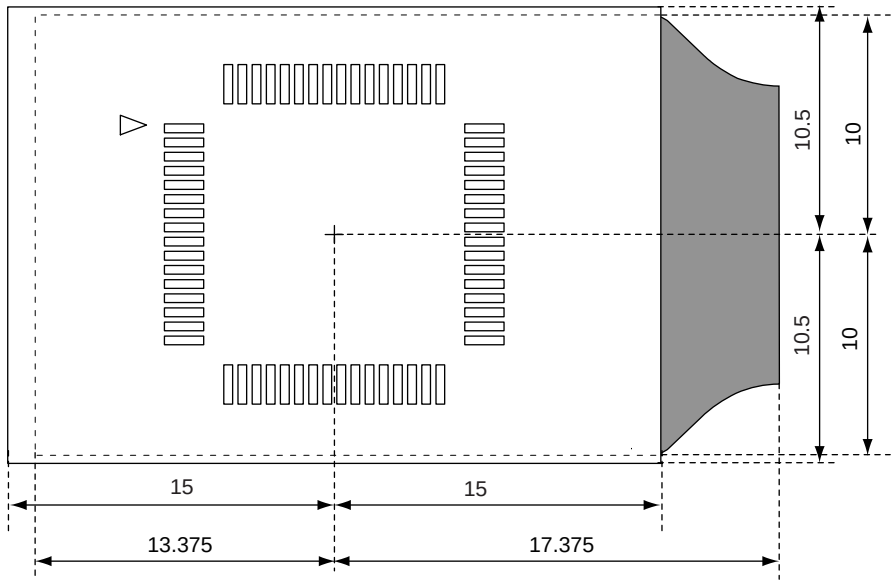
□: Exchange adapter area: Components up to 17.45 mm in height can be mounted[Note]

■: Emulation probe tip area: Components up to 24.45 mm in height can be mounted[Note]

Note Height can be adjusted by using space adapters (each adds 2.4 mm)

(b) Case of 64-pin GK package

Figure B-2. The Restriction Domain on a Target System (Case of 64-pin GK Package)



□: Exchange adapter area: Components up to 17.45 mm in height can be mounted[Note]

■: Emulation probe tip area: Components up to 24.45 mm in height can be mounted[Note]

Note Height can be adjusted by using space adapters (each adds 2.4 mm)

APPENDIX C REGISTER INDEX

C.1 Register Index (In Alphabetical Order with Respect to Register Names)

[O]

[P]

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

[W]

APPENDIX D REVISION HISTORY

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 "Fine what:" field.

D.1 Main Revisions in this Edition

Page	Description
p.7	Addition of QB-MINI2 in **Documents Related to Development Tools (Hardware) (User's Manuals)** Change of PG-FPL3 in **Documents Related to Flash Memory Programming**
p.22	Change of table in **1.5.1 78K0/Fx2 product lineup**
p.25	Change of table in **1.7 Outline of Functions**
p.98	Change of **Caution 1** in **5.2.3 Port 3**
p.103	Change of **Figure 5-12. Block Diagram of P70**
p.105	Change of **Figure 5-14. Block Diagram of P72 and P73**
p.106	Change of **Figure 5-15. Block Diagram of P74**
p.107	Change of **Figure 5-16. Block Diagram of P76**
p.124	Change of **Figure 5-27. Bit Manipulation Instruction (P10)**
p.275	Change of table in **11.4.1 Controlling operation of watchdog timer**
p.287	Change of the explanation in **13.2 (9) AV_{REF} pin** Change of the explanation in **13.2 (12) A/D port configuration register (ADPC)**
p.510	Change of **Figure 16-55. Clear CAN Sleep/Stop Mode**
p.518	Change of the explanation in **17.1 (1) Maskable interrupts** Change of the explanation in **17.2 Interrupt Sources and Configuration**
p.520	Change of **Table 17-1. Interrupt Source List**
p.524	Change of **Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)**
p.526	Change of **Figure 17-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)**
p.527	Change of **Figure 17-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)**
p.616	Deletion of **(2) μPD78F0888 (internal ROM capacity: 60 KB)** in **Table 24-12. Processing Time for Each Command When PG-FP4 Is Used (Reference)**
p.707	Change of **A.5.1 When using in-circuit emulator QB-78K0FX2**

D.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/9)

Edition	Description
3rd	Addition of PG-FPL3 in **Documents Related to Flash Memory Programming**
	Addition of **Caution 1 to 4** to **1.4 Pin Configuration (Top View)**
	Change of 10-bit A/D converter number for 78K0/FC2 in **1.5.1 78K0/Fx2 product lineup**
	Change of EV_{DD} and V_{DD} in **Table 2-1. Pin I/O Buffer Power Supplies**
	Change of REGC in **Table 2-3. Non-port pins (2/2)**
	Change of **2.2.15 REGC**
	Change of **Figure 3-1. Memory Map (μPD78F0887)** Addition of **Note 3 and 4** and **Remark** in **Figure 3-1. Memory Map (μPD78F0887)**
	Change of **Figure 3-2. Memory Map (μPD78F0888)** Addition of **Note 3 and 4** and **Remark** in **Figure 3-2. Memory Map (μPD78F0888)**
	Change of **Figure 3-3. Memory Map (μPD78F0889)** Addition of **Note 3 and 4** and **Remark** in **Figure 3-3. Memory Map (μPD78F0889)**
	Change of **Figure 3-4. Memory Map (μPD78F0890)** Addition of **Note 3 and 4** and **Remark** in **Figure 3-4. Memory Map (μPD78F0890)**
	Addition of **Table 3-2. Correspondence Between Address Values and Block Numbers in Flash Memory**
	Addition of **(5) On-chip debug security ID setting area** in **3.1.1 Internal program memory space** Addition of **Caution 4** in **3.1.2 Bank area (μPD78F0889 and 78F0890 only)**
	Addition of **Note 1** in **Table 3-8. Special Function Register List (4/6)**
	Change of **Note** in **Table 3-8. Special Function Register List (6/6)**
	Addition of **CHAPTER 4 MEMORY BANK SELECT FUNCTION (μPD78F0889, 78F0890 ONLY)**
	Change of EV_{DD} and V_{DD} in **Table 5-1. Pin I/O Buffer Power Supplies**
	Addition of **Caution** in **5.2.1 Port 0**
	Addition of **Caution** in **5.2.2 Port 1**
	Addition of **Caution 1** in **5.2.3 Port 3**
	Change of the explanation in **5.2.6 Port 6**
	5.2.8 Port 8 • Change of the explanation • Addition of **Table 5-3. Setting Functions of P80/ANI0 to P87/ANI7 Pins and Caution**
	5.2.9 Port 9 • Change of the explanation • Addition of **Table 5-4. Setting Functions of P90/ANI8 to P93/ANI11 Pins and Caution**
	Addition of **Caution 1** in **5.2.10 Port 12**
	Change of **Figure 5-17. Block Diagram of P120**
	Change of **Figure 5-18. Block Diagram of P121 to P124**
	Addition of ADPC in **5.3 Registers Controlling Port Function**
	Addition of **(4) A/D port configuration register (ADPC)** in **5.3 Registers Controlling Port Function**
	Addition of **5.5 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn)**
	Change of **Caution 1** and Addition of **Caution 3** in **6.3 (4) Main OSC control register (MOC)**
	Change of **Caution 2 and 3** in **6.3 (5) Clock operation mode select register (OSCCTL)**
	Addition of the explanation in **6.4.1 X1 oscillator** and **6.4.2 XT1 oscillator** Addition of **(b) External clock** in **Figure 6-9. Example of External Circuit of X1 Oscillator** and **Figure 6-10. Example of External Circuit of XT1 Oscillator**

(2/9)

Edition	Description
3rd	Change of explanation and **Remark** in **6.4.3 When subsystem clock is not used**
	Figure 6-12 Operation of the clock generating circuit when power supply voltage injection (When 1.59 V POC mode setup (option byte: LVISTART = 0)) • Change of **Figure 6-12** and **Note 2** • Addition of **Note 1**
	Figure 6-13 Operation of the clock generating circuit when power supply voltage injection (When 2.7 V/1.59V POC mode setup (option byte: LVISTART = 1)) • Change of **Figure 6-13** and **Caution 2** • Addition of **Caution 1**
	Change of **6.6.1 Controlling high-speed system clock**
	Change of explanation and Addition of **Note** in **6.6.1 (2) Example of setting procedure when using the external main system clock**
	Change of **6.6.2 Example of controlling internal high-speed oscillation clock**
	Change of **6.6.3 Example of controlling subsystem clock**
	Figure 6-14. CPU Clock Status Transition Diagram • Change of **Figure 6-14** • Addition of **Remark**
	Change of **Table 6-5. Changing CPU Clock**
	Addition of **6.6.8 Time required for switchover of CPU clock and main system clock**
	Addition of **6.6.9 Conditions before clock oscillation is stopped**
	Change of explanation and Addition of **Caution** in **7.2 (1) 16-bit timer counter 0n (TM0n)**
	Addition of **Caution** in **7.2 (2) 16-bit timer capture/compare register 00n (CR00n)**
	Addition of **7.2 (4) Setting range when CR00n or CR01n is used as a compare register**
	Change of explanation in **7.3 (1) 16-bit timer mode control register 0n (TMC0n)**
	Change of **Figure 7-8. Format of 16-Bit Timer Mode Control Register 00 (TMC00)**
	Change of **Figure 7-9. Format of 16-Bit Timer Mode Control Register 01 (TMC01)**
	Change of **Figure 7-10. Format of 16-Bit Timer Mode Control Register 02 (TMC02)**
	Change of **Figure 7-11. Format of 16-Bit Timer Mode Control Register 03 (TMC03)**
	Change of explanation in **7.3 (2) Capture/compare control register 0n (CRC0n)** Change of **Figure 7-12. Format of Capture/Compare Control Register 00 (CRC00)**
	Addition of **Figure 7-13. Example of CR01n Capture Operation (When Rising Edge Is Specified)** Change of **Figure 7-14. Format of Capture/Compare Control Register 01 (CRC01)**
	Change of **Figure 7-15. Format of Capture/Compare Control Register 02 (CRC02)**
	Change of **Figure 7-16. Format of Capture/Compare Control Register 03 (CRC03)**
	Change of explanation and Addition of **Caution** in **7.3 (3) 16-bit timer output control register 0n (TOC0n)**
	Change of **Figure 7-17. Format of 16-Bit Timer Output Control Register 00 (TOC00)**
	Change of **Figure 7-18. Format of 16-Bit Timer Output Control Register 01 (TOC01)**
	Change of **Figure 7-19. Format of 16-Bit Timer Output Control Register 02 (TOC02)**
	Change of **Figure 7-20. Format of 16-Bit Timer Output Control Register 03 (TOC03)**
	Change of explanation and **Caution 1 to 3** in **7.3 (4) Prescaler mode register 0n (PRM0n)**
	Addition of **7.5 Special Use of TM0n**
	Addition of **7.6 Cautions for 16-Bit Timer/Event Counters 00 and 01**

(3/9)

Edition	Description
3rd	Change of explanation and **Caution** in **9.2 (1) 8-bit timer H compare register 0n (CMP0n)** Change of **9.2 (2) 8-bit timer H compare register 1n (CMP1n)**
	Change of **Caution 1** in **Figure 9-5. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)** Change of **Figure 9-6. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)**
	Change of **Caution 1** in **Figure 9-6. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)** Change of **Figure 9-7. Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)**
	Change of WTM0 bit in **Figure 10-2. Format of Watch Timer Operation Mode Register (WTM)**
	Change of explanation in **10.4.1 Watch timer operation** Change of **Table 10-4. Watch Timer Interrupt Time**
	Change of **Figure 10-3. Operation Timing of Watch Timer/Interval Timer**
	Change of explanation in **11.1 Functions of Watchdog Timer**
	Change of explanation in **Table 11-2. Setting of Option Bytes and Watchdog Timer** and **Figure 11-1. Block Diagram of Watchdog Timer**
	Change of explanation in **11.4.1 Controlling operation of watchdog timer**
	Change of **Caution 4 and 5** in **11.4.1 Controlling operation of watchdog timer**
	Change of **Caution 2** in **Table 11-3. Setting of Overflow Time of Watchdog Timer**
	Change of explanation in **11.4.3 Setting window open period of watchdog timer** Change of **Caution 2** in **Table 11-4. Setting Window Open Period of Watchdog Timer**
	Addition of **Note1, Caution1 and 2** in **Figure 12-2. Format of Clock Output Selection Register (CKS)**
	Change of explanation in **13.2 (2) Sample & hold circuit, (3) Series resistor string, (4) Voltage comparator and (5) Successive approximation register (SAR)**
	Change of explanation in **13.2 (8) Controller**
	Change of **Note 2** in **Figure 13-3. Format of A/D Converter Mode Register (ADM)**
	Change of **Table 13-1. Settings of ADCS and ADCE** **Figure 13-4. Timing Chart When Comparator Is Used** • Change of **Figure 13-4** and **Note**
	Change of **(1), (2)** and **Caution 1** and Addition of **Caution 4** in **Table 13-2. A/D Conversion Time Selection**
	Change of **Figure 13-8. Format of Analog Input Channel Specification Register (ADS)**
	Change of explanation and **Caution 1** in **13.3 (5) A/D port configuration register (ADPC)** Change of **Figure 13-9. Format of A/D Port Configuration Register (ADPC)**
	Addition of explanation in **13.3 (6) Port mode register 8 (PM8)** and **(7) Port mode register 9 (PM9)** Change of **Table 13-3. Setting Functions of P80/ANI0 to P87/ANI7, P90/ANI8 to P93/ANI11 Pins**
	Change of **13.4.1 Basic operations of A/D converter**
	Change of explanation in **13.4.3 (1) A/D conversion operation**
	Change of **13.6 Cautions for A/D Converter**
	Change of explanation and Addition of **Caution 3 to 5** in **14.1 (2) Asynchronous serial interface (UART) mode**
	Change of **Figure 14-1. LIN Transmission Operation**
	Figure 14-2. LIN Reception Operation • Change of **Figure 14-2** and explanation
	Addition of **Figure 14-4. Port Configuration for LIN Reception Operation (UART61)**
	Addition of **Caution 3** in **14.2 (3) Transmit buffer register 6n (TXB6n)**
	Change of **Note 1** in **Figure 14-7. Format of Asynchronous Serial Interface Operation Mode Register 60 (ASIM60) (1/2)**

(4/9)

Edition	Description
3rd	Addition of **Caution 4 and 5** in **Figure 14-7. Format of Asynchronous Serial Interface Operation Mode Register 60 (ASIM60) (2/2)**
	Change of **Note 1** in **Figure 14-8. Format of Asynchronous Serial Interface Operation Mode Register 61 (ASIM61) (1/2)**
	Addition of **Caution 4 and 5** in **Figure 14-8. Format of Asynchronous Serial Interface Operation Mode Register 61 (ASIM61) (2/2)**
	Change of bit 0 from INTSR6n to TXSF6n of **14.3 (3) Asynchronous serial interface transmission status register 6n (ASIF6n), Figure 14-11. Format of Asynchronous Serial Interface Transmission Status Register 60 (ASIF60) and Figure 14-12. Format of Asynchronous Serial Interface Transmission Status Register 61 (ASIF61)**
	Figure 14-15. Format of Baud Rate Generator Control Register 60 (BRGC60) • Change of **Figure 14-15 and Remark 2**
	Figure 14-16. Format of Baud Rate Generator Control Register 61 (BRGC61) • Change of **Figure 14-16 and Remark 2**
	Change of **Caution** in **14.3 (6) Asynchronous serial interface control register 6n (ASICL6n)**
	Change of **Caution 1, 2 and 4** and Addition of **Caution 6** in **Figure 14-17. Format of Asynchronous Serial Interface Control Register 60 (ASICL60) (2/2)**
	Change of **Caution 1, 2 and 4** and Addition of **Caution 6** in **Figure 14-18. Format of Asynchronous Serial Interface Control Register 61 (ASICL61) (2/2)**
	Change of **Caution** in **14.4.1 (1) Register used**
	Change of explanation in **14.4.2 (2) (c) Normal transmission**
	Change of bit 0 from INTSR6n to TXSF6n of **14.4.2 (d) Continuous transmission**
	Change of bit 0 from INTSR6n to TXSF6n of **Figure 14-24. Example of Continuous Transmission Processing Flow**
	Change of bit 0 from INTSR6n to TXSF6n of **Figure 14-25. Timing of Starting Continuous Transmission**
	Change of bit 0 from INTSR6n to TXSF6n of **Figure 14-26. Timing of Ending Continuous Transmission**
	Change of **Caution 1** in **14.4.2 (2) (e) Normal reception**
	Change of explanation in **14.4.2 (2) (h) SBF transmission**
	Change of explanation in **14.4.3 (2) (a) Baud rate**
	Table 14-4. Set Data of Baud Rate Generator • Change of **Table 14-4 and Remark**
	Change of **Table 14-5. Maximum/Minimum Permissible Baud Rate Error**
	Change of **Table 15-1. Configuration of Serial Interfaces CSI10 and CSI11** **Figure 15-1. Block Diagram of Serial Interface CSI10** • Change of **Figure 15-1** • Addition of **Remark**
	Figure 15-2. Block Diagram of Serial Interface CSI11 • Change of **Figure 15-2** • Addition of **Remark** Change of **Caution 2** in **(1) Transmit buffer register 1n (SOTB1n) and (2) Serial I/O shift register 1n (SIO1n)**
	Change of **Note 2 and 5** in **Figure 15-3. Format of Serial Operation Mode Register 10 (CSIM10)**
	Change of **Note 2 and 5** in **Figure 15-4. Format of Serial Operation Mode Register 11 (CSIM11)**
	Change of **Caution 2** in **Figure 15-5. Format of Serial Clock Selection Register 10 (CSIC10)**
	Change of **Caution 2** in **Figure 15-6. Format of Serial Clock Selection Register 11 (CSIC11)**
	Change of **note 1** in **15.4.1 (1) (a) • Serial operation mode register 10 (CSIM10) and • Serial operation mode register 11 (CSIM11)**
	Addition of **Remark 1** in **Figure 15-11. Timing of Clock/Data Phase**

(5/9)

Edition	Description
3rd	Change of **15.4.2 (3) Timing of output to SO1n pin (first bit)**
	Change of **15.4.2 (4) Output value of SO1n pin (last bit)**
	Change of **15.4.2 (5) SO1n output (see (a) in Figures 15-1 and 15-2)**
	Change of **Table 16-1. Overview of Functions**
	Change of **Table 16-11. Error Types**
	Change of **Table 16-13. Types of Error States**
	Change of explanation in **16.3.6 (4) (b) Error counter** Change of **Caution** in **16.3.6 (4) (c) Occurrence of bit error in intermission**
	Change of explanation in **16.3.6 (5) Recovery from bus-off state** Change of explanation and **Caution 2** and Addition of **Caution 1** in **16.3.6 (5) (a) Recovery operation from bus-off state through normal recovery sequence**
	Change of explanation in **16.3.7 (1) Prescaler**
	Addition of **Remark** in **Figure 16-18. Segment Setting and Figure 16-19. Reference: Configuration of Data Bit Time Defined by CAN Specification**
	Addition of **Caution** in **Table 16-17. Bit Configuration of CAN Global Registers** to **Table 16-19. Bit Configuration of Message Buffer Registers**
	Movement of **16.6 Bit Set/Clear Function**
	Change of **Caution** in EFSD bit in **16.7 (1) (a) Read** Addition of **Caution** in GOM bit in **16.7 (1) (b) write**
	Change of **Remark 4** in CCERC bit and **Remark 3** in VAILD bit in **16.7 (6) (a) Read**
	Addition of **Caution 2, 3** in PSMODE1 and PSMODE2 bits and **Caution** in OPMODE2-OPMODE0 bits in **16.7 (6) (a) Read**
	Addition of **Caution** in CINTS5 to CINTS0 bit in **16.7 (11) (b) Write**
	Addition of **Note** in ROVF bit in **16.7 (15) (a) Read**
	Addition of **Note** in TOVF bit in **16.7 (17) (a) Read**
	Addition of **Remark** in TSEN bit in **16.7 (18) (a) Read**
	Change of **Caution 2** in **16.7 (20) CAN message Data Length Register m (C0MDLCm)**
	Addition of **Caution 2** in ID28 to ID0 in **16.7 (22) CAN Message ID Register m (C0MIDLm, C0MIDHm)**
	Addition of **Caution** in TRQ bit and **Caution 2 and 3** in RDY bit in **16.7 (23) (a) Read** Addition of **Caution** in IE bit in **16.7 (23) (b) Write**
	Addition of **Caution** in RDY bit in **16.7 (23) (b) Write**
	Addition of **16.9.2 Receive Data Read**
	Addition of **Caution** in **16.9.3 Receive History List Function**
	Change of the explanation in **16.9.4 Mask Function**
	Change of **Caution** in **16.9.6 Remote Frame Reception**
	Change of the explanation in **16.10.1 Message Transmission**
	Change of **Remark 2** in **16.10.1 Message Transmission**
	Addition of **Caution** in **16.10.2 Transmit History List Function**
	Addition of **Remark** in **16.11.1 (1) Entering CAN sleep mode**
	Change of the explanation in **16.11.1 (2) Status in CAN sleep mode** Change of the explanation and addition of **Caution** in **16.11.1 (3) Releasing CAN sleep mode**
	Change of the explanation in **16.11.2 (2) Status in CAN stop mode**
	Addition of **16.13.4 Receipt/Transmit Operation in Each Operation Mode**
	Change of explanation in **Figure 16-35. Timing Diagram of Capture Signal TSOUT**

(6/9)

Edition	Description
3rd	Addition of **Note 2** in **Figure 16-40. Message Buffer Redefinition**
	Addition of **Remark** in **Figure 16-44. Transmission via Interrupt (Using C0LOPT Register)**
	Addition of **Remark** in **Figure 16-45. Transmission via Interrupt (Using C0TGPT Register)**
	Addition of **Remark** in **Figure 16-46. Transmission via Software Polling**
	Addition of **Note** in **Figure 16-47. Transmission Abort Processing (Except Normal Operation Mode with ABT)**
	Addition of **Note** in **Figure 16-48. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)**
	Change of **Figure 16-51. Reception via Interrupt (Using C0LIPT Register)** and addition of **Remark**
	Change of **Figure 16-52. Reception via Interrupt (Using C0RGPT Register)** and addition of **Remark**
	Change of **Figure 16-53. Reception via Software Polling** and addition of **Remark**
	Change of **Figure 16-54. Setting CAN Sleep Mode/Stop Mode**
	Change of **Figure 16-55. Clear CAN Sleep/Stop Mode**
	Addition of **Note, Caution** and **Remark** in **Figure 16-56. Bus-Off Recovery (Except Normal Operation Mode with ABT)**
	Addition of **Note, Caution** and **Remark** in **Figure 16-57. Bus-Off Recovery (Normal Operation Mode with ABT)**
	Change of **Figure 16-61. Setting CPU Standby (from CAN Sleep Mode)** and addition of **Caution**
	Change of **Figure 16-62. Setting CPU Standby (from CAN Stop Mode)**
	Change of explanation in **17.1 (1) Maskable interrupts** Change of explanation in **17.2 Interrupt Sources and Configuration**
	Addition of **Note 3** in **Table 17-1. Interrupt Source List (2/2)**
	Addition of **Note 3** in **Table 17-2. Flags Corresponding to Interrupt Request Sources**
	Change of **Caution 3** in **18.1.1 (2) STOP mode**
	Figure 18-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC) • Change of **Figure 18-1** and **Caution 2**
	Change of explanation and **Caution 3** in **18.1.2 (2) Oscillation stabilization time select register (OSTS)**
	Change of **Table 18-1. Operating Statuses in HALT Mode**
	Addition of **Note** in **Table 18-1. Operating Statuses in HALT Mode (2/2)**
	Change of **Figure 18-4. HALT Mode Release by Reset**
	Change and addition of **Note** in **Table 18-3. Operating Statuses in STOP Mode**
	Change of **Caution 4** in **Table 18-3. Operating Statuses in STOP Mode**
	Figure 18-5. Operation Timing When STOP Mode Is Released • Change of **Figure 18-5** • Addition of **Note1 and 2**
	Change of **Figure 18-6. STOP Mode Release by Interrupt Request Generation**
	Change of **18.2.2 (2) (b) Release by reset signal generation** and **Figure 18-7. STOP Mode Release by Reset**
	Change of explanation in **CHAPTER 19 RESET FUNCTION**
	Change of **Figure 19-2. Timing of Reset by RESET Input** and **Figure 19-3. Timing of Reset Due to Watchdog Timer Overflow**
	Change of **Figure 19-4. Timing of Reset in STOP Mode by RESET Input**
	Change of **Table 19-1. Operation Statuses During Reset Period**
	Addition of **Note 1** in **Table 19-2. Hardware Statuses After Reset Acknowledgment (2/3)**
	Addition of **Note 1** and change of **Note 2** in **Table 19-2. Hardware Statuses After Reset Acknowledgment (3/3)**
	Change of explanation in **21.1 Functions of Power-on-Clear Circuit**
	Change of **21.3 Operation of Power-on-Clear Circuit**

(7/9)

Edition	Description
3rd	**Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)** • Change of **Figure** and **Note 2, 4** • Addition of **Note 1, 3**
	Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2) • Change of **Figure** and **Note 2** • Addition of **Note 1** and **Caution 2**
	Change of **Figure 21-3. Example of Software Processing After Reset Release (1/2)**
	Change of explanation in **22.1 Functions of Low-Voltage Detector**
	Change of **Figure 22-1. Block Diagram of Low-Voltage Detector**
	Figure 22-2. Format of Low-Voltage Detection Register (LVIM) • Change of **Figure** and **Note 3**
	Change of **Caution 3** in **Figure 22-3. Format of Low-Voltage Detection Level Selection Register (LVIS)**
	Change of explanation and **Remark** in **22.4 Operation of Low-Voltage Detector**
	Change of explanation in **22.4.1 (1) When detecting level of supply voltage (V_{DD})**
	Change of explanation and **Note 2** in **22.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**
	Change of **Figure 22-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))**
	Change of explanation in **22.4.1 (1) When detecting level of supply voltage (V_{DD})**
	Figure 22-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) • Change of **Figure 22-7** and **Note 2**
	Change of explanation in **22.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**
	Figure 22-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI)) • Change of **Figure 22-8** and **Note 2**
	Change of explanation in **22.5 Cautions for Low-Voltage Detector**
	Change of **Figure 22-9. Example of Software Processing After Reset Release (1/2)**
	Change of **CHAPTER 23 OPTION BYTE**
	Addition of **Caution** in **24.1 Internal Memory Size Switching Register**
	Addition of **Caution** in **24.2 Internal Expansion RAM Size Switching Register**
	Change of **Note 2** in **Table 24-3. Wiring Between 78K0/FE2 and Dedicated Flash Memory Programmer**
	Change of **Figure 24-3. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode**
	Figure 24-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART60) Mode • Change of **Figure 24-4** • Addition of **Note**
	Change of explanation in **24.5 (1) CSI10** Change of **Figure 24-6. Communication with Dedicated Flash Memory Programmer (CSI10)**

(8/9)

Edition	Description
3rd	**Figure 24-7. Communication with Dedicated Flash Memory Programmer (UART60)** • Change of **Figure** and **Note** Change of explanation in **24.5 (2) UART60** **Table 24-4. Pin Connection** • Change of **Table** and **Note 1**
	Change of **Figure 24-8. FLMD0 Pin Connection Example**
	Change of explanation in **24.6.5 REGC pin** Change of explanation and **Caution 3** in **24.6.6 Other signal pins**
	Change of explanation in **24.6.7 Power supply**
	Change of **Note 1** in **Table 24-7. Communication Modes**
	Addition of **24.8 Security Settings**
	Addition of **24.9 Processing Time for Each Command When PG-FP4 Is Used (Reference)**
	Change of **Figure 24-16. Self-Programming Procedure**
	Addition of **Table 24-14. Processing Time and Interrupt Response Time for Self Programming Sample Library**
	Change of **24.11 Boot swap function**
	Change of **Caution** in **25.1 Outline of Functions**
	Addition of **Note** and **Caution** in **Figure 25-2. Connection Circuit Example (When QB-78K0MINI Is Not Used)** and **Figure 25-3. Connection Circuit Example (When Using QB-78K0MINI: X1 and X2 Are Used)**
	Addition of **Note** in **Figure 25-4. Connection Circuit Example (When Using QB-78K0MINI: Ports 31 and 32 Are Used)** Addition of **Figure 25-5. Connection of FLMD0 Pin for Self Programming by Means of On-Chip Debugging**
	Addition of **25.4 On-Chip Debug Security** Change of **25.5 Restrictions and Cautions on On-Chip Debug Function**
	27.1 Absolute Maximum Ratings • Addition of REGC pin input voltage of **Absolute Maximum Ratings** • Addition of I_{OH2} and I_{OH3} in **Output current, high of Absolute Maximum Ratings** • Addition of I_{OL2} and I_{OL3} in **Output current, low of Absolute Maximum Ratings**
	27.2 Oscillator Characteristics • Addition of RSTS = 1 and RSTS = 0 in the condition of **8 MHz internal oscillator** • Change of MAX. and MIN. value of **240 kHz internal oscillator** • Addition of **Remark** in **(2) On-chip Internal Oscillator Characteristics**
	27.3 DC Characteristics • Change of MAX. value of **Output current, high in 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V** • Addition of **Note 1 to 3** and change of **Remark** of **Output current, high and Output current, low** • Change of MAX. value of **Output current, high** • Addition of $V_{IH}4$ of **Input voltage, high** • Addition of I_{OL} = 5.0 mA, 2.0 mA in the condition of **Output voltage, low** • Change of MAX. value of **Output voltage, low (I_{OL} = 1.0 mA and V_{OL3}, I_{OL} = 5.0 mA, 3.0 mA, 1.0 mA)** • Change of **Supply current, A/D converter operating current, Watchdog timer operating current, LVI operating current**

(9/9)

Edition	Description
3rd	**27.4 (1) Basic operation** • Change of MIN. value of **External clock input high level width, low level width, External sub clock input high level width, low level width and TI000, TI001, TI002, TI003, TI010, TI011, TI012, TI013 input high-level width, low-level width** • Addition of **Peripheral hardware clock frequency** and **Note 1, 2** • Change of **TCY vs. V_{DD} (Main System Clock Operation)** • Change of **External clock input timing**
	Change of MAX. value of Transfer rate in (a) UART mode (UART6n, dedicated baud rate generator output) Change of MIN. value of SCK1n cycle time, SCK1n high-/low-level width, SI1n setup time (to SCK1n↑) in **27.4 (2) (b) 3-wire serial I/O mode (master mode, SCK1n... internal clock output)** Change of MAX. value of Delay time from SCK1n↓ to SO1n output in **27.4 (2) (c) 3-wire serial I/O mode (slave mode, SCK1n... external clock input)**
	27.4 (4) A/D Converter Characteristics • Addition of MAX. value of **Overall, Conversion time, Zero-scale error, Full-scale error, Integral non-linearity error, Differential non-linearity error**
	Change of MIN. value of Power supply rise time, Minimum pulse width in **27.4 (5) POC Circuit Characteristics**
	27.4 (6) LVI Circuit Characteristics • Addition of condition • Addition of MIN. and MAX. value of **External input pin** • Addition of **Detection voltage on application of supply voltage of Detection voltage** • Change of value of **Minimum pulse width** • Change of **LVI Circuit Timing**
	Change of value of Starting maximum time to VDD min (1.8 V) (V_{DD}: 0 V→1.8 V) and Starting maximum time to V_{DD} min (1.8 V) (pin RESET release→V_{DD}: 1.8 V) in **27.4 (7) Power Supply Starting Time**
	Change of **Note** in **27.5 Data Retention Characteristics**
	Change of **27.6 Flash EEPROM Programming Characteristics**
	Addition of **CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A2) GRADE PRODUCTS)**
	Change of 64-PIN PLASTIC LQFP(FINE PITCH)(10x10) and 64-PIN PLASTIC LQFP (12x12) in **CHAPTER 29 PACKAGE DRAWINGS**
	Addition of **CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS**
	Change of **Table 31-1. Registers That Generate Wait and Number of CPU Wait Clocks**
	Change of **31.3 Example of Wait Occurrence**
	Change of explanation in **Windows**
	Addition of **Note 1** in **Figure A-1. Development Tool Configuration**
	Addition of **(3) When using the on-chip debug emulator with programming function QB-MINI2** in **Figure A-1. Development Tool Configuration**
	Change of DF780893 Device file, **Note 1** and **Remark** in **A.2 Language Processing Software**
	Change of **A.4 Flash Memory Programming Tools**
	Change of explanation and **Note** in **A.5.1 When using in-circuit emulator QB-78K0FX2** Addition of **Remark** in **A.5.2 When using on-chip debug emulator QB-78K0MINI**
	Addition of **A.5.3 When using on-chip debug emulator with programming function QB-MINI2** Change of **A.6 Debugging Tools (Software)**
2nd	Modification of part number in **1. 3 Ordering Information**
	Addition of **Caution 2** in **2.2.3 P30 to P33 (port 3)**

Edition	Description
2nd	Addition of **Caution** in **2.2.10 P120 to P124 (port 12)**
	Addition of **Note** in **Table 2-4 Pin I/O Circuit Types (1/2)**
	Addition of **Note 3** in **Table 2-4 Pin I/O Circuit Types (2/2)**
	Addition of **Caution 2** in **4.2.3 Port 3**
	Addition of **Caution** in **4.2.10 Port 12**
	Modification of processing time in **Figure 5-12 Operation of the clock generating circuit when power supply voltage injection (When 1.59 V POC mode setup (option byte: LVISTART = 0))**
	Modification of processing time in **Figure 5-13 Operation of the clock generating circuit when power supply voltage injection (When 2.7 V/1.59V POC mode setup (option byte: LVISTART = 1))**
	Modification of address to FF8FH in **Figure 9-2 Format of Watch Timer Operation Mode Register (WTM)**
	Addition of **Caution** in **23.7.4 Port pins** Addition of **Caution 3** in **23.7.6 Other signal pins**
	Modification of standard setting in **Table 23-7 Communication Modes**
	Modification of **Note 4** in **26.3 DC characteristics**

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2511-2512, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

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