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

μPD78014H Subseries

8-bit Single-chip Microcontrollers

μPD78011H　　μPD78011H(A)
μPD78012H　　μPD78012H(A)
μPD78013H　　μPD78013H(A)
μPD78014H　　μPD78014H(A)
μPD78P018F　　μPD78P018F(A)

Document No. U12220EJ2V0UM00 (2nd edition)
Date Published February 1998 N CP (K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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 once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build 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 bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

License not needed	: μPD78P018FDW, 78P018FKK-S
The customer must judge the need for license	: μPD78011HCW-×××, 78011HGC-×××-AB8, 78011HGK-×××-8A8, μPD78011HCW(A)-×××, 78011HGC(A)-×××-AB8, 78011HGK(A)-×××-8A8 μPD78012HCW-×××, 78012HGC-×××-AB8, 78012HGK-×××-8A8, μPD78012HCW(A)-×××, 78012HGC(A)-×××-AB8, 78012HGK(A)-×××-8A8 μPD78013HCW-×××, 78013HGC-×××-AB8, 78013HGK-×××-8A8, μPD78013HCW(A)-×××, 78013HGC(A)-×××-AB8, 78013HGK(A)-×××-8A8 μPD78014HCW-×××, 78014HGC-×××-AB8, 78014HGK-×××-8A8, μPD78014HCW(A)-×××, 78014HGC(A)-×××-AB8, 78014HGK(A)-×××-8A8 μPD78P018FCW, 78P018FGC-AB8, 78P018FGK-8A8 μPD78P018FCW(A), 78P018FGC(A)-AB8

The application circuits and their parameters are for reference only and are not intended for use in actual design-ins.

M7 96.5

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Cumbica-Guarulhos-SP, Brasil
Tel: 011-6465-6810
Fax: 011-6465-6829

J97. 8

Major Revisions in This Edition

Page	Description
P.92 P.93 P.94	The following block diagrams were changed. • **Figure 4-6 Block Diagram of P20, P21, P23 to P26** • **Figure 4-7 Block Diagram of P22 and P27** • **Figure 4-8 Block Diagram of P30 to P37**
P.111	**Table 5-2 Relationship Between CPU Clock and Minimum Instruction Execution Table** was added.
P.168, 173	**Figures 7-10** and **7-13 Square Wave Output Operation Timing** were added.
P.226, 244	**Caution** indicating that serial interface channenl 0 operation stops in SBI mode was added.
P.227, 246	Condition to release (output ready signal) busy mode of serial interface channel 0 was added.
P.239	**Caution** on wiring was added to **13.4.3 (a) Bus release signal (REL), (b) Command signal (CMD)**.
P.411 to 424	**APPENDIX B DEVELOPMENT TOOLS** Entirely revised: Compatible with in-circuit emulator IE-78K0-NS and IE-78001-R-A.
P.425, 426	**APPENDIX C EMBEDDED SOFTWARE** Entirely revised: Fussy Interface Development Support System was deleted.
P.433	**APPENDIX E REVISION HISTORY** was added.

The mark ★ shows major revised points.

INTRODUCTION

Readers

This manual is intended for user engineers who understand the functions of the μPD78014H Subseries and wish to design and develop its application systems and programs.

Target products are as follows:

- μPD78014H Subseries: μPD78011H, 78012H, 78013H, 78014H
  μPD78011H(A), 78012H(A), 78013H(A), 78014H(A)

Purpose

This manual is designed to deepen your understanding of the following functions using the following organization.

Organization

Two manuals are available for the μPD78014H Subseries: this manual and Instruction Manual (common to the 78K/0 Series).

μPD78014H Subseries User's Manual (This manual)	78K/0 Series User's Manual Instruction
• Pin functions	• CPU function
• Internal block functions	• Instruction set
• Interrupt	• Instruction description
• Other internal peripheral functions	
• Outline of μPD78P018F[Note]	

Note The PROM version of the μPD78011H, 78012H, 78013H, and 78014H is the μPD78P018F, and that of the μPD78011H(A), 78012H(A), 78013H(A), and 78014H(A) is the μPD78P018F(A).

Caution Among the μPD78P018F products, the μPD78P018FDW and 78P018FKK-S do not have the reliability required for the mass production of systems. Use these models for experiment or function evaluation only.

How to Read This Manual

It is assumed that the readers of this manual have general knowledge on electric engineering, logic circuits, and microcontrollers.

☐ For the μPD78011H(A), 78012H(A), 78013H(A), 78014H(A), and 78P018F(A) users:

→ These devices are identical to the μPD78011H, 78012H, 78013H, 78014H, and 78P018F, respectively, except for their quality grade. The users of these versions can use this manual just by changing the part numbers as follows:

μPD78011H → μPD78011H(A)
μPD78012H → μPD78012H(A)
μPD78013H → μPD78013H(A)
μPD78014H → μPD78014H(A)
μPD78P018F → μPD78P018F(A)

☐ For the μPD78P018F users:

→ Check the differences between the μPD78014H Subseries and the μPD78P018F Subseries in **1.12 Differences between μPD78014H Subseries and μPD78P018F Subseries**. Also read **CHAPTER 19 OUTLINE OF μPD78P018F** to understand the specific functions of the μPD78P018F. Note that this manual does not explain in detail the specific functions of the μPD78P018F because these functions are not available in the μPD78014H Subseries.

☐ To understand the overall functions of the μPD78014H Subseries

→ Read this manual in the order of the **TABLE OF CONTENTS**.

☐ How to read register formats

→ The name of a bit whose number is encircled is reserved for the RA78K/0 and is defined for the CC78K/0 by the header file sfrbit.h.

☐ To learn the detailed functions of a register whose register name is known

→ Refer to **APPENDIX C REGISTER INDEX**.

☐ To learn the details of the instruction functions of the μPD78014H Subseries

→ Refer to **78K/0 Series User's Manual - Instruction (U12326E)** separately available.

☐ To check the electrical characteristics of the μPD78014H Subseries

→ Refer to the **Data Sheet** separately available.

☐ For application examples of the functions of the μPD78014H Subseries

→ Refer to the **Data Sheet** separately available.

Caution The application examples shown in this manual assume that the device is to be used in general electronic equipment and its quality grade is "Standard". If an application example in this manual is used for applications that require the "Standard" quality grade, check the quality grade of the parts to be connected and circuit configurations beforehand.

Legend

Data significance	:	Left: higher digit, right: lower digit
Active low	:	$\overline{\times\times\times}$ (top bar over pin or signal name)
Note	:	Footnote
Caution	:	Important information
Remark	:	Supplement
Numerical representation	:	Binary ... ×××× or ××××B
		Decimal ... ××××
		Hexadecimal ... ××××H

Related Documents

Some related documents listed below are preliminary editions but not so specified here.

- **Device-related documents**

Document Name	Document Number	
	English	Japanese
μPD78011H, 78012H, 78013H, 78014H Data Sheet	U11898E	U11898J
★ μPD78011H(A), 78012H(A), 78013H(A), 78014H(A) Data Sheet	U12174E	U12174J
μPD78P018F Data Sheet	U10955E	U10955J
μPD78P018F(A) Data Sheet	U12132E	U12132J
μPD78014H Subseries User's Manual	This manual	U12220J
78K/0 Series User's Manual - Instruction	U12326E	U12326J
78K/0 Series Instruction List	–	U10903J
78K/0 Series Instruction Set	–	U10904J
★ 78K/0 Series Application Note - Basic (I)	U12704E	U12704J

Caution The contents of the above related documents are subject to change without notice. Be sure to use the latest edition of a document for designing.

- **Development tool-related documents (user's manual)**

	Document Name		Document Number	
			English	Japanese
	RA78K0 Assembler Package	Operation	U11802E	U11802J
		Assembly Language	U11801E	U11801J
		Structured Assembly Language	U11789E	U11789J
	RA78K Series Structured Assembler Preprocessor		EEU-1402	U12323J
	CC78K0 C Compiler	Operation	U11517E	U11517J
		Language	U11518E	U11518J
	CC78K/0 C Compiler Application Note	Programming Know-how	EEA-1208	U13034J
	CC78K Series Library Source File		–	U12322J
	PG-1500 PROM Programmer		U11940E	U11940J
	PG-1500 Controller PC-9800 Series (MS-DOS™) Based		EEU-1291	EEU-704
	PG-1500 Controller IBM PC Series (PC DOS™) Based		U10540E	EEU-5008
★	IE-78K0-NS		To be prepared	To be prepared
★	IE-78001-R-A		To be prepared	To be prepared
★	IE-78018-NS-EM1		To be prepared	To be prepared
★	IE-78K0-R-EX1		To be prepared	To be prepared
	IE-78014-R-EM-A		U10418E	EEU-962
	EP-78240		U10332E	EEU-986
	EP-78012GK-R		EEU-1538	EEU-5012
	SM78K0 System Simulator Windows™ Based	Reference	U10181E	U10181J
	SM78K Series External Part User Open Interface Specifications		U10092E	U10092J
	ID78K0 Integrated Debugger EWS Based	Reference	–	U11151J
	ID78K0 Integrated Debugger PC Based	Reference	U11539E	U11539J
	ID78K0 Integrated Debugger Windows Based	Guidance	U11649E	U11649J
★	ID-78K0-NS Integrated Debugger PC Based	Reference	To be prepared	U12900J

Caution The contents of the above related documents are subject to change without notice. Be sure to use the latest edition of a document for designing.

- **Embedded software-related documents (user's manual)**

Document Name		Document Number	
		English	Japanese
78K/0 Series Real-time OS	Basics	U11537E	U11537J
	Installation	U11536E	U11536J
78K/0 Series OS MX78K0	Basics	U12257E	U12257J

- **Other related documents**

Document Name	Document Number	
	English	Japanese
IC Package Manual	C10943X	
Semiconductor Device Mounting Technology Manual	C10535E	C10535J
Quality Grades on NEC Semiconductor Devices	C11531E	C11531J
Reliability Quality Control on NEC Semiconductor Devices	C10983E	C10983J
★ Guide to Prevent Damage for Semiconductor Devices by Electro Static Discharge (ESD)	C11892E	C11892J
Guide to Quality Assurance for Semiconductor Devices	MEI-1202	–
Microcomputer Product Series Guide	–	U11416J

Caution The contents of the above related documents are subject to change without notice. Be sure to use the latest edition of a document for designing.

[MEMO]

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

[MEMO]

CHAPTER 1 GENERAL

1.1 Features

- Low EMI (Electro Magnetic Interferece) noise compared with the μPD78018F Subseries
- High-capacity ROM and RAM

Item / Part Number	Program Memory (ROM)	Data Memory		
		Internal High-Speed RAM	Internal Extended RAM	Internal Buffer RAM
μPD78011H	8 Kbytes	512 bytes	—	32 bytes
μPD78012H	16 Kbytes			
μPD78013H	24 Kbytes	1024 bytes		
μPD78014H	32 Kbytes			
μPD78P018F	60 Kbytes[Note 1]	1024 bytes[Note 2]	1024 bytes[Note 3]	

Notes 1. The capacity of the internal PROM can be changed by using memory size select register (IMS).

2. 512 or 1024 bytes is selectable by using IMS.

3. The capacity of the internal extended RAM can be changed by using internal extended RAM size select register (IXS).

- External memory extension space: 64 Kbytes
- Variable minimum instruction execution time - from high speed (0.4 μs: with 10.0-MHz main system clock) to ultra slow (122 μs: with 32.768-kHz subsystem clock)
- Instruction set suitable for system control
  - Bit processing in entire address space
  - Multiplication/division instructions
- I/O port: 53 lines (N-ch open-drain: 4 lines)
- 8-bit resolution A/D converter: 8 channels
  - Low-voltage operation (AV_{DD} = 1.8 to 5.5 V: operable in supply voltage range same as that of the CPU)
- Serial interface: 2 channels
  - 3-wire serial I/O / SBI / 2-wire serial I/O mode : 1 channel
  - 3-wire serial I/O mode (with automatic transmission/reception function) : 1 channel
- Timer: 5 channels
  - 16-bit timer/event counter : 1 channel
  - 8-bit timer/event counter : 2 channels
  - Watch timer : 1 channel
  - Watchdog timer : 1 channel
- Vectored interrupt source: 14
- Test input: 2 lines
- Two types of clock oscillation circuits (main system clock and subsystem clock)
- Supply voltage: V_{DD} = 1.8 to 5.5 V

1.2 Application Field

μPD78011H, 78012H, 78013H, 78014H:

Telephones, VCRs, audio sets, cameras, home appliances, etc.

μPD78011H(A), 78012H(A), 78013H(A), 78014H(A):

Electronic control units of automobile, gas detector/breakers, various safety equipment, etc.

1.3 Ordering Information

Part Number	Package	Internal ROM
μPD78011HCW-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78011HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78011HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78012HCW-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78012HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78012HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78013HCW-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78013HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78013HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78014HCW-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78014HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78014HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78011HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78011HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78011HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78012HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78012HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78012HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78013HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78013HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78013HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78014HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Mask ROM
μPD78014HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78014HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78P018FCW	64-pin plastic shrink DIP (750 mils)	One-time PROM
μPD78P018FDW	64-pin ceramic shrink DIP (with window) (750 mils)	EPROM
μPD78P018FGC-AB8	64-pin plastic QFP (14 × 14 mm)	One-time PROM
μPD78P018FGK-8A8	64-pin plastic LQFP (12 × 12 mm)	One-time PROM
μPD78P018FKK-S	64-pin ceramic WQFN (14 × 14 mm)	EPROM
μPD78P018FCW(A)	64-pin plastic shrink DIP (750 mils)	One-time PROM
μPD78P018FGC(A)-AB8	64-pin plastic QFP (14 × 14 mm)	One-time PROM

Remark ××× indicates ROM code suffix.

1.4 Quality Grade

Part Number	Package	Quality Grade
μPD78011HCW-×××	64-pin plastic shrink DIP (750 mils)	Standard
μPD78011HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78011HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78012HCW-×××	64-pin plastic shrink DIP (750 mils)	Standard
μPD78012HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78012HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78013HCW-×××	64-pin plastic shrink DIP (750 mils)	Standard
μPD78013HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78013HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78014HCW-×××	64-pin plastic shrink DIP (750 mils)	Standard
μPD78014HGC-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78014HGK-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78011HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Special
μPD78011HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78011HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Special
μPD78012HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Special
μPD78012HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78012HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Special
μPD78013HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Special
μPD78013HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78013HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Special
μPD78014HCW(A)-×××	64-pin plastic shrink DIP (750 mils)	Special
μPD78014HGC(A)-×××-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78014HGK(A)-×××-8A8	64-pin plastic LQFP (12 × 12 mm)	Special
μPD78P018FCW	64-pin plastic shrink DIP (750 mils)	Standard
μPD78P018FDW	64-pin ceramic shrink DIP (with window) (750 mils)	Not applicable
μPD78P018FGC-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78P018FGK-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78P018FKK-S	64-pin ceramic WQFN (14 × 14 mm)	Not applicable
μPD78P018FCW(A)	64-pin plastic shrink DIP (750 mils)	Special
μPD78P018FGC(A)-AB8	64-pin plastic QFP (14 × 14 mm)	Special

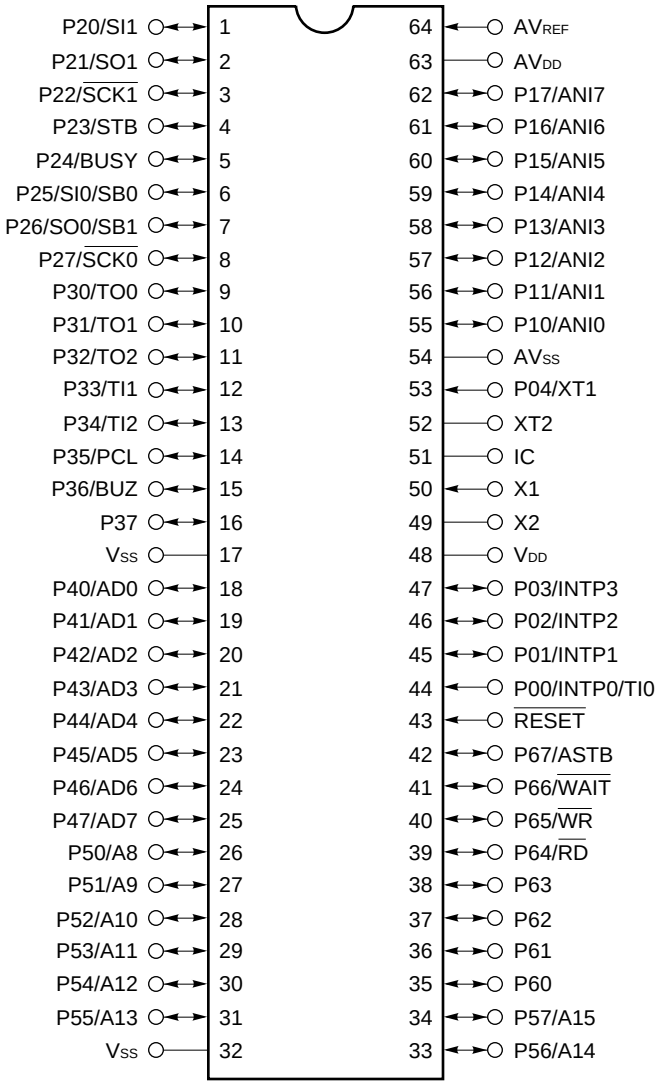
Caution Of the μPD78P018FDW, 78P018FKK-S, do not have a reliability intended for mass production of your systems. Use these models for experiment or function evaluation only.

Remark ××× indicates ROM code suffix.

Please refer to “Quality grades on NEC Semiconductor Devices” (Document number C11531E) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

1.5 Pin Configuration (Top View)

- **64-pin plastic shrink DIP (750 mils)**
  μPD78011HCW-×××, 78012HCW-×××, 78013HCW-×××, 78014HCW-×××
  μPD78011HCW(A)-×××, 78012HCW(A)-×××, 78013HCW(A)-×××, 78014HCW(A)-×××



Cautions 1. Directly connect the IC (Internally Connected) pins to V_{SS}.

2. The AV_{DD} pin serves as a power supply for both A/D converter and port circuitry. Connect this pin to a power supply unit different from but providing the same potential as the power supply to be connected to the V_{DD} pin, especially if the device is to be used in an application field where the EMI noise level is critical.

3. The AV_{SS} pin provides ground potential for both A/D converter and port circuitry. Connect this pin to a ground line different from but having the same potential as the ground line to be connected to the V_{SS} pin, especially if the device is to be used in an application field where the EMI noise level is critical.

- **64-pin plastic QFP (14 × 14 mm)**

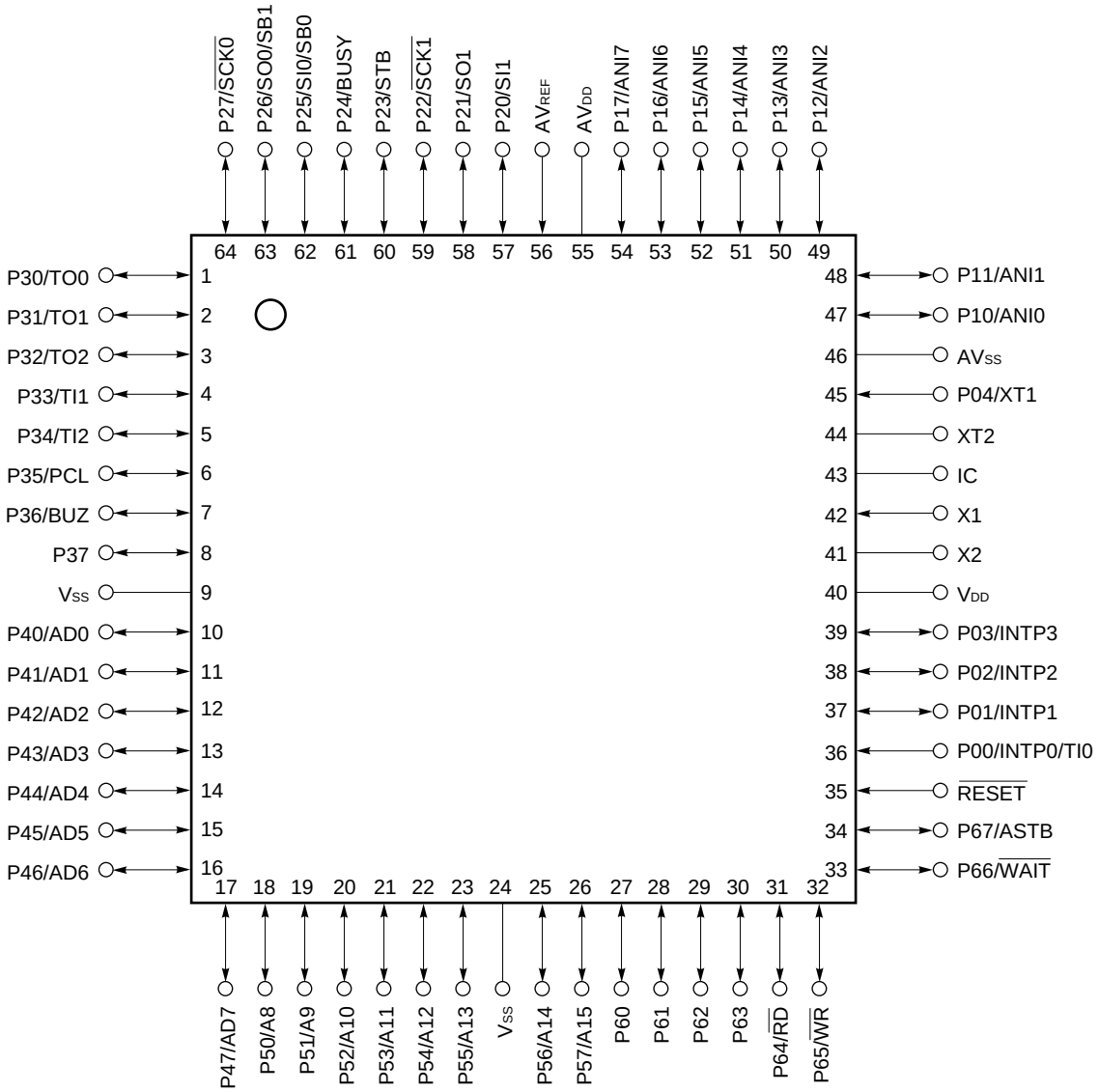
  μPD78011HGC-×××-AB8, 78012HGC-×××-AB8, 78013HGC-×××-AB8, 78014HGC-×××-AB8

  μPD78011HGC(A)-×××-AB8, 78012HGC(A)-×××-AB8, 78013HGC(A)-×××-AB8, 78014HGC(A)-×××-AB8

- **64-pin plastic LQFP (12 × 12 mm)**

  μPD78011HGK-×××-8A8, 78012HGK-×××-8A8, 78013HGK-×××-8A8, 78014HGK-×××-8A8

  μPD78011HGK(A)-×××-8A8, 78012HGK(A)-×××-8A8, 78013HGK(A)-×××-8A8, 78014HGK(A)-×××-8A8



Cautions 1. Directly connect the IC (Internally Connected) pins to V_{SS}.

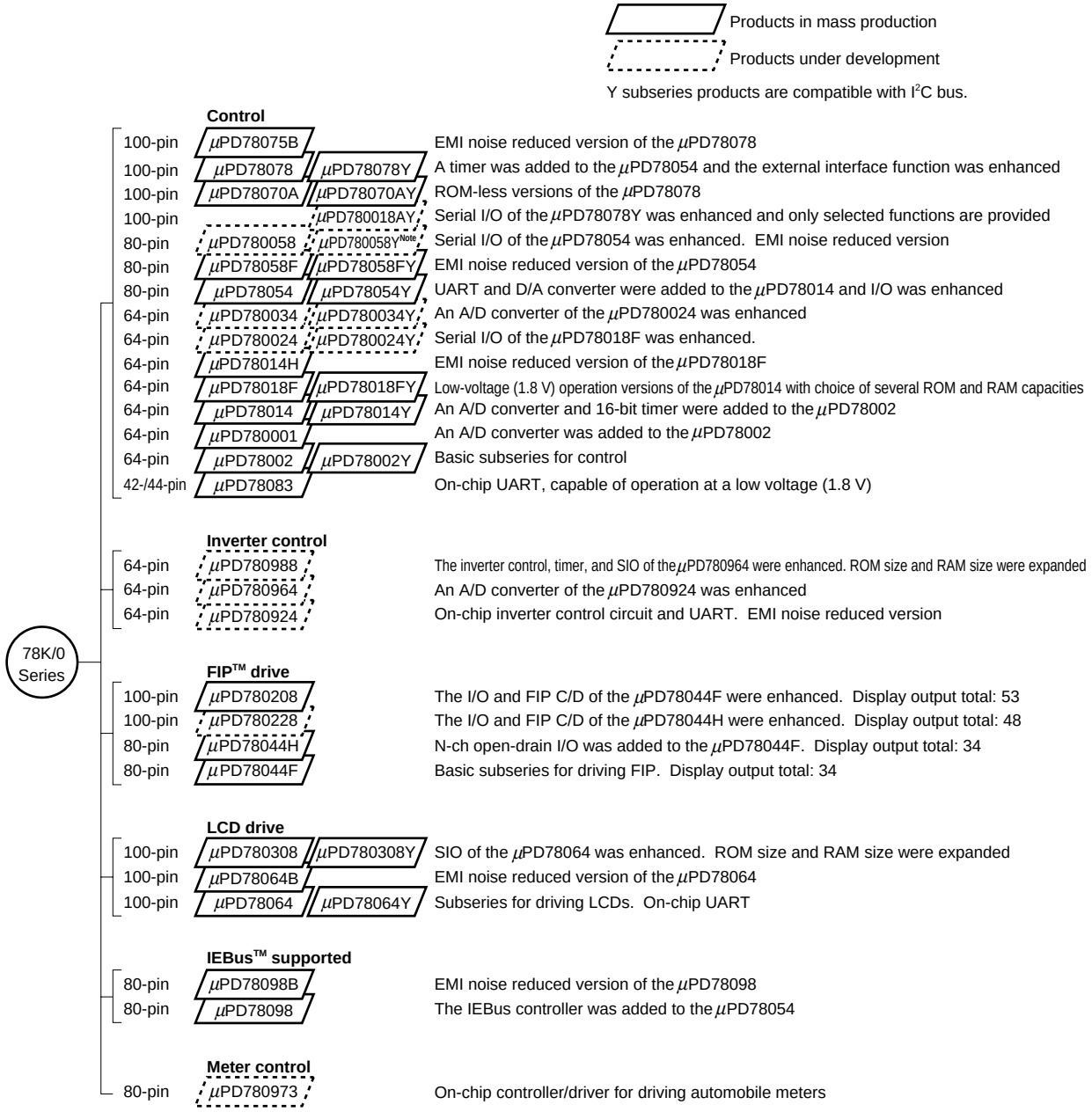
2. The AV_{DD} pin serves as a power supply for both A/D converter and port circuitry. Connect this pin to a power supply unit different from but providing the same potential as the power supply to be connected to the V_{DD} pin, especially if the device is to be used in an application field where the EMI noise level is critical.

3. The AV_{SS} pin provides ground potential for both A/D converter and port circuitry. Connect this pin to a ground line different from but having the same potential as the ground line to be connected to the V_{SS} pin, especially if the device is to be used in an application field where the EMI noise level is critical.

A8 to A15 : Address Bus
AD0 to AD7 : Address/Data Bus
ANI0 to ANI7 : Analog Input
ASTB : Address Strobe
AV_{DD} : Analog Power Supply
AV_{REF} : Analog Reference Voltage
AV_{SS} : Analog Ground
BUSY : Busy
BUZ : Buzzer Clock
IC : Internally Connected
INTP0 to INTP3 : Interrupt from Peripherals
P00 to P04 : Port0
P10 to P17 : Port1
P20 to P27 : Port2
P30 to P37 : Port3
P40 to P47 : Port4
P50 to P57 : Port5
P60 to P67 : Port6
PCL : Programmable Clock
$\overline{RD}$: Read Strobe
$\overline{RESET}$: Reset
SB0, SB1 : Serial Bus
$\overline{SCK0}$, $\overline{SCK1}$: Serial Clock
SI0, SI1 : Serial Input
SO0, SO1 : Serial Output
STB : Strobe
TI0 to TI2 : Timer Input
TO0 to TO2 : Timer Output
V_{DD} : Power Supply
V_{SS} : Ground
$\overline{WAIT}$: Wait
$\overline{WR}$: Write Strobe
X1, X2 : Crystal (Main System Clock)
XT1, XT2 : Crystal (Subsystem Clolck)

★ 1.6 78K/0 Series Expansion

The 78K/0 Series expansion is shown below. The names in frames are subseries.



Note Under planning

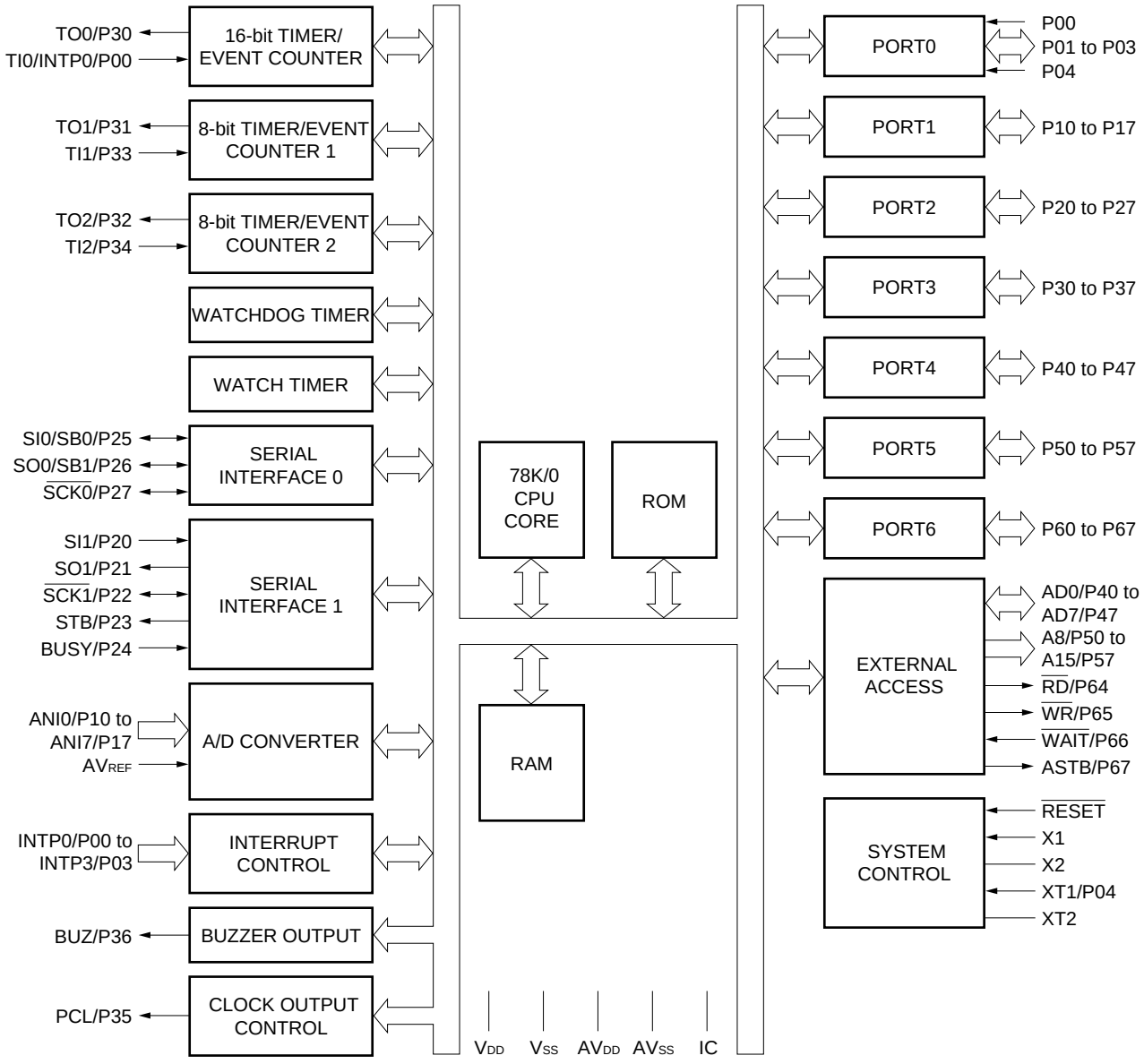
Major differences among those subseries are indicated below.

Subseries \ Function		ROM Capacity	Timer				8-bit A/D	10-bit A/D	8-bit D/A	Serial Interface	I/O	V_{DD} MIN. value	External extension
			8-bit	16-bit	Watch	WDT							
Control	μPD78075B	32 K to 40 K	4 ch	1 ch	1 ch	1 ch	8 ch	—	2 ch	3 ch (UART: 1 ch)	88	1.8 V	√
	μPD78078	48 K to 60 K											
	μPD78070A	—									61	2.7 V	
	μPD780058	24 K to 60 K	2 ch							3 ch (time-division UART: 1 ch)	68	1.8 V	
	μPD78058F	48 K to 60 K								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16 K to 60 K										2.0 V	
	μPD780034	8 K to 32 K					—	8 ch	—	3 ch (UART: 1 ch, time -division 3-wire: 1 ch)	51	1.8 V	
	μPD780024						8 ch	—					
	μPD78014H									2 ch	53		
	μPD78018F	8 K to 60 K											
	μPD78014	8 K to 32 K										2.7 V	
	μPD780001	8 K		—	—					1 ch	39		—
	μPD78002	8 K to 16 K			1 ch		—				53		√
	μPD78083				—		8 ch			1 ch (UART: 1 ch)	33	1.8 V	—
Inverter control	μPD780988	32 K to 60 K	3 ch	**Note 1**	—	1 ch	—	8 ch	—	3 ch (UART: 2 ch)	47	4.0 V	√
	μPD780964	8 K to 32 K		**Note 2**						2 ch (UART: 2 ch)		2.7 V	
	μPD780924						8 ch	—					
FIP drive	μPD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	—	—	2 ch	74	2.7 V	—
	μPD780228	48 K to 60 K	3 ch	—	—					1 ch	72	4.5 V	
	μPD78044H	32 K to 48 K	2 ch	1 ch	1 ch						68	2.7 V	
	μPD78044F	16 K to 40 K								2 ch			
LCD drive	μPD780308	48 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	—	—	3 ch (time-division UART: 1 ch)	57	2.0 V	—
	μPD78064B	32 K								2 ch (UART: 1 ch)			
	μPD78064	16 K to 32 K											
IEBus support	μPD78098B	40 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	—	2 ch	3 ch (UART: 1 ch)	69	2.7 V	√
	μPD78098	32 K to 60 K											
Meter control	μPD780973	24 K to 32 K	3 ch	1 ch	1 ch	1 ch	5 ch	—	—	2 ch (UART: 1 ch)	56	4.5 V	—

Notes 1. 16-bit timer: 2 channels
10-bit timer: 1 channel

2. 10-bit timer: 1 channel

1.7 Block Diagram



Remark The internal ROM and RAM capacities differ depending on the product.

1.8 Functional Outline

Item \ Part Number		μPD78011H	μPD78012H	μPD78013H	μPD78014H	μPD78P018F
Internal memory	ROM	Mask ROM				PROM
		8 Kbytes	16 Kbytes	24 Kbytes	32 Kbytes	60 Kbytes[Note 1]
	High-speed RAM	512 bytes		1024 bytes		1024 bytes[Note 1]
	Extended RAM	–				1024 bytes[Note 2]
	Buffer RAM	32 bytes				
Memory space		64 Kbytes				
General-purpose register		8 bits × 8 × 4 banks				
Minimum Instruction execution time	With main system clock	0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs (at 10.0 MHz)				
	With subsystem clock	122 μs (at 32.768 kHz)				
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD correction, etc.				
I/O port		• Total : 53 lines • CMOS input : 2 lines • CMOS I/O : 47 lines (Port lines to which internal pull-up resistor can be connected via software: 47 lines) • N-ch open drain I/O : 4 lines (15 V, pull-up resistor can be connected by mask option to mask ROM version only: 4 lines)				
A/D converter		• 8-bit resolution × 8 channels • Low-voltage operation: AV_{DD} = 1.8 to 5.5 V				
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O mode selectable : 1 channel • 3-wire serial I/O mode (with automatic transfer/reception function of up to 32 bytes) : 1 channel				
Timer		• 16-bit timer/event counter : 1 channel • 8-bit timer/event counter : 2 channels • Watch timer : 1 channel • Watchdog timer : 1 channel				
Timer output		3 lines (14-bit PWM output: 1 line)				
Clock output		39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz, 1.25 MHz (with main system clock: 10.0 MHz) 32.768 kHz (with subsystem clock: 32.768 kHz)				
Buzzer output		2.4 kHz, 4.9 kHz, 9.8 kHz (with main system clock: 10.0 MHz)				

Notes 1. The capacities of the internal PROM and internal high-speed RAM can be changed by using memory size select register (IMS).

2. The capacity of the internal extended RAM can be changed by using internal extended RAM size select register (IXS).

Item \ Part Number		μPD78011H	μPD78012H	μPD78013H	μPD78014H	μPD78P018F
Vectored interrupt source	Maskable	Internal: 8, external: 4				
	Non-maskable	Internal: 1				
	Software	1				
Test input		Internal: 1, external: 1				
Supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = –40 to +85 °C				
Package		• 64-pin plastic shrink DIP (750 mils) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm) • 64-pin ceramic shrink DIP (with window) (750 mils): μPD78P018F only • 64-pin ceramic WQFN (14 × 14 mm): μPD78P018F only				

1.9 Differences between Standard and Special Versions

Table 1-1. Differences between Standard and Special Versions

Item \ Part Number	Standard (μPD78011H, 78012H, 78013H, 78014H, 78P018F)	Special (μPD78011H(A), 78012H(A), 78013H(A), 78014H(A), 78P018F(A))
Quality grade	Standard	Special
Package	• 64-pin plastic shrink DIP (750 mils) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm) • 64-pin ceramic shrink DIP (with window) (750 mil: μPD78P018F only) • 64-pin ceramic WQFN (14 × 14 mm: μPD78P018F only)	• 64-pin plastic shrink DIP (750 mils) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm: μPD78011H(A), 78012H(A), 78013H(A), 78014H(A) only)

1.10 Mask Option

The mask ROM versions (μPD78011H, 78012H, 78013H, 78014H) have a mask option. By specifying the mask option when placing your order, the pull-up resistors shown in Table 1-2 can be connected. By using the mask option when a pull-up resistor is necessary, the number of components and the mounting area can be reduced.

Table 1-2 shows the mask option for the μPD78014H Subseries.

Table 1-2. Mask Option for Mask ROM Version

Pin Name	Mask Option
P60 to P63	Pull-up resistors can be connected in 1-bit units.

1.11 Differences between μPD78018F Subseries and μPD78014H Subseries

The μPD78014H Subseries is the low EMI version of the μPD78018F Subseries. The differences between the two subseries are shown in Table 1-3. The features not shown in the table are identical between the two subseries.

Table 1-3. Differences between μPD78018F Subseries and μPD78014H Subseries

Item \ Part Number	μPD78018F Subseries	μPD78014H Subseries
EMI noise prevention	Not provided	Provided
Internal extended RAM	1024 bytes	Not provided
ROM correction	Provided	Not provided

1.12 Differences between μPD78014H Subseries and μPD78P018F

Item \ Part Number		μPD78014H Subseries				μPD78P018F
		μPD78011H	μPD78012H	μPD78013H	μPD78014H	
EMI noise prevention		Provided				Not provided
Internal memory	ROM	Mask ROM				One-time PROM/EPROM
		8 Kbytes	16 Kbytes	24 Kbytes	32 Kbytes	60 Kbytes[Note 1]
	Extended RAM	Not provided				1024 bytes[Note 2]
	Buffer RAM	32 Kbytes				
Internally connected pin		Provided				Not provided
V_{PP} pin		Not provided				Provided
ROM correction		Not provided				Provided
Electrical specification		Refer to the Data Sheet of the respective part number.				

Notes 1. The capacity can be changed with the memory size select register (IMS).

2. The capacity can be changed with the internal extended RAM size select register (IXS).

2.1 List of Pin Functions

(1) Port pins (1/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P00	Input	Port 0. 5-bit I/O port.	Input only	Input	INTP0/TI0
P01	I/O		Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.	Input	INTP1
P02					INTP2
P03					INTP3
P04[Note 1]	Input		Input only	Input	XT1
P10 to P17	I/O	Port 1. 8-bit I/O port. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.[Note 2]		Input	ANI0 to ANI7
P20	I/O	Port 2. 8-bit I/O port. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.		Input	SI1
P21					SO1
P22					$\overline{SCK1}$
P23					STB
P24					BUSY
P25					SI0/SB0
P26					SO0/SB1
P27					$\overline{SCK0}$

Notes 1. To use the P04/XT1 pin as an input port line, set the bit 6 (FRC) of the processor clock control register (PCC) to 1 (do not use the internal feedback resistor of the subsystem clock oscillation circuit).

2. When using the P10/ANI0 through P17/ANI7 pins as the analog input lines of the A/D converter, set Port 1 to the input mode. The internal pull-up resistors are automatically disconnected.

(1) Port pins (2/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P30	I/O	Port 3. 8-bit I/O port. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.		Input	TO0
P31					TO1
P32					TO2
P33					TI1
P34					TI2
P35					PCL
P36					BUZ
P37					—
P40 to P47	I/O	Port 4. 8-bit I/O port. Can be specified for input/output in 8-bit units. When used as input port, internal pull-up resistor can be connected by software. Test input flag (KRIF) is set to 1 at falling edge of these pins.		Input	AD0 to AD7
P50 to P57	I/O	Port 5. 8-bit I/O port. Can directly drive LED. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.		Input	A8 to A15
P60	I/O	Port 6. 8-bit I/O port. Can be specified for input/output bitwise.	N-ch open-drain I/O port. Connection of internal pull-up resistor can be specified by mask option. Can directly drive LED.	Input	—
P61					
P62					
P63					
P64			When used as input port, internal pull-up resistor can be connected by software.		$\overline{RD}$
P65					$\overline{WR}$
P66					$\overline{WAIT}$
P67					ASTB

Caution For the port pins having alternate functions, the following operations are prohibited. If these prohibitions are not observed, the total error-free operation during A/D conversion cannot be guaranteed.

<1> Changing the value of the output latch of the given port when it is used as a port

<2> Changing the output level of the pin used as an output even if it is not used as a port

(2) Pins other than port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which valid edge can be specified (rising edge, falling edge, and both rising and falling edges)	Input	P00/TI0
INTP1				P01
INTP2				P02
INTP3		Falling edge detection external interrupt request input		P03
SI0	Input	Serial data input of serial interface	Input	P25/SB0
SI1				P20
SO0	Output	Serial data output of serial interface	Input	P26/SB1
SO1				P21
SB0	I/O	Serial data I/O of serial interface	Input	P25/SI0
SB1				P26/SO0
$\overline{SCK0}$	I/O	Serial clock I/O of serial interface	Input	P27
$\overline{SCK1}$				P22
STB	Output	Strobe output for serial interface automatic transmission/reception	Input	P23
BUSY	Input	Busy input for serial interface automatic transmission/reception	Input	P24
TI0	Input	External count clock input to 16-bit timer (TM0)	Input	P00/INTP0
TI1		External count clock input to 8-bit timer (TM1)		P33
TI2		External count clock input to 8-bit timer (TM2)		P34
TO0	Output	16-bit timer (TM0) output (shared by 14-bit PWM output)	Input	P30
TO1		8-bit timer (TM1) output		P31
TO2		8-bit timer (TM2) output		P32
PCL	Output	Clock output (for trimming of main system clock, subsystem clock)	Input	P35
BUZ	Output	Buzzer output	Input	P36
AD0 to AD7	I/O	Lower address/data bus when memory is externally extended	Input	P40 to P47
A8 to A15	Output	Higher address bus when memory is externally extended	Input	P50 to P57
$\overline{RD}$	Output	Strobe signal output for external memory read	Input	P64
$\overline{WR}$		Strobe signal output for external memory write		P65
$\overline{WAIT}$	Input	Wait insertion when external memory is accessed	Input	P66
ASTB	Output	Strobe output that externally latches address information output to ports 4 and 5 to access external memory	Input	P67

(2) Pins other than port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
ANI0 to ANI7	Input	Analog input to A/D converter	Input	P10 to P17
AV_{REF}	Input	Reference voltage input to A/D converter	—	—
AV_{DD}	—	Analog power to A/D converter (alternative to port power supply)	—	—
AV_{SS}	—	Ground of A/D converter (alternative to ground potential of port)	—	—
$\overline{RESET}$	Input	System reset input	—	—
X1	Input	Connect crystal resonator for main system clock oscillation	—	—
X2	—		—	—
XT1	Input	Connect crystal resonator for subsystem clock oscillation	Input	P04
XT2	—		—	—
V_{DD}	—	Positive power supply (except ports)	—	—
V_{SS}	—	Ground (except ports)	—	—
IC	—	Internally connected. Directly connect to V_{SS}	—	—

Cautions 1. The AV_{DD} pin serves as a power supply for both A/D converter and port circuitry. Connect this pin to a power supply unit different from but providing the same potential as the power supply to be connected to the V_{DD} pin, especially if the device is to be used in an application field where the EMI noise level is critical.

2. The AV_{SS} pin provides ground potential for both A/D converter and port circuitry. Connect this pin to a ground line different from but having the same potential as the ground line to be connected to the V_{SS} pin, especially if the device is to be used in an application field where the EMI noise level is critical.

2.2 Description of Pin Functions

2.2.1 P00 to P04 (Port0)

These pins constitute a 5-bit I/O port, port 0. In addition, these pins are also used to input external interrupt request signals, an external count clock to timer, a capture trigger signal, and to connect a crystal resonator for subsystem clock oscillation.

Port 0 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In the port mode, P00 and P04 function as input port lines, and P01 through P03 function as I/O port lines. P01 through P03 can be set in the input or output port mode in 1-bit units by using port mode register 0 (PM0). When these pins are used as an input port, an internal pull-up resistor can be used if so specified by the pull-up resistor option register (PUO).

(2) Control mode

In this mode, P00 through P04 are used to input external interrupt requests, an external count clock to timer, and to connect a crystal resonator for subsystem clock oscillation.

(a) INTP0 to INTP3

INTP0 through INTP2 are external interrupt request input pins for which valid edge can be specified (rising edge, falling edge, and both rising and falling edges). INTP0 also functions as the capture trigger input pin of the 16-bit timer/event counter when a valid edge is input. INTP3 is a falling edge-triggered external interrupt input pin.

(b) TI0

External count clock input pin of the 16-bit timer/event counter

(c) XT1

Subsystem clock oscillation crystal connecting pin

2.2.2 P10 to P17 (Port1)

These pins form an 8-bit I/O port, port1. These pins also serve as the analog input pins of the A/D converter. They can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P10 through P17 constitute an 8-bit I/O port which can be set in the input or output mode in 1-bit units by using the port mode register 1 (PM1). When used as an input port, an internal pull-up resistor can be used if so specified by the pull-up resistor option register (PUO).

(2) Control mode

In this mode, P10 through P17 function as the analog input pins (ANI0 to ANI7) of the A/D converter. When these pins are specified to serve as analog input pins, the internal pull-up resistor is automatically disconnected.

2.2.3 P20 to P27 (Port2)

These pins constitute an 8-bit I/O port, port 2. In addition, these pins are also used to input/output the data of the serial interface, input/output a clock signal, input a busy signal used for automatic transfer/reception, and output a strobe signal.

Port 2 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In the port mode, P20 and P27 function as an 8-bit I/O port. Port 2 can be set in the input or output mode in 1-bit units by using the port mode register 2 (PM2). When the port is used as an input port, an internal pull-up resistor can be used if so specified by the pull-up resistor option register (PUO).

(2) Control mode

In this mode, P20 through P27 input/output the data of the serial interface, input/output a clock, input a busy signal for automatic transfer/reception, and output a strobe signal.

(a) SI0, SI1, SO0, SO1

These are the serial data I/O pins of the serial interface.

(b) $\overline{\text{SCK0}}$, $\overline{\text{SCK1}}$

These are serial clock I/O pins of the serial interface.

(c) SB0, SB1

These are I/O pins of the NEC standard serial bus interface.

(d) BUSY

This pin inputs the busy signal for the automatic transmission/receive function of the serial interface.

(e) STB

This pin outputs a strobe signal for the automatic transmission/receive function of the serial interface.

Caution When using P20 through P27 as serial interface pins, the input/output mode and output latch must be set according to the functions to be used. For the details of the setting, refer to Figure 13-3. Format of Serial Operation Mode Register 0 and Figure 14-3. Format of Serial Operation Mode Register 1.

2.2.4 P30 to P37 (Port3)

These pins constitute an 8-bit I/O port, port 3. In addition, they also functions as timer I/O, clock output, and buzzer output pins.

Port 3 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, port 3 functions as an 8-bit I/O port which can be set in the input or output mode in 1-bit units by using the port mode register 3 (PM3). When used as an input port, an internal pull-up resistor can be used if so specified by the pull-up resistor option register (PUO).

(2) Control mode

In this mode, the pins of port 3 can be used as timer I/O, clock output, and buzzer output pins.

(a) TI1, TI2

These pins input an external count clock to the 8-bit timer/event counter.

(b) TO0 to TO2

Timer output pins.

(c) PCL

Clock output pin.

(d) BUZ

Buzzer output pin.

2.2.5 P40 to P47 (Port4)

These pins form an 8-bit I/O port, port 4. In addition, they also form an address/data bus.

When the falling edge of these pins is detected, the test input flag (KRIF) can be set to 1.

This port can be set in the following operation modes in 8-bit units.

(1) Port mode

In this mode, P40 through P47 function as an 8-bit I/O port which can be set in the input or output mode in 8-bit units by using the memory extension mode register (MM). When used as an input port, an internal pull-up resistor can be used if so specified by the pull-up resistor option register (PUO).

(2) Control mode

In this mode, P40 through P47 function as the lower address/data bus pins (AD0 to AD7) in the external memory extension mode. The pins used as address/data bus pins are automatically disconnected from the internal pull-up resistor.

2.2.6 P50 to P57 (Port5)

These pins form an 8-bit I/O port, port 5, which also serves as an address bus.

These pins can directly drive LEDs.

Port 5 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P50 through P57 constitute an 8-bit I/O port which can be set in the input or output mode in 1-bit units by using the port mode register 5 (PM5). When used as an input port, an internal pull-up resistor can be used if so specified by the pull-up resistor option register (PUO).

(2) Control mode

In this mode, P50 through P57 function as the higher address bus pins (A8 to A15) in the external memory extension mode. The pins used as address bus pins are automatically disconnected from the internal pull-up resistor.

2.2.7 P60 to P67 (Port6)

These pins constitute an 8-bit I/O port, port 6, which can be also used to output control signals in the external memory extension mode.

P60 through P63 can directly drive LEDs.

Port 6 can be set in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P60 through P67 constitute an 8-bit I/O port, which can be set in the input or output mode in 1-bit units by using the port mode register 6 (PM6).

P60 through P63 are N-ch open drain pins. These pins can be equipped with internal pull-up resistors by mask option.

When using P64 through P67 as input port pins, an internal pull-up resistor can be used if so specified by the pull-up resistor option register (PUO).

(2) Control mode

In this mode, P60 through P67 functions as control signal output pins ($\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, and ASTB) in the external memory extension mode. The pins used as control signal output pins are automatically disconnected from the internal pull-up resistor.

Caution If the external wait state is not used in the external memory extension mode, P66 can be used as an I/O port pin.

2.2.8 AV_{REF}

This pin inputs a reference voltage to the A/D converter.

Connect this pin to V_{SS} when the A/D converter is not used.

2.2.9 AV_{DD}

This is the analog power supply for the A/D converter and the power supply for port circuitries.

Keep this pin at the same voltage as the V_{DD} pin even when the A/D converter is not used.

2.2.10 AV_{SS}

This is the ground pin for the A/D converter and the power supply for port circuitries.

Keep this pin at the same voltage as the V_{SS} pin even when the A/D converter is not used.

2.2.11 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.2.12 X1 and X2

These pins are used to connect a crystal resonator for main system clock oscillation.

To supply an external clock, input the clock to X1 and input the inverted signal to X2.

2.2.13 XT1 and XT2

These pins are used to connect a crystal resonator for subsystem clock oscillation.

To supply an external clock, input the clock to XT1 and input the inverted signal to XT2.

2.2.14 V_{DD}

Positive power supply pin (except port)

2.2.15 V_{SS}

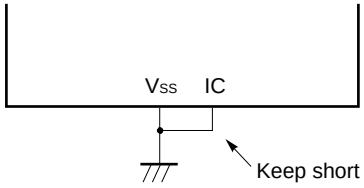
Ground pin (except port)

2.2.16 IC

The IC (Internally Connected) pin is used to set the μPD78011H, 78012H, 78013H, and 78014H in the test mode before shipment. In the normal operation mode, directly connect this pin to the V_{SS} pin with as short a wiring length as possible.

If the wiring length between the IC pin and V_{SS} pin is too long, or if a potential difference is generated between the IC pin and V_{SS} pin because an external noise is superimposed on the IC pin, user's program may not run correctly.

• Directly connect the IC pin to the V_{SS} pin.



2.3 I/O Circuits of Pins and Handling of Unused Pins

Table 2-1 shows the I/O circuit type of each pin and how to handle unused pins.
For the configuration of each type of I/O circuit, refer to **Figure 2-1**.

Table 2-1. I/O Circuit Type of Each Pin

Pin Name	I/O Circuit Type	I/O	Recommended Connection when Unused
P00/INTP0/TI0	2	Input	Connect to V_{SS}.
P01/INTP1	8-D	I/O	Independently connect to V_{SS} via a resistor.
P02/INTP2			
P03/INTP3			
P04/XT1	16	Input	Connect to V_{DD}.
P10/ANI0 to P17/ANI7	11-C	I/O	Independently connect to V_{DD} or V_{SS} via a resistor.
P20/SI1	8-D		
P21/SO1	5-J		
P22/$\overline{SCK1}$	8-D		
P23/STB	5-J		
P24/BUSY	8-D		
P25/SI0/SB0	10-C		
P26/SO0/SB1			
P27/$\overline{SCK0}$			
P30/TO0	5-J		
P31/TO1			
P32/TO2			
P33/TI1	8-D		
P34/TI2			
P35/PCL	5-J		
P36/BUZ			
P37			
P40/AD0 to P47/AD7	5-O	I/O	Independently connect to V_{DD} via a resistor.
P50/A8 to P57/A15	5-J	I/O	Independently connect to V_{DD} or V_{SS} via a resistor.
P60 to P63	13-I	I/O	Independently connect to V_{DD} via a resistor.
P64/$\overline{RD}$	15-J		Independently connect to V_{DD} or V_{SS} via a resistor.
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			
$\overline{RESET}$	2	Input	—
XT2	16	—	Leave open.
AV_{REF}	—		Connect to V_{SS}.
AV_{DD}			Connect to a power supply different from but providing the same potential as V_{DD}.
AV_{SS}			Connect to a ground line different from but having the same potential as V_{SS}.
IC			Directly connect to V_{SS}.

Figure 2-1. I/O Circuits of Pins (1/2)

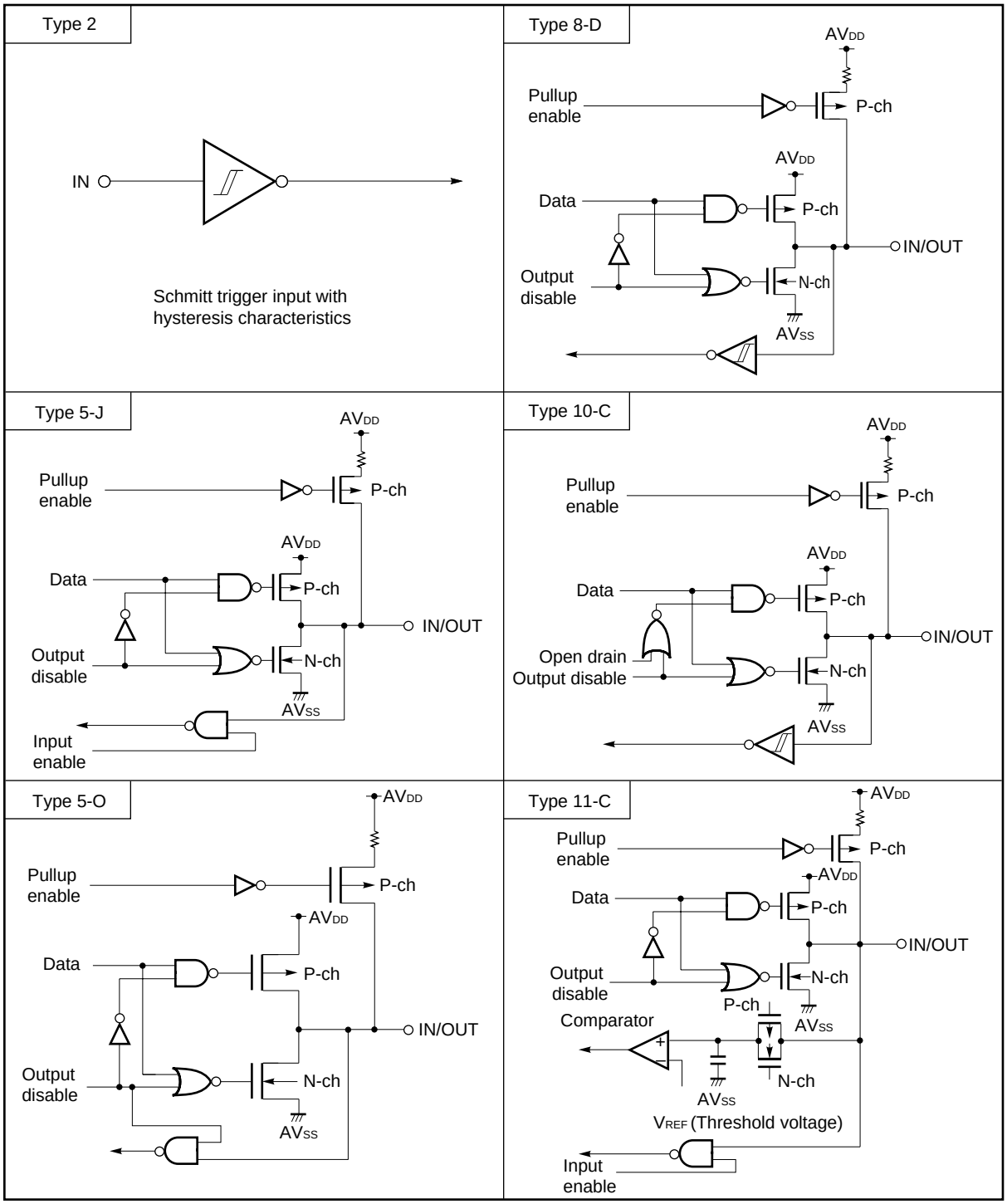
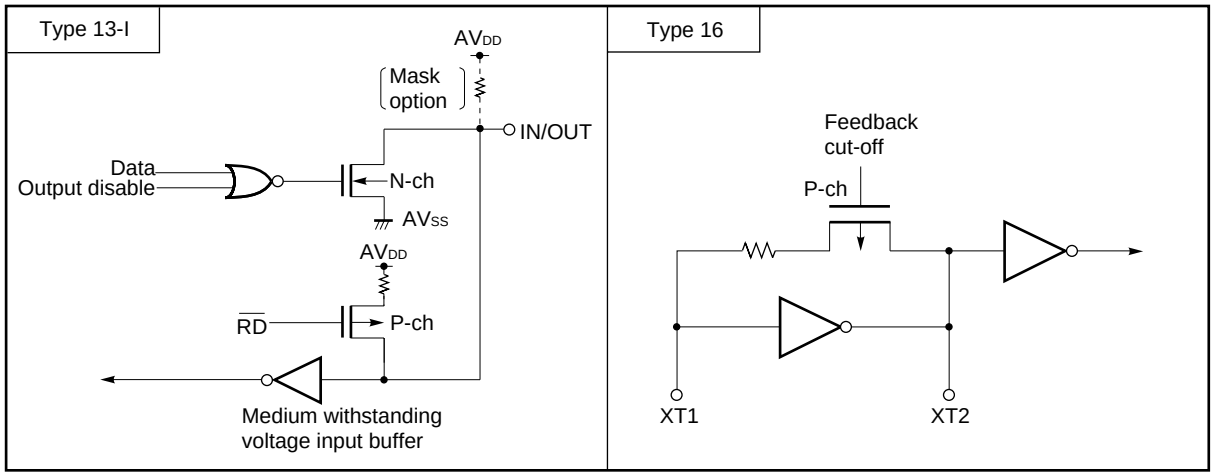


Figure 2-1. I/O Circuits of Pins (2/2)



3.1 Memory Space

Each model in the μPD78014H Subseries can access a memory space of 64 Kbytes. Figures 3-1 through 3-4 show memory maps of the respective models.

Figure 3-1. Memory Map (μPD78011H)

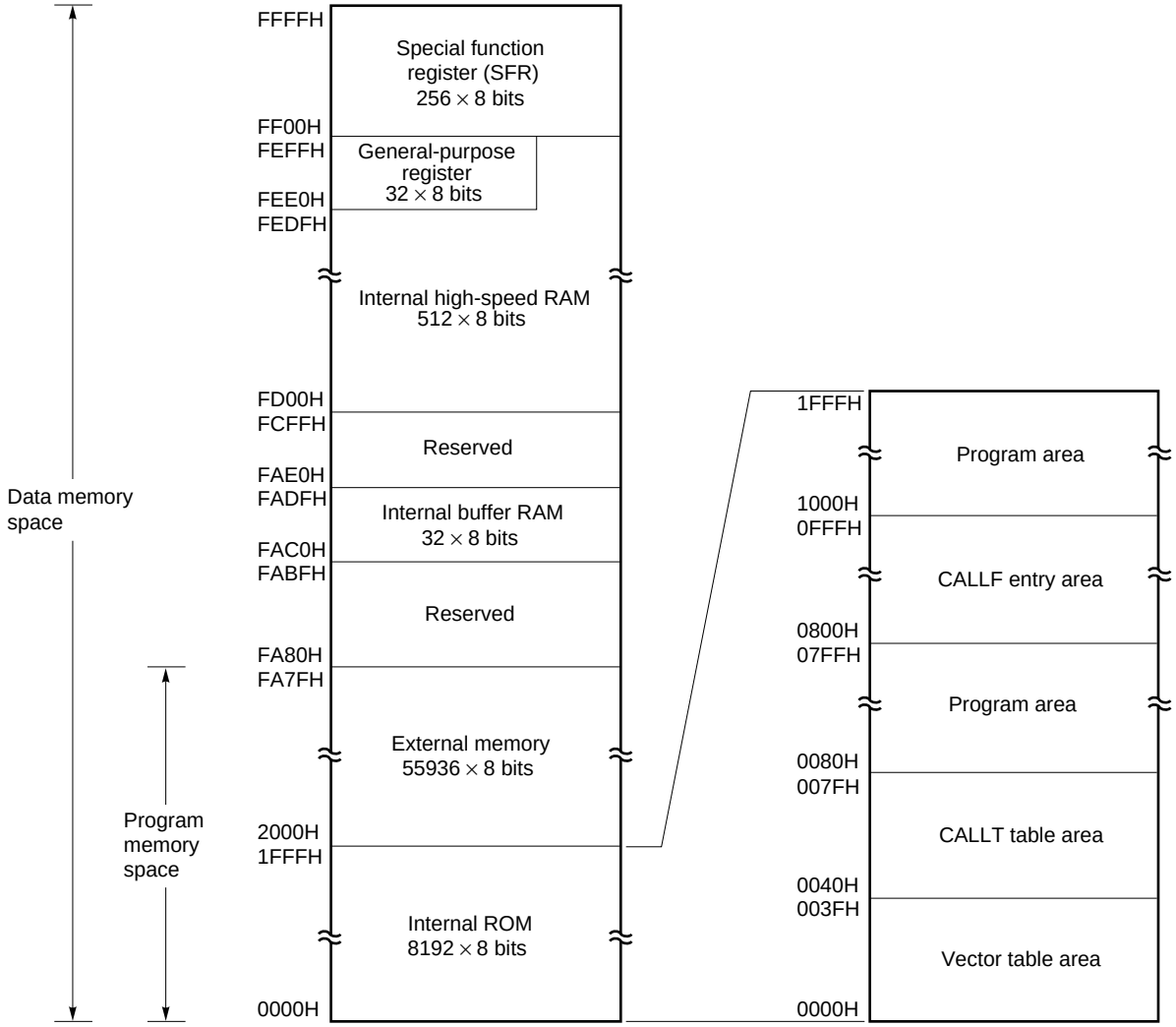


Figure 3-2. Memory Map (μPD78012H)

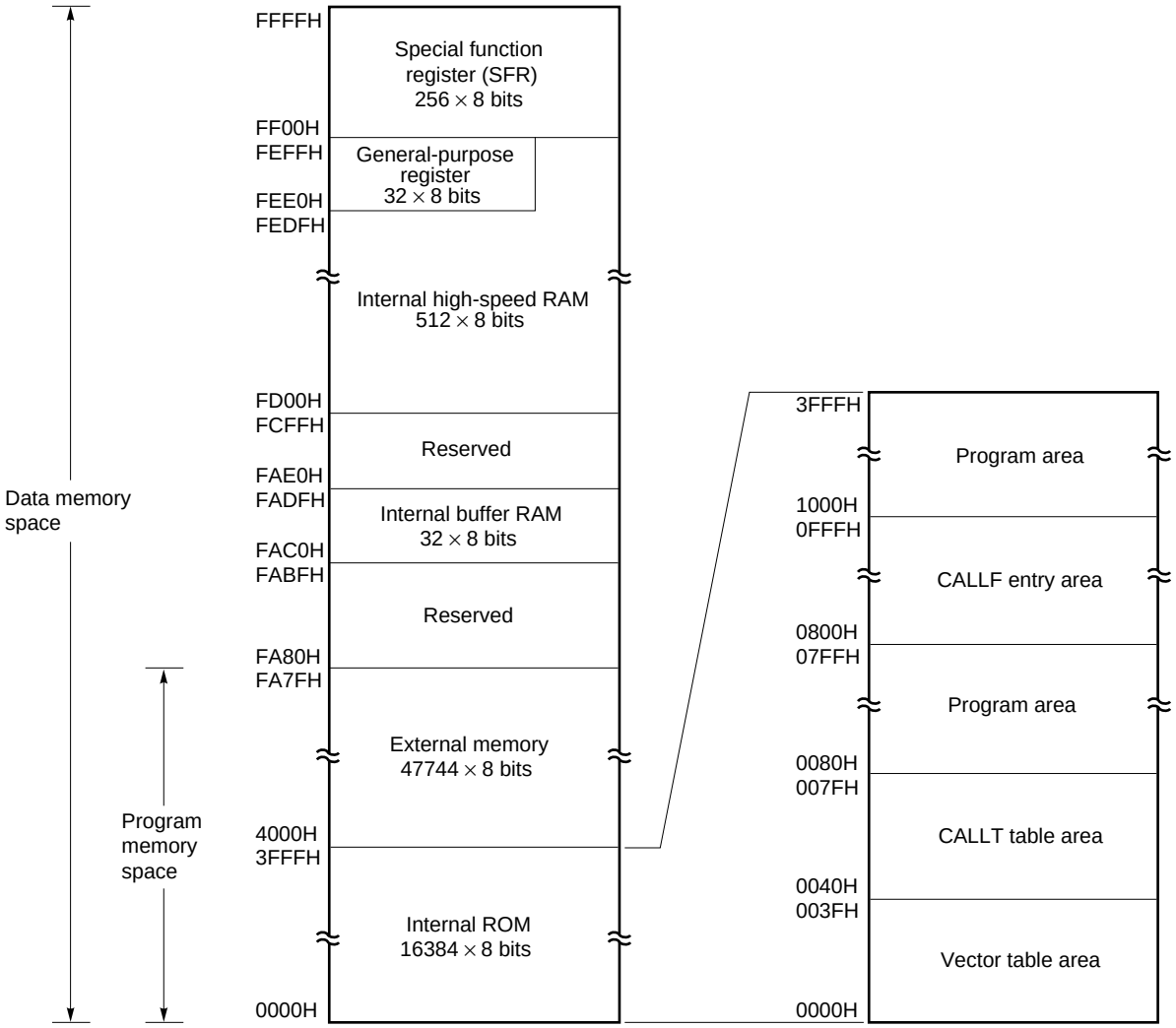


Figure 3-3. Memory Map (μPD78013H)

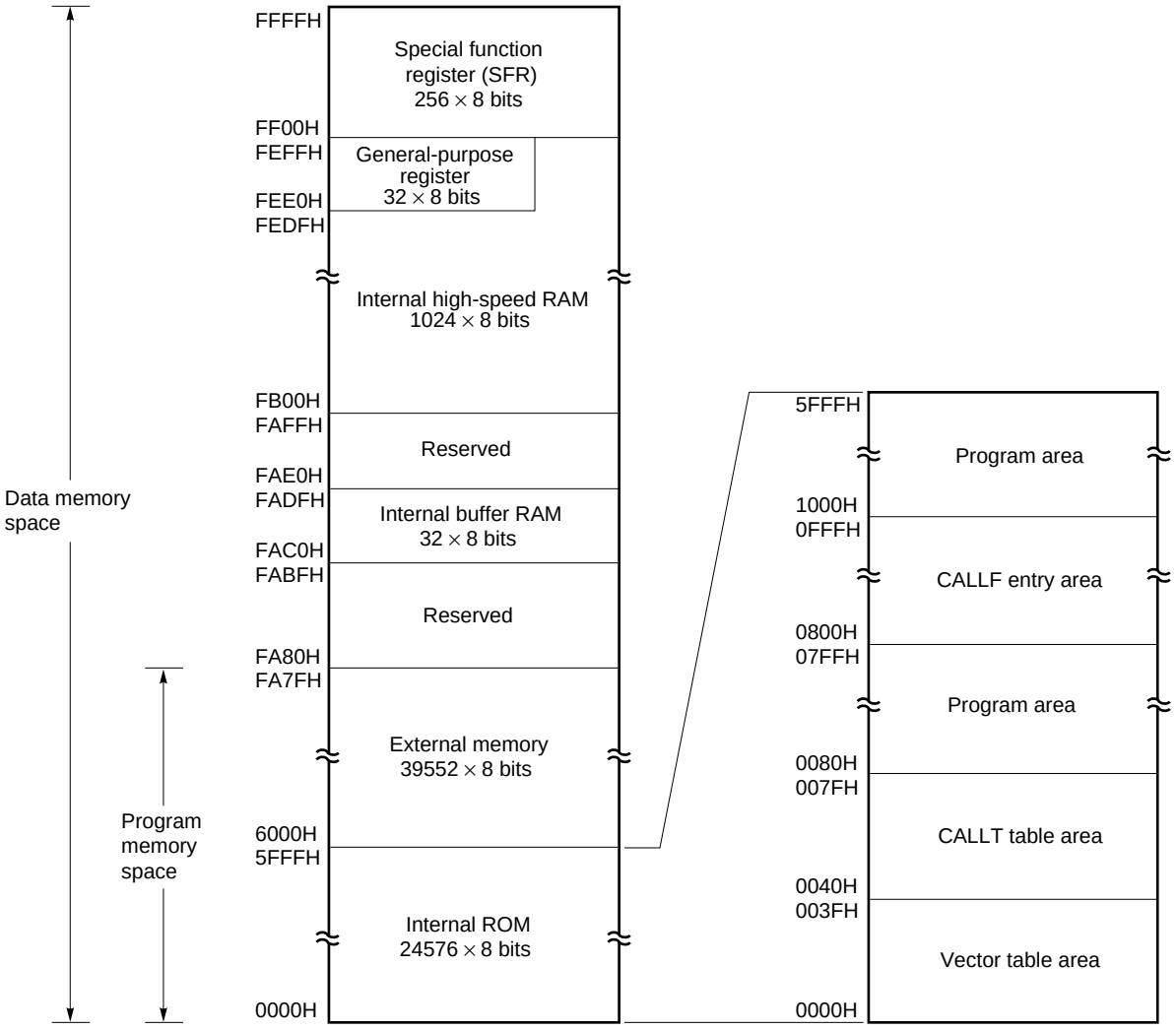
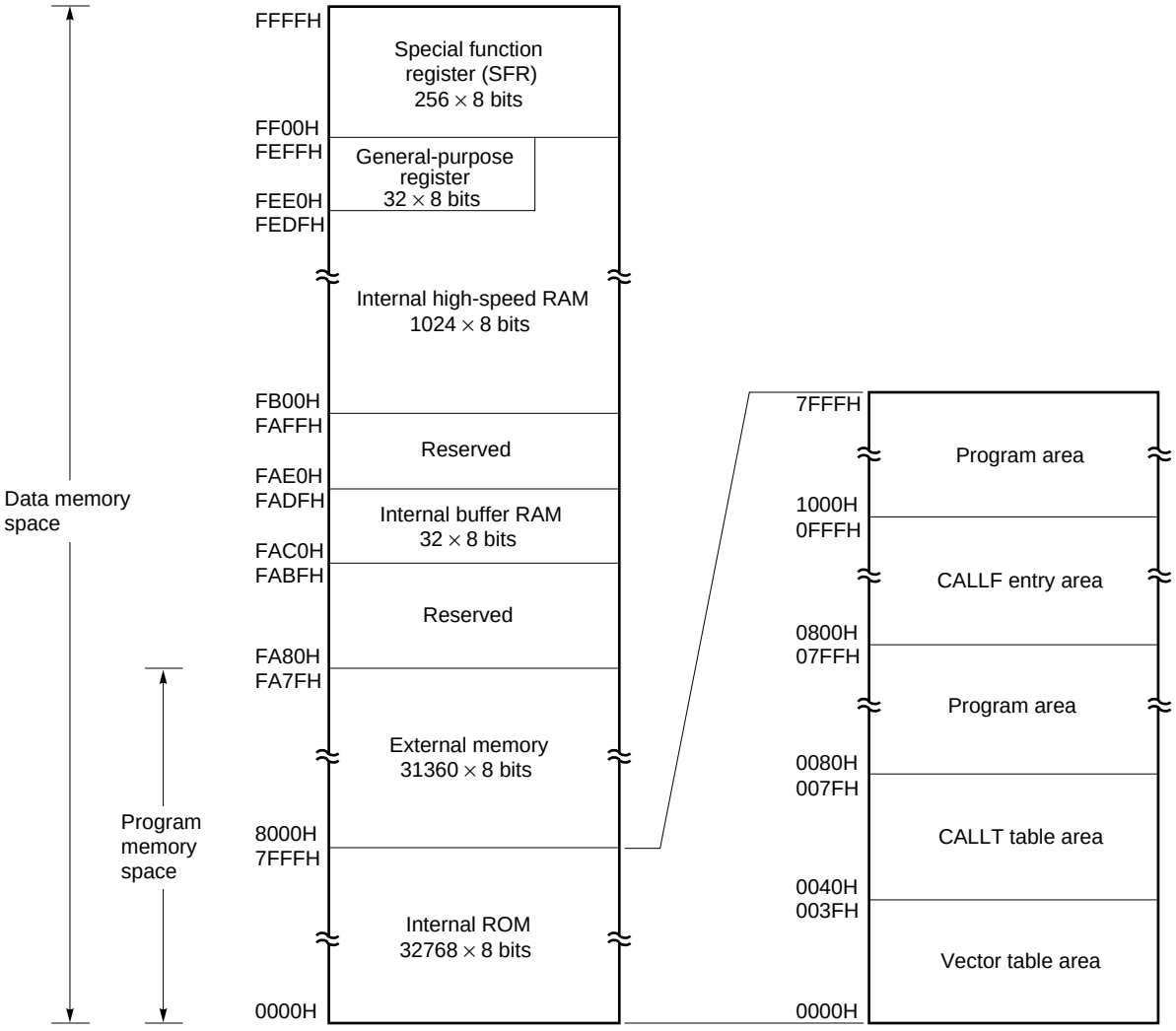


Figure 3-4. Memory Map (μPD78014H)



3.1.1 Internal program memory space

The internal program memory space stores programs and table data. This space is usually addressed by the program counter (PC).

Each model in the μPD78014H Subseries is provided with the following internal ROM:

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78011H	Mask ROM	8192 × 8 bits (0000H to 1FFFH)
μPD78012H		16384 × 8 bits (0000H to 3FFFH)
μPD78013H		24576 × 8 bits (0000H to 5FFFH)
μPD78014H		32768 × 8 bits (0000H to 7FFFH)

The following areas are allocated to the internal program memory space:

(1) Vector table area

A 64-byte area of addresses 0000H to 003FH is reserved as a vector table area. This area stores program start addresses to which execution branches when the $\overline{\text{RESET}}$ signal is input or when an interrupt request is generated. Of a 16-bit program start address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input	0010H	INTCSI1
0004H	INTWDT	0012H	INTTM3
0006H	INTP0	0014H	INTTM0
0008H	INTP1	0016H	INTTM1
000AH	INTP2	0018H	INTTM2
000CH	INTP3	001AH	INTAD
000EH	INTCSI0	003EH	BRK instruction

(2) CALLT instruction table area

In a 64-byte area of addresses 0040H to 007FH, the subroutine entry address of a 1-byte call instruction (CALLT) can be stored.

(3) CALLF instruction entry area

From an area of addresses 0800H to 0FFFH, a subroutine can be directly called by using a 2-byte call instruction (CALLF).

3.1.2 Internal data memory space

The μPD78014H Subseries has the following RAMs:

(1) Internal high-speed RAM

The μPD78014H Subseries is provided with the following internal high-speed RAM.

Table 3-3. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD78011H	512 × 8 bits (FD00H to FEFFH)
μPD78012H	
μPD78013H	1024 × 8 bits (FB00H to FEFFH)
μPD78014H	

A 32-byte area of addresses FEE0H to FEFFH is assigned four banks of general registers. Each bank consists of eight 8-bit registers.

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

(2) Internal buffer RAM

To a 32-byte area of addresses FAC0H to FADFH, an internal buffer RAM is allocated. The internal buffer RAM is used to save the transmit/receive data of serial interface channel 1 (3-wire serial I/O mode with automatic transfer function). If not used in the 3-wire serial I/O mode with automatic transfer function, the internal buffer RAM can also be used as an ordinary RAM.

3.1.3 Special function register (SFR) area

Special function registers (SFRs) of on-chip peripheral hardware are allocated to an area of FF00H to FFFFH (refer to **Table 3-5**).

Caution Do not access an address to which no SFR is allocated.

3.1.4 External memory space

This is an external memory space that can be accessed by using the memory extension mode register (MM) setting. This space can store programs and table data, and can be assigned peripheral devices.

3.2 Processor Registers

The μPD78014H Subseries is provided with the following processor registers:

3.2.1 Control registers

Each of these registers has a dedicated function such as to control the program sequence, status, and stack memory. The control registers include the program counter (PC), program status word (PSW), and stack pointer (SP).

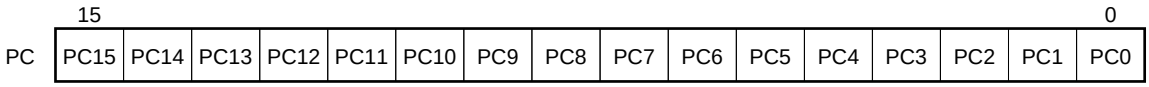
(1) Program counter (PC)

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

The contents of this register are automatically incremented according to the number of bytes of an instruction to be fetched when a normal operation is performed. When a branch instruction is executed, immediate data or the contents of a register is set to the program counter.

When the $\overline{\text{RESET}}$ signal is input, the value of the reset vector table at addresses 0000H and 0001H is set to the program counter.

Figure 3-5. Program Counter Configuration



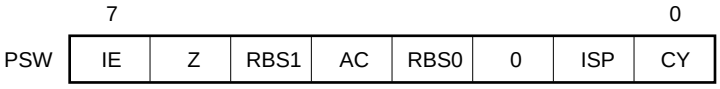
(2) Program status word (PSW)

The program status word is an 8-bit register consisting of flags that are set or reset as a result of instruction execution.

The contents of the program status word are automatically pushed to the stack when an interrupt request is generated or when the PUSH PSW instruction is executed, and are automatically popped from the stack when the RETB, RETI, or POP PSW instruction is executed.

The contents of the program status word are set to 02H when the $\overline{\text{RESET}}$ signal is input.

Figure 3-6. Program Status Word Configuration



(a) Interrupt enable flag (IE)

This flag controls acknowledgement of an interrupt request by the CPU.

When this flag is reset to 0, the device enters DI (disable interrupt) status and does not acknowledge any interrupt requests other than non-maskable ones.

When this flag is set to 1, the device enters EI (enable interrupt) status and acknowledges interrupts if there are. Acknowledgement of interrupt requests is controlled by the in-service priority flag (ISP), interrupt mask flag corresponding to each interrupt source, and priority flag.

The IE flag is reset to 0 when the DI instruction is executed or when an interrupt request is acknowledged, and is set to 1 when the EI instruction is executed.

(b) Zero flag (Z)

This flag is set to 1 when the result of an operation performed is zero; otherwise, it is reset to 0.

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

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

Information of 2 bits that indicate the register bank selected by execution of the “SEL RBn” instruction is stored in these flags.

(d) Auxiliary carry flag (AC)

This flag is set to 1 when a carry occurs from bit 3 or a borrow to bit 3 occurs as a result of an operation performed; otherwise, it is reset to 0.

(e) In-service priority flag (ISP)

This flag controls the priority of maskable vectored interrupts that can be acknowledged.

When this flag is 0, acknowledgement of the vectored interrupt requests for which the lower priority is specified by the priority specification flag register (PR0L, PR0H) is disabled (refer to **15.3 (3) Priority specification flag register**).

Whether an interrupt request is actually acknowledged or not depends on the interrupt enable flag (IE) content.

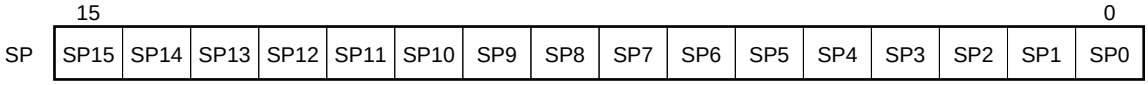
(f) Carry flag (CY)

This flag records an overflow or underflow that occurs as the result of executing an add or subtract instruction. It also records the value shifted out when a rotate instruction is executed. In addition, it also functions as a bit accumulator when a bit operation instruction is executed.

(3) Stack pointer (SP)

This is a 16-bit register that holds the first address of the stack area in the memory. As the stack area, only the internal high-speed RAM area can be specified (FD00H to FEFFH for μPD78011H and 78012H while FB00H to FEFFH for μPD78013H and 78014H).

Figure 3-7. Stack Pointer Configuration



The contents of the stack pointer are decremented when data is written (saved) to the stack memory, and incremented when data are read (restored) from the stack memory.

The data saved/restored as a result of each stack operation are as shown in Figures 3-8 and 3-9.

Caution The contents of the SP become undefined when the $\overline{\text{RESET}}$ signal is input. Be sure to initialize the SP before executing an instruction.

Figure 3-8. Data Saved to Stack Memory

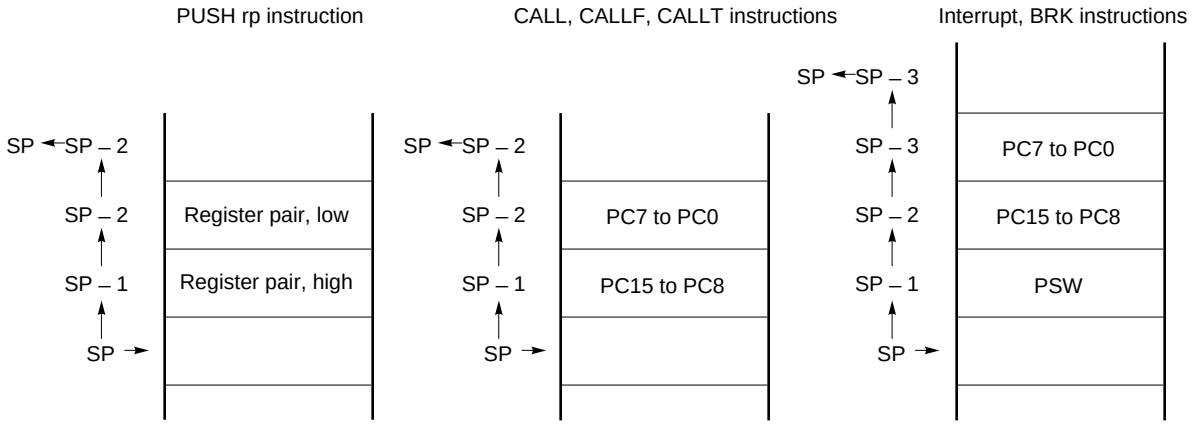
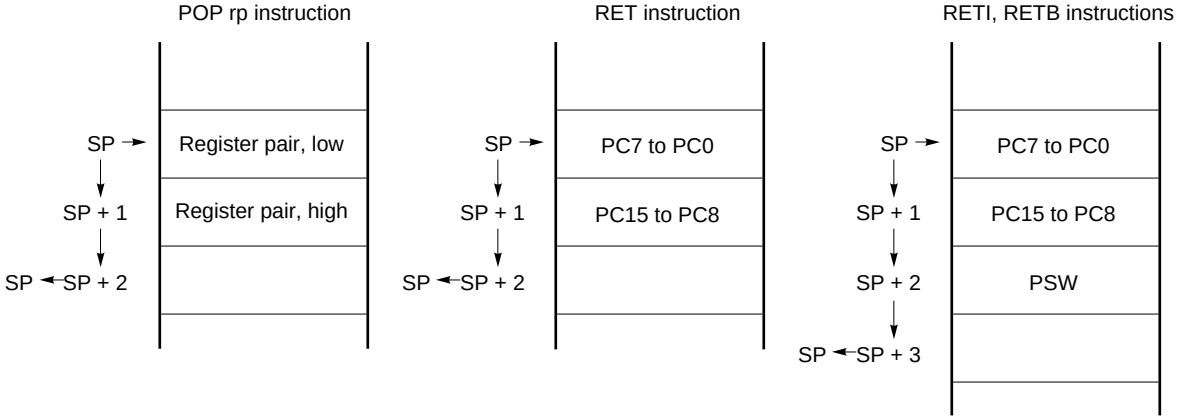


Figure 3-9. Data Restored from Stack Memory



3.2.2 General-purpose registers

General-purpose registers are mapped to the specific addresses of the data memory (FEE0H to FEFFH). Four banks of general-purpose registers, each consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H) are available.

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

Each register can be described not only in function name (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) but also in absolute name (R0 to R7, RP0 to RP3).

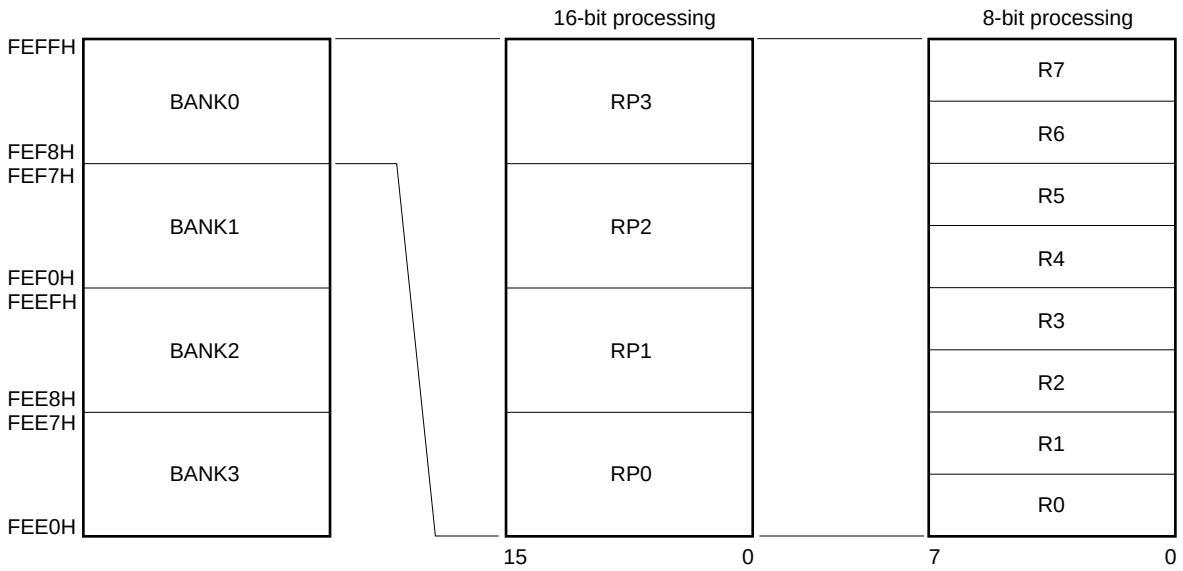
The register bank used for instruction execution is set by the CPU control instruction (SEL RBn). Because four register banks are provided, an efficient program can be developed by using one register bank for ordinary processing and another bank for interrupt request.

Table 3-4. Absolute Addresses of General-Purpose Registers

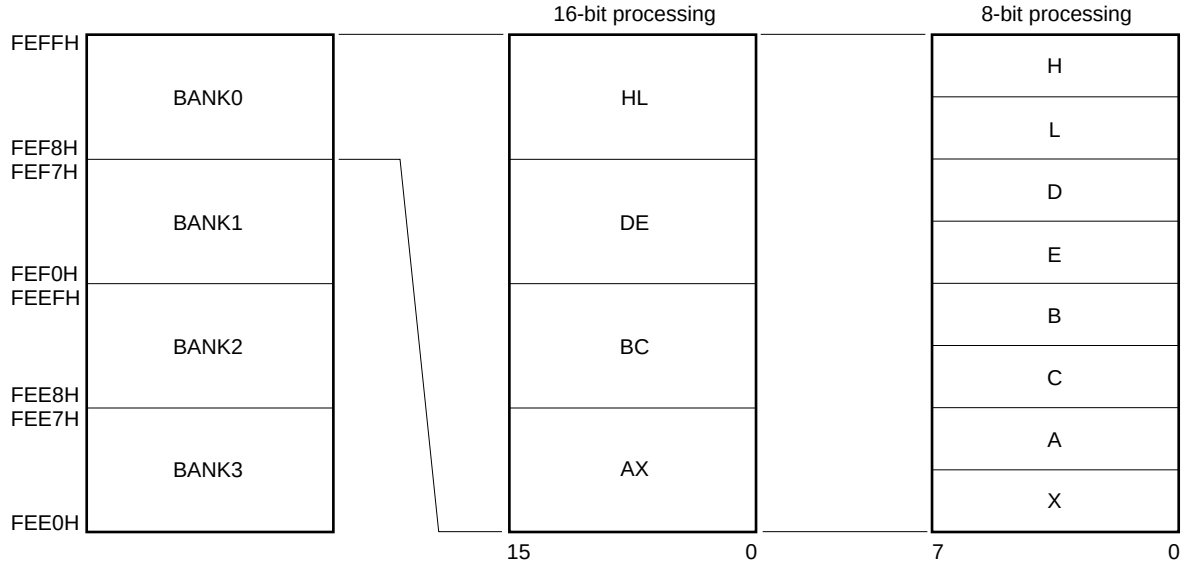
Bank Name	Register		Absolute Address	Bank Name	Register		Absolute Address
	Function Name	Absolute Name			Function Name	Absolute Name	
BANK0	H	R7	FEFFH	BANK2	H	R7	FEEFH
	L	R6	FEFEH		L	R6	FEEEH
	D	R5	FEFDH		D	R5	FEEDH
	E	R4	FEFCH		E	R4	FEECH
	B	R3	FEFBH		B	R3	FEEBH
	C	R2	FEFAH		C	R2	FEEAH
	A	R1	FEF9H		A	R1	FEE9H
	X	R0	FEF8H		X	R0	FEE8H
BANK1	H	R7	FEF7H	BANK3	H	R7	FEE7H
	L	R6	FEF6H		L	R6	FEE6H
	D	R5	FEF5H		D	R5	FEE5H
	E	R4	FEF4H		E	R4	FEE4H
	B	R3	FEF3H		B	R3	FEE3H
	C	R2	FEF2H		C	R2	FEE2H
	A	R1	FEF1H		A	R1	FEE1H
	X	R0	FEF0H		X	R0	FEE0H

Figure 3-10. General-Purpose Register Configuration

(a) Absolute name



(b) Function name



3.2.3 Special function registers (SFRs)

Unlike the general-purpose registers, special function registers have their own functions and are allocated to an area of addresses FF00H to FFFFH.

The special function registers can also be manipulated in the same manner as the general-purpose registers by using operation, transfer, and bit manipulation instructions. The bit units in which one register is to be manipulated (1, 8, or 16 bits) differ from that of another register.

The bit unit for manipulation is specified as follows:

- **1-bit manipulation**
  A symbol reserved by the assembler is described as the operand (sfr.bit) of a 1-bit manipulation instruction. An address can also be specified.
- **8-bit manipulation**
  A symbol reserved by the assembler is described as the operand (sfr) of an 8-bit manipulation instruction. An address can also be specified.
- **16-bit manipulation**
  A symbol reserved by the assembler is described as the operand (sfrp) of a 16-bit manipulation instruction. To specify address, describe an even address.

Table 3-5 lists the special function register. The meanings of the symbols in this table are as follows:

- **1-bit manipulation**
- **Symbol**
  These symbols indicate the addresses of the special function registers.
  They are reserved words for the RA78K/0 and defined by header file sfrbit.h for the CC78K/0. These symbols
★ can be described as the operands of instructions when the RA78K/0, ID78K0, ID78K0-NS, or SM78K0 is used.
- **R/W**
  Indicates whether the special function register in question can be read or written.
  R/W : Read/write
  R : Read only
  W : Write only
- **Bit units for manipulation**
  Circles in this column indicate the bit units (1, 8, 16) in which the special function register in question can be manipulated. Dashes indicate impossible manipulation bit units.
- **After reset**
  Indicates the status of the special function register when the $\overline{\text{RESET}}$ signal is input.

Table 3-5. Special Function Register List (1/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Access Unit			After Reset
				1 bit	8 bits	16 bits	
FF00H	Port 0	P0	R/W	○	○	–	00H
FF01H	Port 1	P1		○	○	–	
FF02H	Port 2	P2		○	○	–	
FF03H	Port 3	P3		○	○	–	
FF04H	Port 4	P4		○	○	–	Undefined
FF05H	Port 5	P5		○	○	–	
FF06H	Port 6	P6		○	○	–	
FF10H FF11H	16-bit compare register	CR00		–	–	○	
FF12H FF13H	16-bit capture register	CR01	R	–	–	○	
FF14H FF15H	16-bit timer register	TM0		–	–	○	0000H
FF16H	8-bit compare register	CR10	R/W	–	○	–	Undefined
FF17H	8-bit compare register	CR20		–	○	–	
FF18H	8-bit timer register 1	TMS TM1	R	–	○	○	00H
FF19H	8-bit timer register 2	TM2		–	○		
FF1AH	Serial I/O shift register 0	SIO0	R/W	–	○	–	Undefined
FF1BH	Serial I/O shift register 1	SIO1		–	○	–	
FF1FH	A/D conversion result register	ADCR	R	–	○	–	
FF20H	Port mode register 0	PM0	R/W	○	○	–	1FH
FF21H	Port mode register 1	PM1		○	○	–	FFH
FF22H	Port mode register 2	PM2		○	○	–	
FF23H	Port mode register 3	PM3		○	○	–	
FF25H	Port mode register 5	PM5		○	○	–	
FF26H	Port mode register 6	PM6		○	○	–	
FF40H	Timer clock select register 0	TCL0		○	○	–	00H
FF41H	Timer clock select register 1	TCL1		–	○	–	
FF42H	Timer clock select register 2	TCL2		–	○	–	
FF43H	Timer clock select register 3	TCL3		–	○	–	88H
FF47H	Sampling clock select register	SCS		–	○	–	00H
FF48H	16-bit timer mode control register	TMC0		○	○	–	
FF49H	8-bit timer mode control register	TMC1		○	○	–	
FF4AH	Watch timer mode control register	TMC2		○	○	–	
FF4EH	16-bit timer output control register	TOC0		○	○	–	

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Access Unit			After Reset
					1 bit	8 bits	16 bits	
FF4FH	8-bit timer output control register	TOC1		R/W	○	○	–	00H
FF60H	Serial operation mode register 0	CSIM0			○	○	–	
FF61H	Serial bus interface control register	SBIC			○	○	–	
FF62H	Slave address register	SVA			–	○	–	Undefined
FF63H	Interrupt timing specification register	SINT			○	○	–	00H
FF68H	Serial operation mode register 1	CSIM1			○	○	–	
FF69H	Automatic data transmission/reception control register	ADTC			○	○	–	
FF6AH	Automatic data transmission/reception address pointer	ADTP			–	○	–	
FF6BH	Automatic transmission/reception interval specification register	ADTI			○	○	–	
FF80H	A/D converter mode register	ADM			○	○	–	01H
FF84H	A/D converter input select register	ADIS			–	○	–	00H
FFD0H to FFDFH	External access area[Note 1]				○	○	–	Undefined
FFE0H	Interrupt request flag register 0L	IF0	IF0L		○	○	○	00H
FFE1H	Interrupt request flag register 0H		IF0H		○	○		
FFE4H	Interrupt mask flag register 0L	MK0	MK0L		○	○	○	FFH
FFE5H	Interrupt mask flag register 0H		MK0H		○	○		
FFE8H	Priority specification flag register 0L	PR0	PR0L		○	○	○	
FFE9H	Priority specification flag register 0H		PR0H		○	○		
FFECH	External interrupt mode register	INTM0			–	○	–	00H
FFF0H	Memory size select register	IMS		W	–	○	–	**Note 2**
FFF6H	Key return mode register	KRM		R/W	○	○	–	02H
FFF7H	Pull-up resistor option register	PUO			○	○	–	00H
FFF8H	Memory extension mode register	MM			○	○	–	10H
FFF9H	Watchdog timer mode register	WDTM			○	○	–	00H
FFFAH	Oscillation stabilization time select register	OSTS			–	○	–	04H
FFFBH	Processor clock control register	PCC			○	○	–	

Notes 1. The external access area cannot be accessed by SFR addressing. Access this area by means of direct addressing.

2. The value on reset differs depending on the part number as shown below:

42H for μPD78011H, 44H for μPD78012H, C6H for μPD78013H, C8H for μPD78014H

3.3 Addressing Instruction Address

An instruction address is determined by the contents of the program counter (PC). The contents of the PC is usually automatically incremented by the number of bytes of an instruction to be fetched (by 1 per byte) every time an instruction is excuted. When an instruction that causes program execution to branch is performed, the address information of the branch destination is set to the PC by means of the following addressing (for details of each instruction, refer to **78K/0 Series User's Manual - Instruction (U12326E)**).

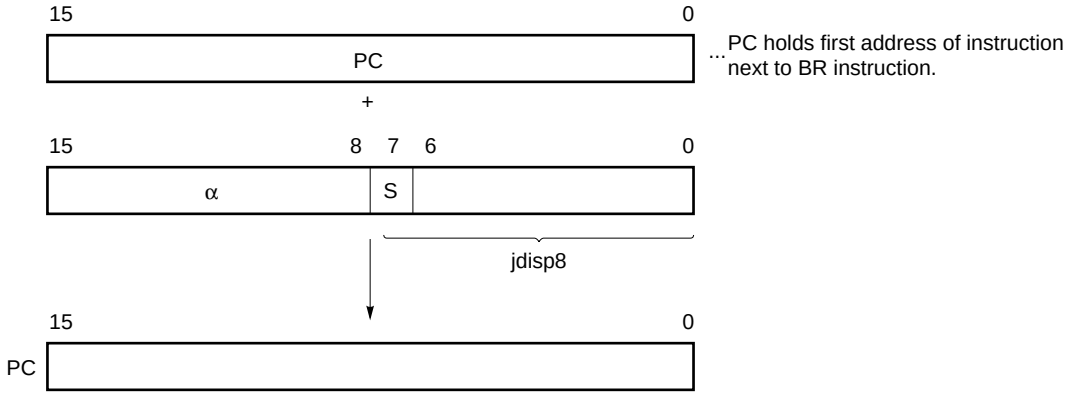
3.3.1 Relative addressing

[Function]

The 8-bit immediate data (displacement value: jdisp8) of the instruction code is added to the first address of the next instruction, the resultant sum is transferred to the program counter (PC), and the program branches. The displacement value is treated as signed 2's complement data (–128 to +127), and bit 7 serves as a sign bit. In other words, in the relative addressing, program branches to an address in the range –128 to +127 counting from the first address of the instruction that follows a branch instruction.

This addressing is used when "BR $addr16" instruction or conditional branch instruction is executed.

[Operation]



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

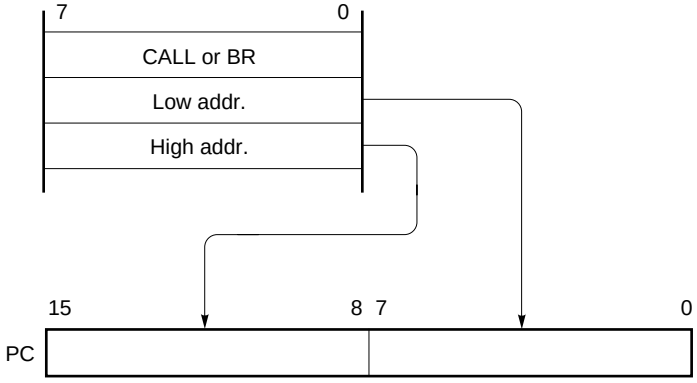
3.3.2 Immediate addressing

[Function]

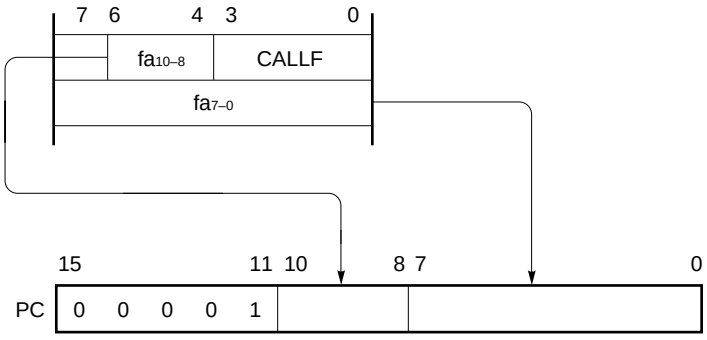
The immediate data in an instruction word is transferred to the program counter (PC), and execution branches. This addressing is used when the “CALL !addr16”, “BR !addr16”, or “CALLF !addr11” instruction is executed. CALL !addr16 and BR !addr16 instructions branches to the entire memory area while CALLF !addr11 branches to the area 0800H to 0FFFH.

[Operation]

When “CALL !addr16” or “BR !addr16” instruction is executed



When “CALLF !addr11“ instruction is executed

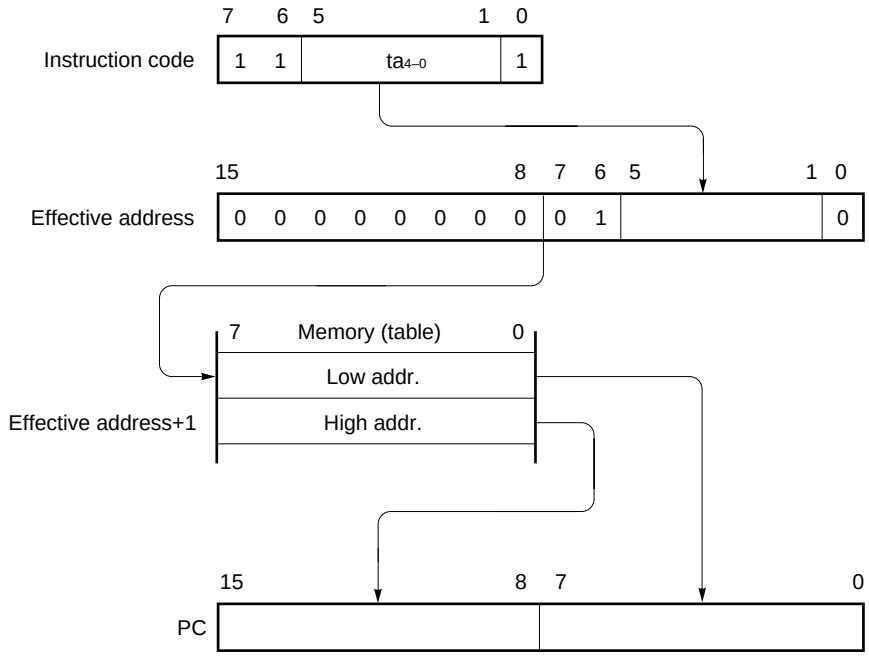


3.3.3 Table indirect addressing

[Function]

The contents of a specific location table (branch destination address) addressed by the immediate data of bits 1 to 5 of an instruction code are transferred to the program counter (PC), and program execution branches. This addressing is used when the “CALLT [addr5]” instruction is executed. In this addressing, the address value stored in the memory table (40H to 7FH) is referenced, based on which a program can branch to anywhere in the entire memory space.

[Operation]



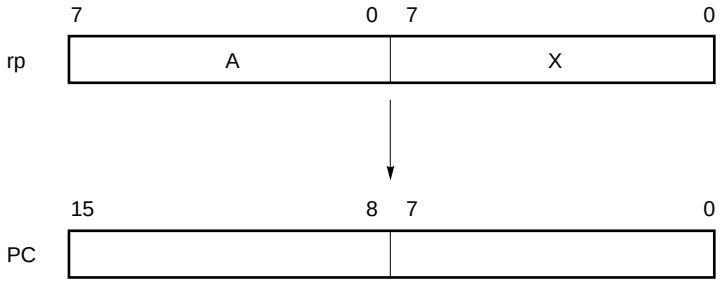
3.3.4 Register addressing

[Function]

The contents of the register pair (AX) specified by an instruction word are transferred to the program counter (PC), and program execution branches.

This addressing is used when the “BR AX” instruction is executed.

[Operation]



3.4 Addressing of Operand Address

3.4.1 Data memory addressing

The method of specifying the address of the instruction to be executed next or the address of the register or memory to be manipulated when an instruction is executed is called addressing.

The address of the instruction to be executed next is specified by the program counter (PC) (for details, refer to **3.3 Addressing Instruction Address**).

To address the memory to be manipulated when an instruction is executed, the μPD78014H Subseries are provided with many addressing modes to facilitate manipulation of the memory. By using these addressing modes, special function registers (SFRs) and general-purpose registers can be addressed according to their own functions. Figures 3-11 through 3-14 illustrate the addressing of the data memory.

Figure 3-11. Data Memory Addressing (μPD78011H)

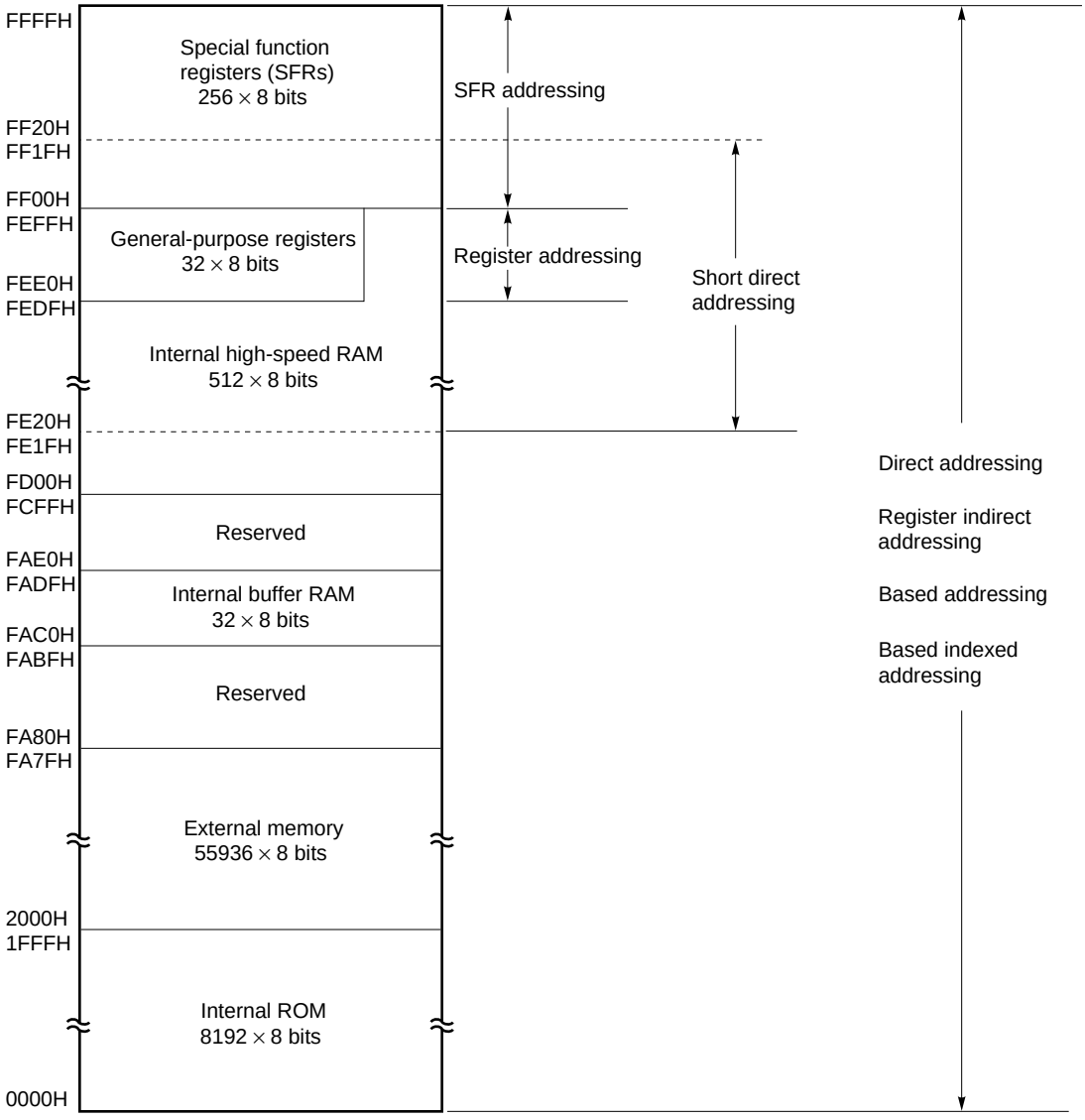


Figure 3-12. Data Memory Addressing (μPD78012H)

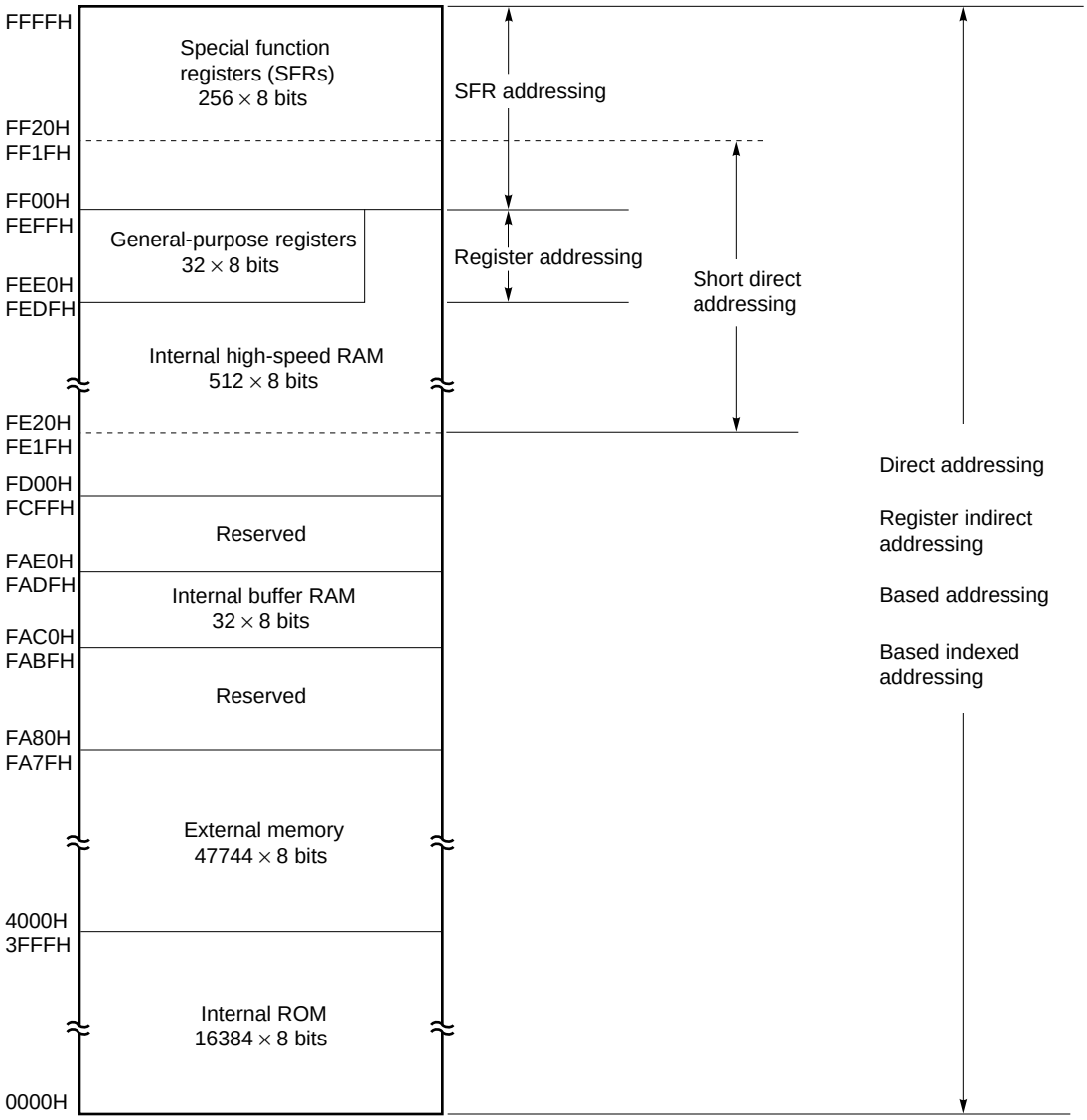


Figure 3-13. Data Memory Addressing (μPD78013H)

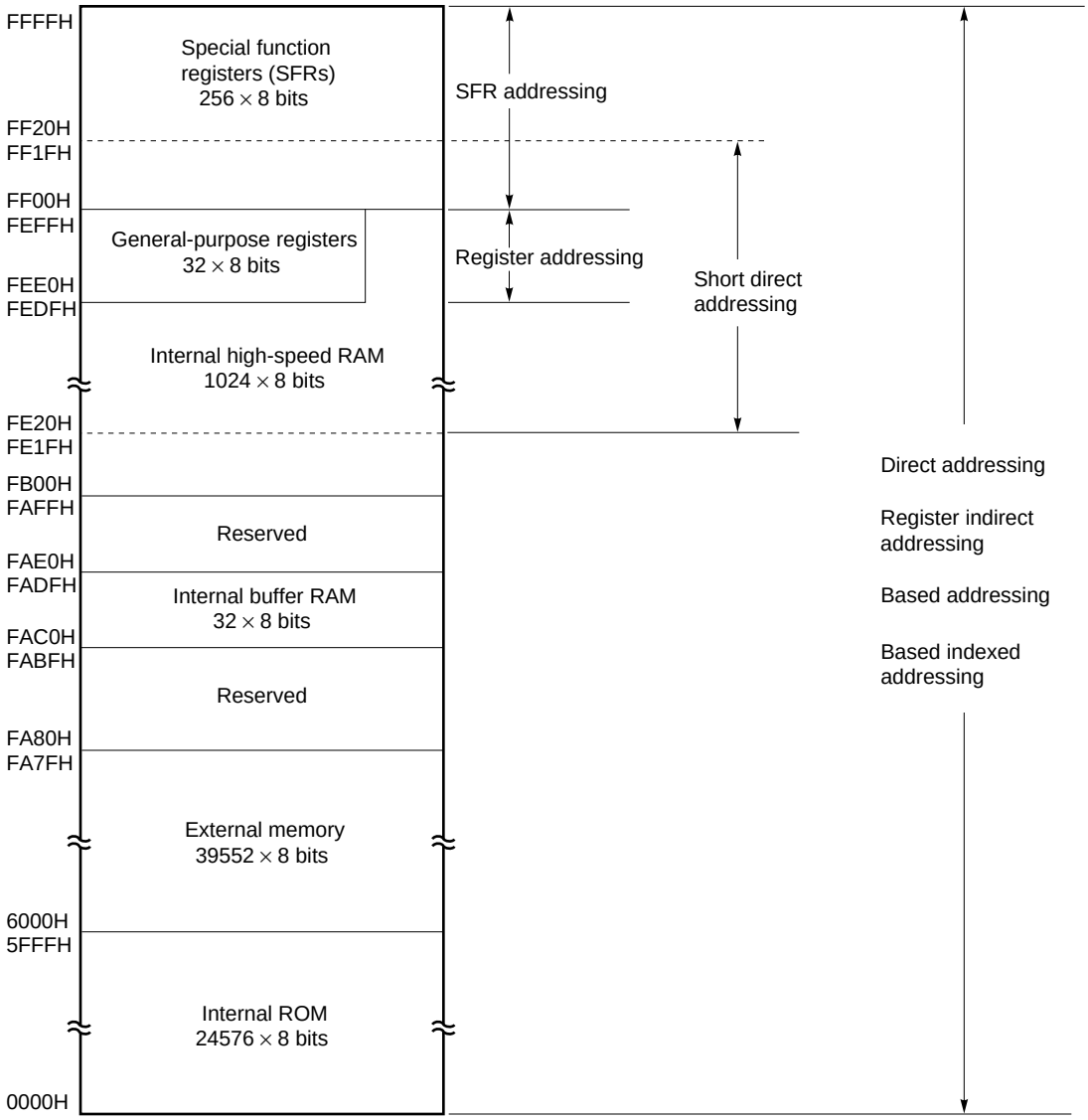
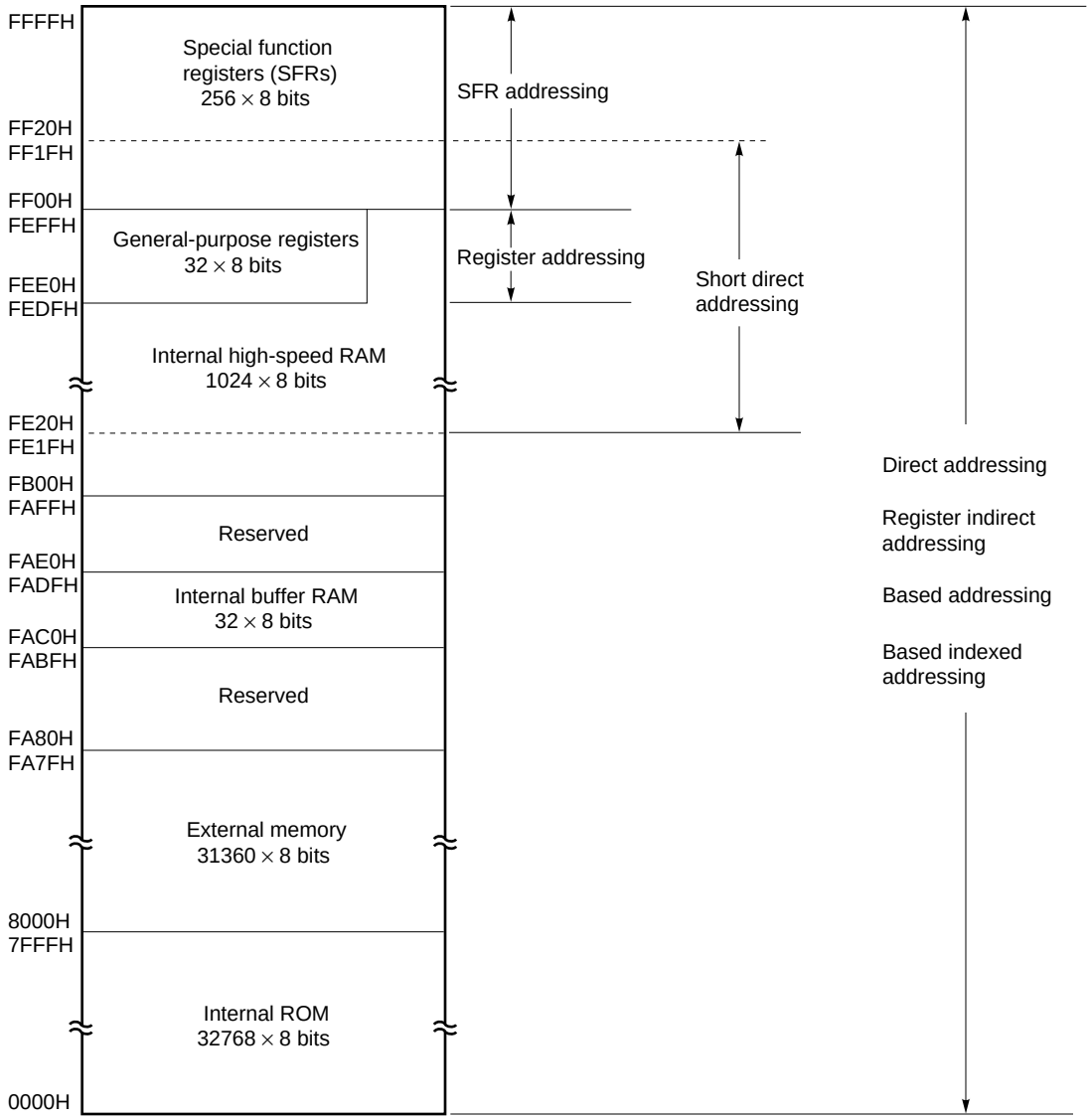


Figure 3-14. Data Memory Addressing (μPD78014H)



3.4.2 Implied addressing

[Function]

This addressing is to automatically (implicitly) address a register that functions as an accumulator (A or AX) in the general-purpose register area.

Of the instruction words of the μPD78014H Subseries, those that use implied addressing are as follows:

Instruction	Register Specified by Implied Addressing
MULU	Register A to store multiplicand and register AX to store product
DIVUW	Register AX to store dividend and quotient
ADJBA/ADJBS	Register A to store numeric value subject to decimal adjustment
ROR4/ROL4	Register A to store digit data subject to digit rotation

[Operand Format]

No specific operand format is used because the operand format is automatically determined by an instruction.

[Example]

MULU X

The product between registers A and X is stored in register AX as a result of executing a multiply instruction of 8 bits x 8 bits. In this operation, registers A and AX are specified by implied addressing.

3.4.3 Register addressing

[Function]

This addressing accesses as an operand a general-purpose register selected by the register specification code (Rn,RPn) in an instruction word from the register bank specified by the register bank select flags (RBS0 and RBS1).

Register addressing is used when an instruction that has the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified by 3 bits in the instruction code.

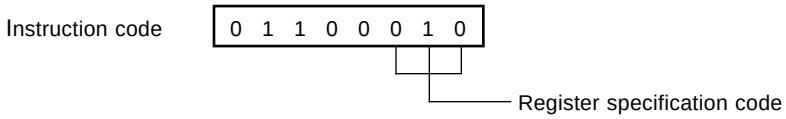
[Operand Format]

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

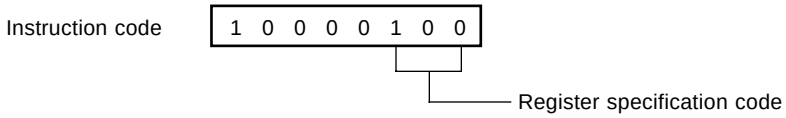
r and rp can be described not only in function name (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) but also in absolute name (R0 to R7, RP0 to RP3).

[Example]

MOV A, C; To select C register as r



INCW DE; To select DE register pair as rp



3.4.4 Direct addressing

[Function]

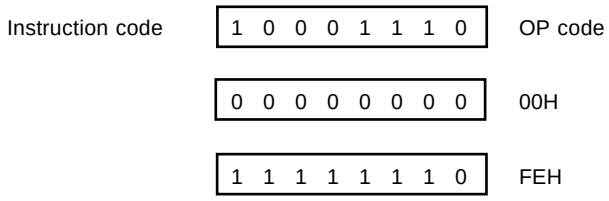
This addressing is to directly address a memory area specified by the immediate data in an instruction word as an operand address.

[Operand Format]

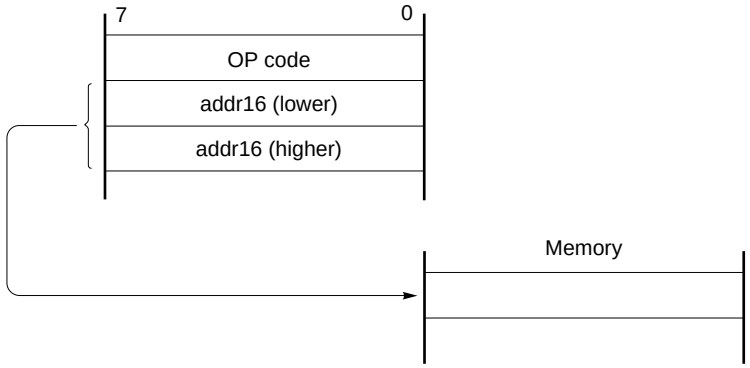
Representation	Description
addr16	Label or 16-bit immediate data

[Example]

MOV A, !FE00H; To specify FE00H as !addr16



[Operation]



3.4.5 Short direct addressing

[Function]

This addressing directly addresses a memory area to be manipulated from a fixed space by using the 8-bit data in an instruction word.

The fixed memory space which this addressing is applicable to is a 256-byte space of FE20H to FF1FH. The internal high-speed RAM is mapped to addresses FE20H to FEFFH, and special function registers (SFRs) are mapped to addresses FF00H to FF1FH.

The SFR area (FF00H to FF1FH) supports short direct addressing. Mapped in this area are ports, and compare and capture registers of timer/event counters that are frequently accessed on program. These SFRs can be manipulated with a few bytes and clocks.

Bit 8 of the effective address is 0 if the 8-bit immediate data is in a range of 20H to FFH, and 1 if the data is in a range of 00H to 1FH. Refer to **[Operation]** on the following page.

[Operand Format]

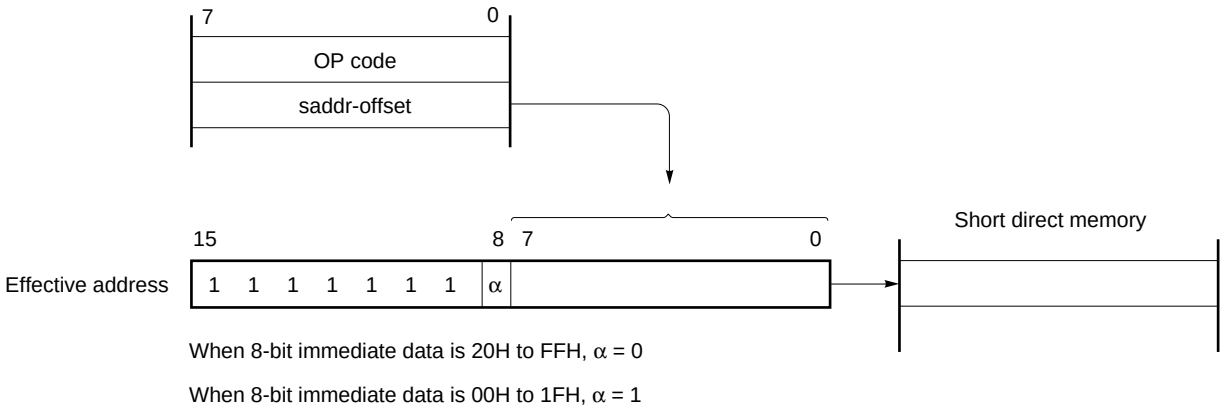
Representation	Description
saddr	Label or immediate data FE20H to FF1FH
saddrp	Label or immediate data FE20H to FF1FH (even address only)

[Example]

MOV FE30H, #50H; To specify FE30H as saddr and 50H as immediate data

Instruction code	0 0 0 1 0 0 0 1	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)
	0 1 0 1 0 0 0 0	50H (immediate data)

[Operation]



When 8-bit immediate data is 20H to FFH, α = 0

When 8-bit immediate data is 00H to 1FH, α = 1

3.4.6 Special function register (SFR) addressing

[Function]

This addressing is to address special function registers (SFRs) mapped to the memory by using an 8-bit immediate data in an instruction word.

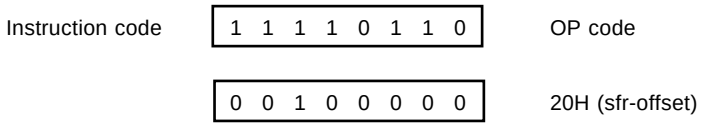
This addressing is applied to a 240-byte space of FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped to an area of FF00H to FF1FH can also be accessed by means of short direct addressing.

[Operand Format]

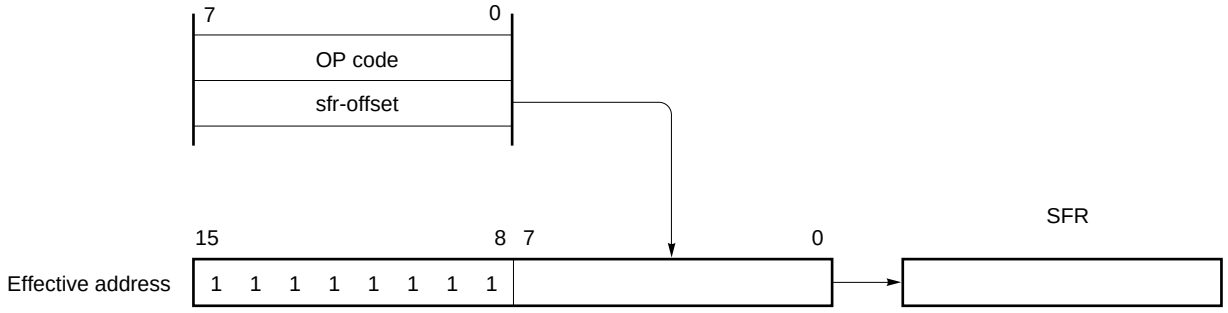
Representation	Description
sfr	Special function register name
sfrp	Name of special function register that can be manipulated in 16-bit units (even address only)

[Example]

MOV PM0, A: To select PM0 (FF20H) as sfr



[Operation]



3.4.7 Register indirect addressing

[Function]

This addressing is to address a memory area by using the contents of the register pair specified as an operand. The register pair to be accessed is specified by the register bank selection flag (RBS0, RBS1) and the register-pair specification code in instruction codes. This addressing can address the entire memory space.

[Operand Format]

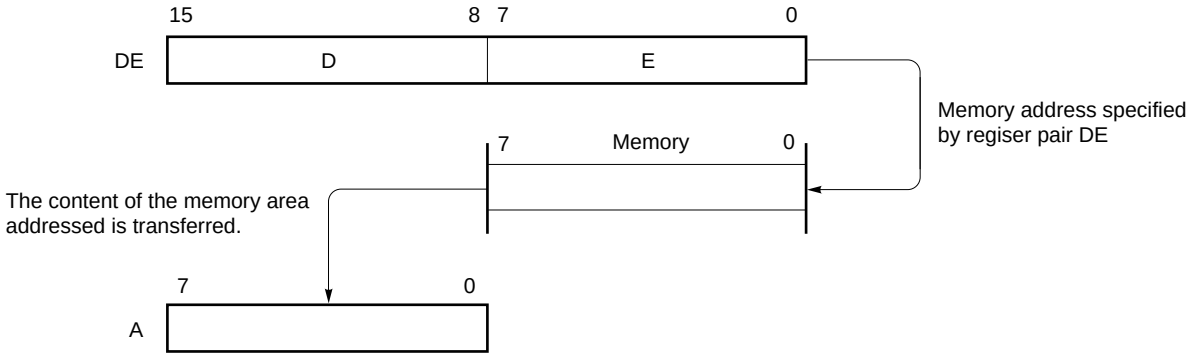
Representation	Description
—	[DE], [HL]

[Example]

MOV A, [DE]; To select [DE] as register pair

Instruction code	1	0	0	0	0	1	0	1

[Operation]



3.4.8 Based addressing

[Function]

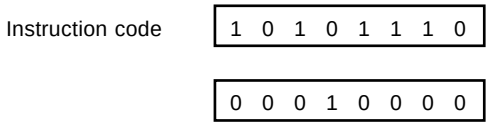
This addressing is to address a memory area by using the result of adding 8-bit immediate data to the contents of the HL register pair as a base register. The HL register pair is selected from the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is executed by extending the offset data to 16 bits as a positive number. A carry from the 16th bit is ignored. This addressing is used to address the entire memory space.

[Operand Format]

Representation	Description
—	[HL + byte]

[Example]

MOV A, [HL+10H]; To specify 10H as byte



3.4.9 Based indexed addressing

[Function]

This addressing is to address a memory area by using the result of adding the contents of the B or C register specified in an instruction word to the HL register pair as a base register. HL, B, or C register to be accessed is selected from the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is executed by extending the content of B or C register to 16 bits as a positive number. A carry from the 16th bit is ignored.

This addressing is used to address the entire memory space.

[Operand Format]

Representation	Description
—	[HL + B], [HL + C]

[Example]

When MOV A, [HL+B]

Instruction code

1	0	1	0	1	0	1	1

3.4.10 Stack addressing

[Function]

This addressing is to indirectly address the stack area by using the contents of the stack pointer (SP).

This addressing is automatically used to save/restore register contents when the PUSH, POP, subroutine call, or return instruction is executed, or when an interrupt request is generated.

The stack addressing can access the internal high-speed RAM area only.

[Example]

When PUSH DE is executed

Instruction code	1	0	1	1	0	1	0	1

4.1 Functions of Ports

The μPD78014H Subseries is provided with two input port pins and 51 I/O port pins. Figure 4-1 shows these port pins. Each port can be manipulated in 1-bit or 8-bit units and controlled in various ways. Moreover, some port pins also serve as the I/O pins of the internal hardware.

Figure 4-1. Types of Ports

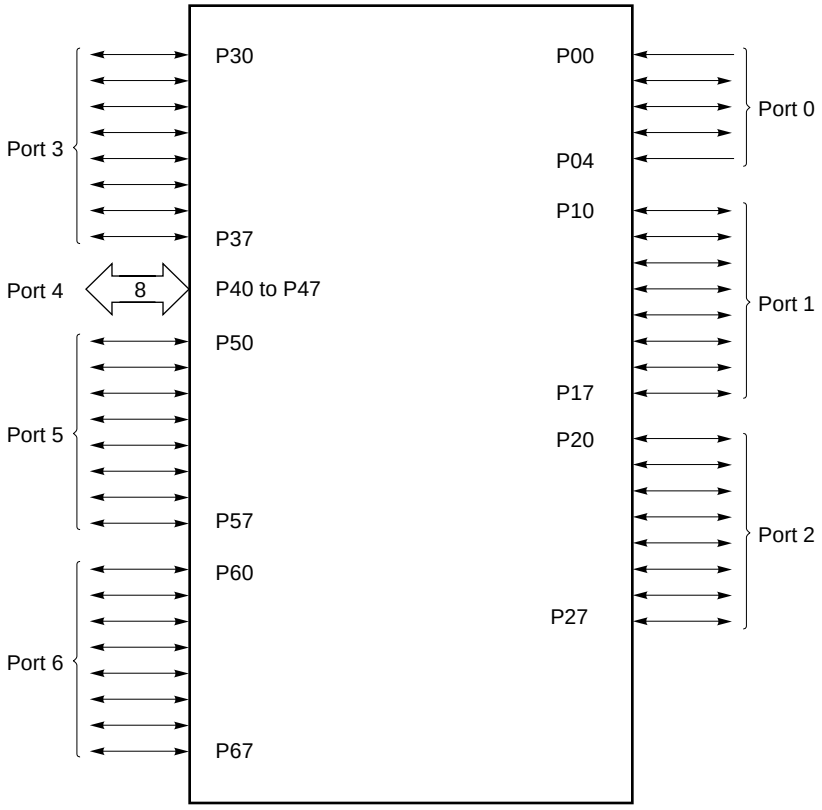


Table 4-1. Port Functions

Pin Name		Function		Alternate Function
Port 0	P00	5-bit I/O port.	Input only.	INTP0/TI0
	P01		Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.	INTP1
	P02			INTP2
	P03			INTP3
	P04		Input only.	XT1
Port 1	P10 to P17	8-bit I/O port. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.		ANI0 to ANI7
Port 2	P20	8-bit I/O port. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.		SI1
	P21			SO1
	P22			$\overline{\text{SCK1}}$
	P23			STB
	P24			BUSY
	P25			SI0/SB0
	P26			SO0/SB1
	P27			$\overline{\text{SCK0}}$
Port 3	P30	8-bit I/O port. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.		TO0
	P31			TO1
	P32			TO2
	P33			TI1
	P34			TI2
	P35			PCL
	P36			BUZ
	P37			–
Port 4	P40 to P47	8-bit I/O port. Can be specified for input/output in 8-bit units. When used as input port, internal pull-up resistor can be connected by software. Test input flag (KRIF) is set to 1 at falling edge of these pins.		AD0 to AD7
Port 5	P50 to P57	8-bit I/O port. Can directly drive LED. Can be specified for input/output bitwise. When used as input port, internal pull-up resistor can be connected by software.		A8 to A15
Port 6	P60	8-bit I/O port. Can be specified for input/output bitwise.	N-ch open drain I/O port. Internal pull-up resistor can be specified by mask option. Can directly drive LED.	–
	P61			
	P62			
	P63			
	P64		When used as input port, internal pull-up resistor can be connected by software.	$\overline{\text{RD}}$
	P65			$\overline{\text{WR}}$
	P66			$\overline{\text{WAIT}}$
	P67			ASTB

Caution For the port pins having alternate functions (refer to 2.1 (1) Port pins), the following operations are prohibited. If these prohibitions are not observed, the operation within the total error rating during A/D conversion cannot be guaranteed.

<1> Changing the value of the output latch of the given port when it is used as a port

<2> Changing the output level of the pin used as an output even if it is not used as a port

4.2 Port Configuration

A port consists of the following hardware:

Table 4-2. Port Configuration

Item		Configuration
Control register		Port mode register (PMm: m = 0, 1, 2, 3, 5, or 6) Pull-up resistor option register (PUO) Memory extension mode register (MM)[Note] Key return mode register (KRM)
Port	Total	53 lines
	Input	2 lines
	I/O	51 lines
Pull-up resistor	Total	51 lines
	Software control	47 lines
	Mask option control	4 lines

Note The memory extension mode register (MM) specifies the input/output mode of port 4.

4.2.1 Port 0

This is a 5-bit I/O port with output latch. P01 to P03 pins can be specified in the input or output mode in 1-bit units by using the port mode register 0 (PM0). P00 and P04 pins are input port pins. When P01 to P03 pins are used as input port pins, internal pull-up resistors can be connected in 3-bit units by using the pull-up resistor option register (PUO).

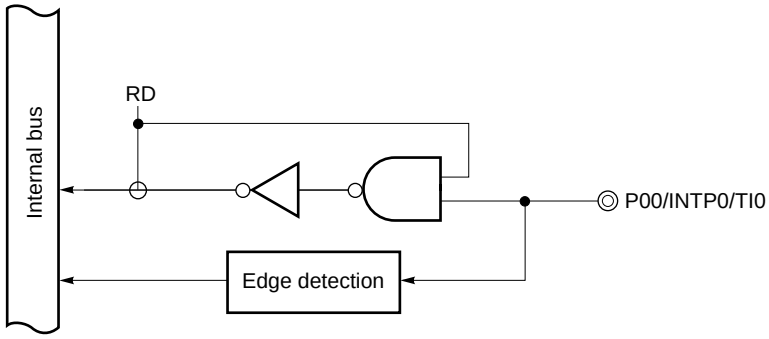
The five port pins are also used to input external interrupt requests, an external count clock to the timer, and connect a crystal for oscillator for subsystem clock oscillation.

Port 0 is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 4-2 through 4-4 show the block diagrams of port 0.

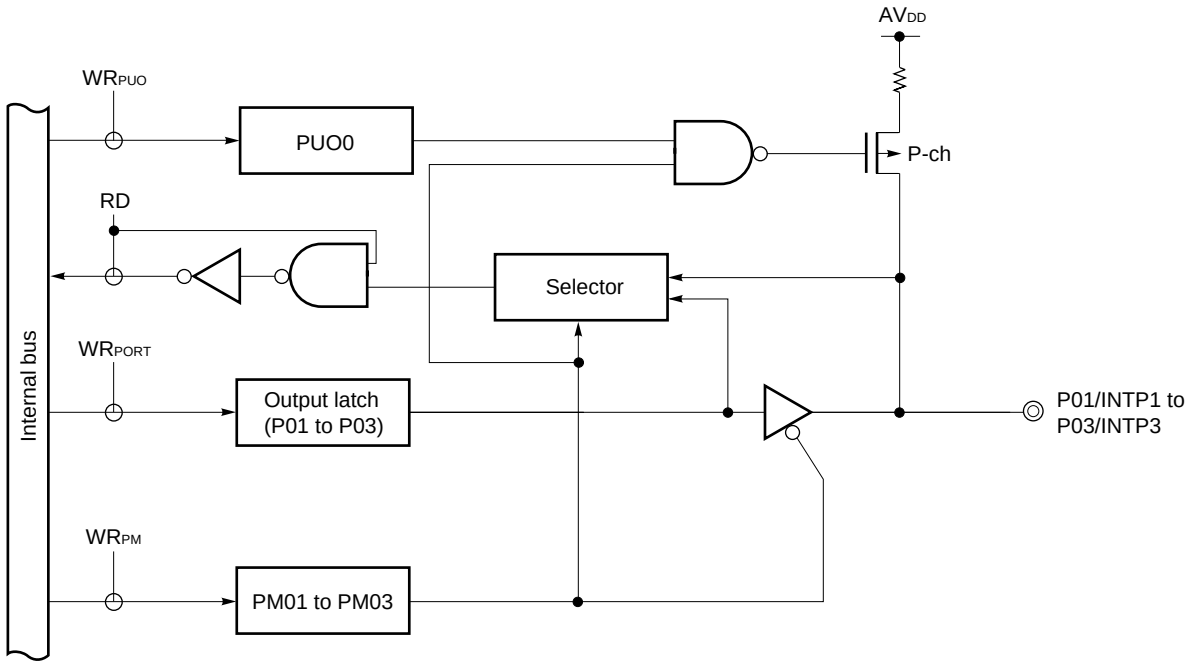
Caution Because port 0 is also used as an external interrupt request input pin, an interrupt request flag is set when the port is specified in the output mode and its output level is changed. When using port 0 in the output mode, therefore, set the interrupt mask flag to 1.

Figure 4-2. Block Diagram of P00



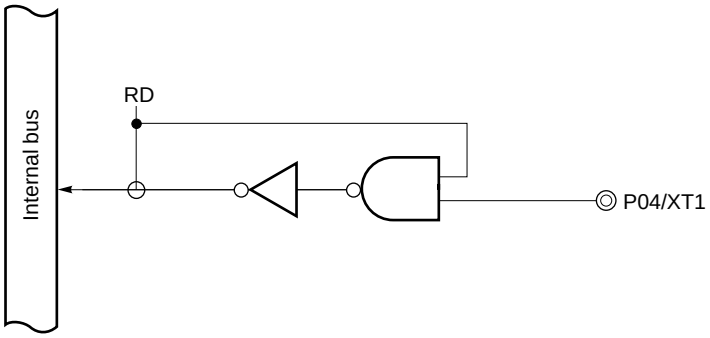
RD: read signal of port 0

Figure 4-3. Block Diagram of P01 to P03



PUO : pull-up resistor option register
PM : port mode register
RD : read signal of port 0
WR : write signal of port 0

Figure 4-4. Block Diagram of P04



RD: read signal of port 0

4.2.2 Port 1

This is an 8-bit I/O port with output latch. It can be specified in the input or output mode in 1-bit units by using the port mode register 1 (PM1). When using P10 to P17 pins as input port pins, internal pull-up resistors can be connected in 8-bit units by using the pull-up resistor option register (PUO).

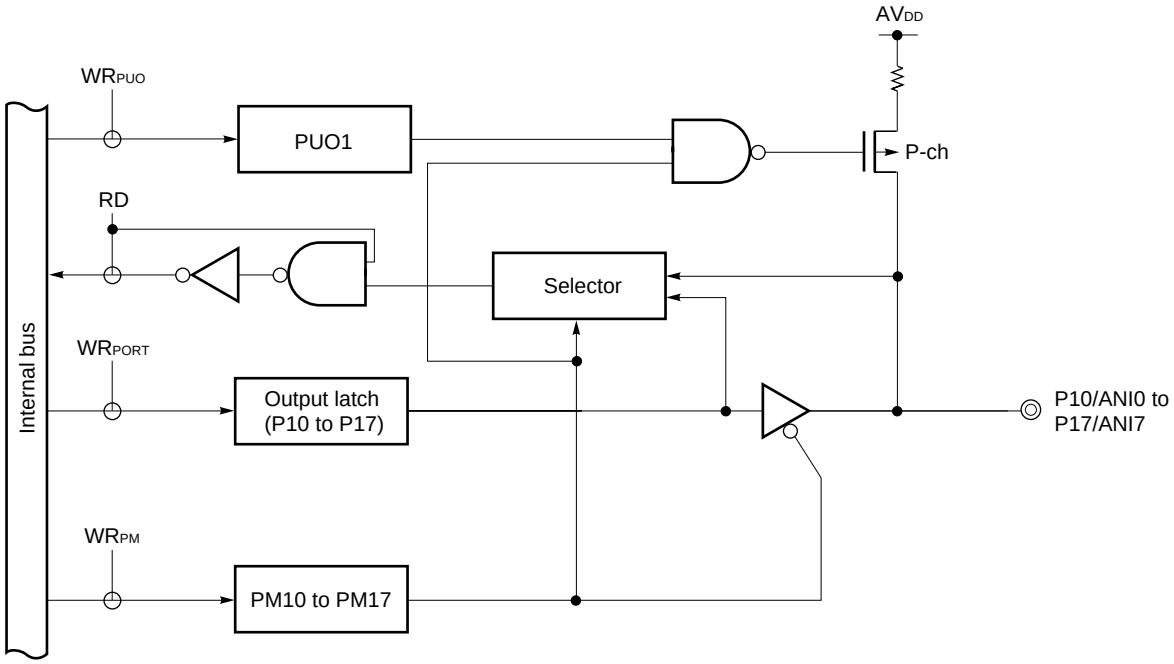
The pins of this port are also used as the analog input pins of the A/D converter.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-5 shows the block diagram of port 1.

Caution The internal pull-up resistor cannot be connected to the pin that is used as the analog input pin of the A/D converter.

Figure 4-5. Block Diagram of P10 to P17



PUO : pull-up resistor option register
PM : port mode register
RD : read signal of port 1
WR : write signal of port 1

4.2.3 Port 2

This is an 8-bit I/O port with output latch. P20 to P27 pins can be specified in the input or output mode in 1-bit units by using the port mode register 2 (PM2). When using P20 to P27 pins as input port pins, internal pull-up resistors can be connected in 8-bit units by using the pull-up resistor option register (PUO).

The pins of this port are also used as the data I/O pin, clock I/O pin, busy signal input pin for automatic transmission/reception, and strobe signal output pin of the serial interface.

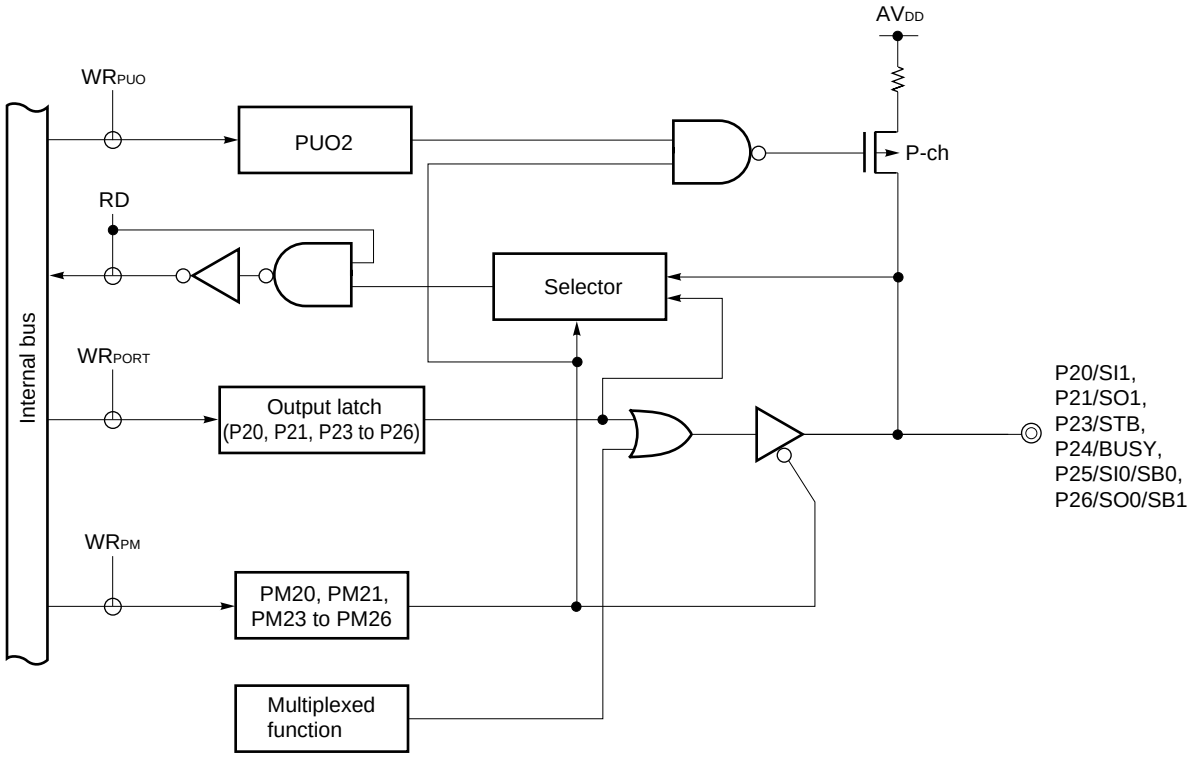
This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 4-6 and 4-7 show the block diagrams of port 2.

Cautions 1. When using the pins of port 2 as multiplexed pins, the I/O or output latch must be set according to the function to be used. For how to set the latches, refer to Figure 13-3. Format of Serial Operation Mode Register 0 and Figure 14-3. Format of Serial Operation Mode Register 1.

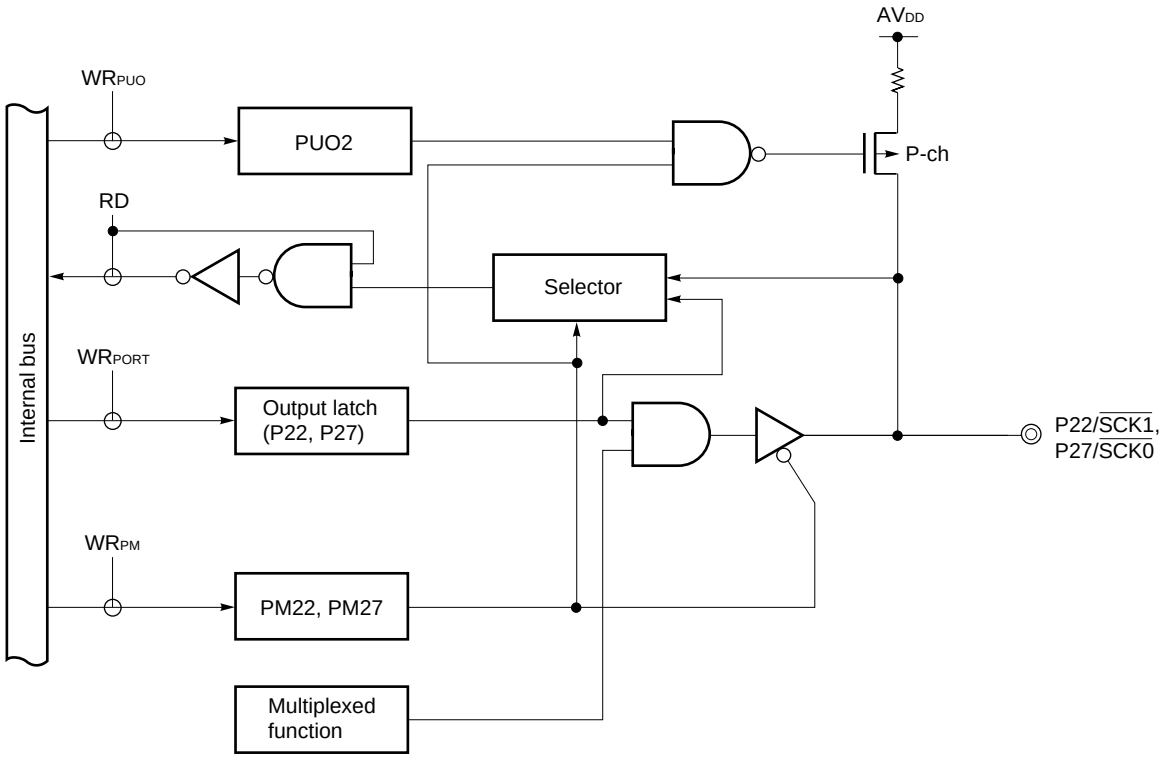
2. To read the status of the pin in the SBI mode, set PM2n of PM2 to 1 (n = 5, 6) (refer to 13.4.3 (10) Slave busy status judgement).

★ **Figure 4-6. Block Diagram of P20, P21, and P23 to 26**



PUO : pull-up resistor option register
PM : port mode register
RD : read signal of port 2
WR : write signal of port 2

★ **Figure 4-7. Block Diagram of P22 and P27**



PUO : pull-up resistor option register
PM : port mode register
RD : read signal of port 2
WR : write signal of port 2

4.2.4 Port 3

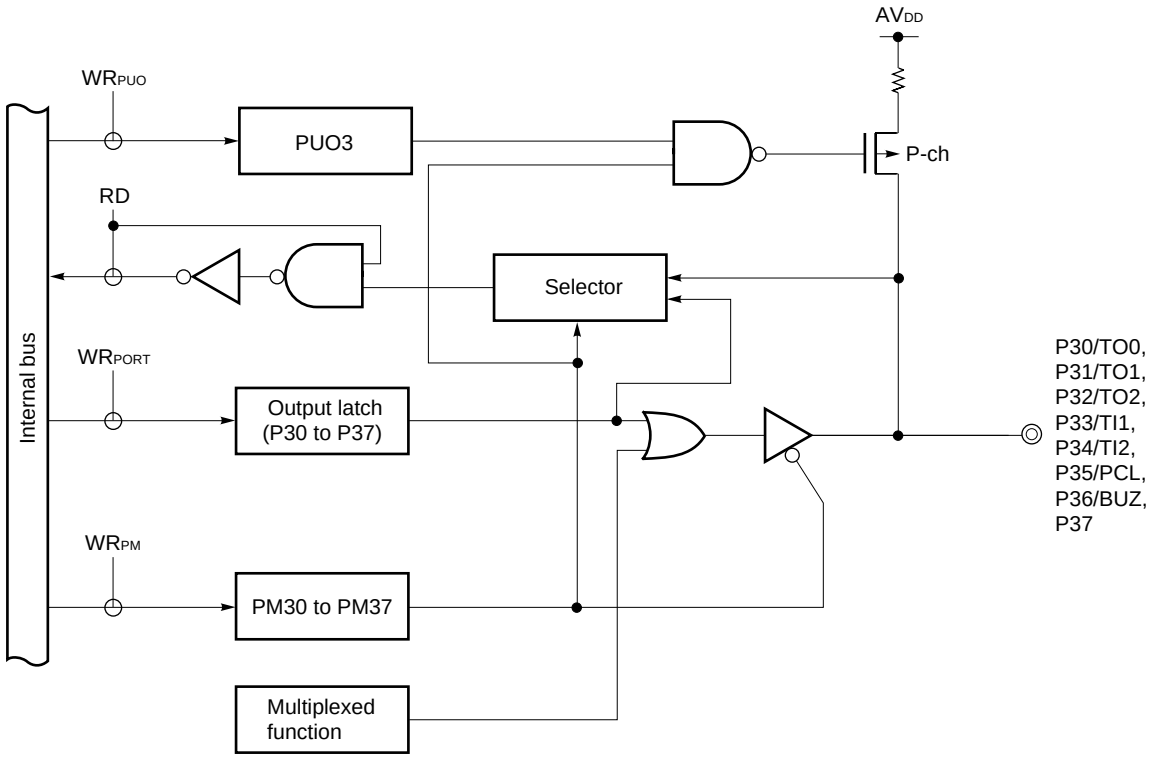
This is an 8-bit I/O port with output latch. P30 to P37 pins can be specified in the input or output mode in 1-bit units by using the port mode register 3 (PM3). When using P30 to P37 pins as input port pins, internal pull-up resistors can be connected in 8-bit units by using the pull-up resistor option register (PUO).

The pins of this port are also used as the timer I/O, clock output, and buzzer output pins.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

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

★ **Figure 4-8. Block Diagram of P30 to P37**



PUO : pull-up resistor option register
PM : port mode register
RD : read signal of port 3
WR : write signal of port 3

4.2.5 Port 4

This is an 8-bit I/O port with output latch. P40 to P47 pins can be specified in the input or output mode in 8-bit units by using the memory extension mode register (MM). When using P40 to P47 pins as input port pins, internal pull-up resistors can be connected in 8-bit units by using the pull-up resistor option register (PUO).

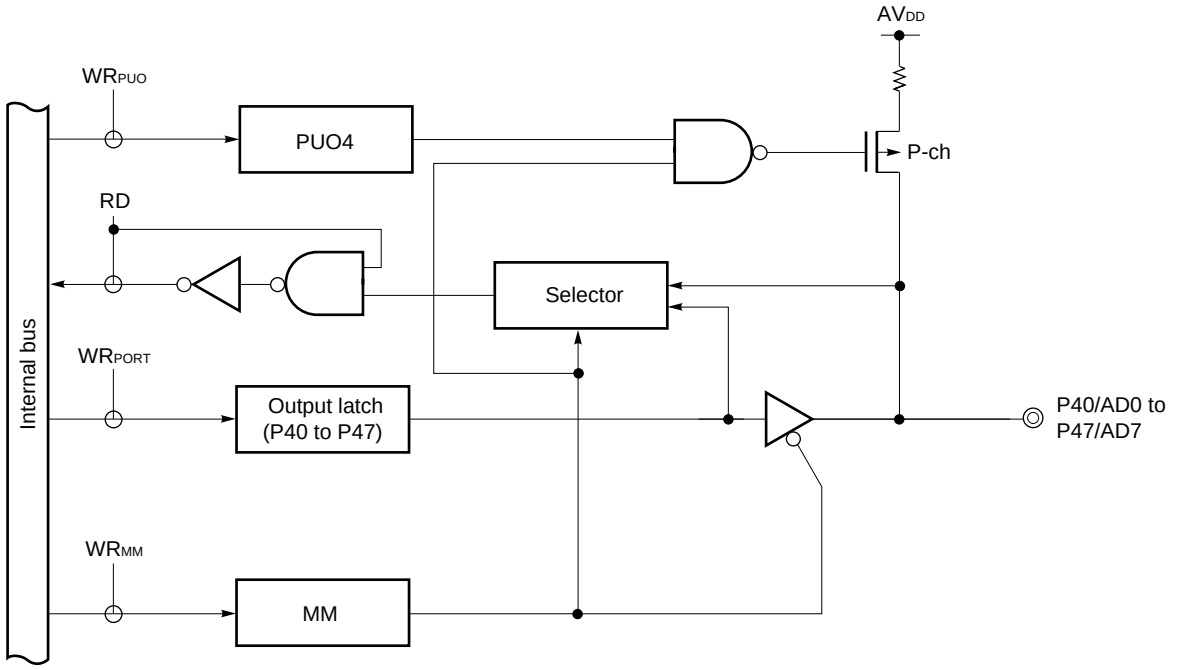
When the falling edge of any of the pins of this port is detected, a test input flag (KRIF) can be set to 1.

These port pins are also multiplexed with an address/data bus that is used in the external memory extension mode.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

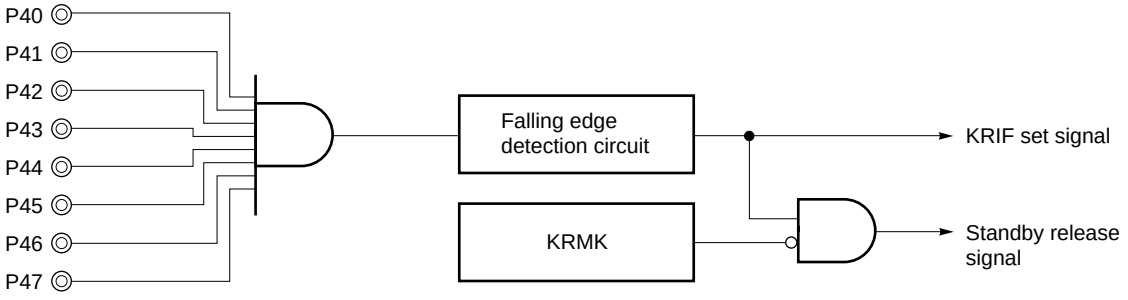
Figure 4-9 shows the block diagram of port 4, and Figure 4-10 shows the block diagram of the falling edge detection circuit.

Figure 4-9. Block Diagram of P40 to P47



PUO : pull-up resistor option register
MM : memory extension mode register
RD : read signal of port 4
WR : write signal of port 4

Figure 4-10. Block Diagram of Falling Edge Detection Circuit



4.2.6 Port 5

This is an 8-bit I/O port with output latch. P50 to P57 pins can be specified in the input or output mode in 1-bit units by using the port mode register 5 (PM5). When using P50 to P57 pins as input port pins, internal pull-up resistors can be connected in 8-bit units by using the pull-up resistor option register (PUO).

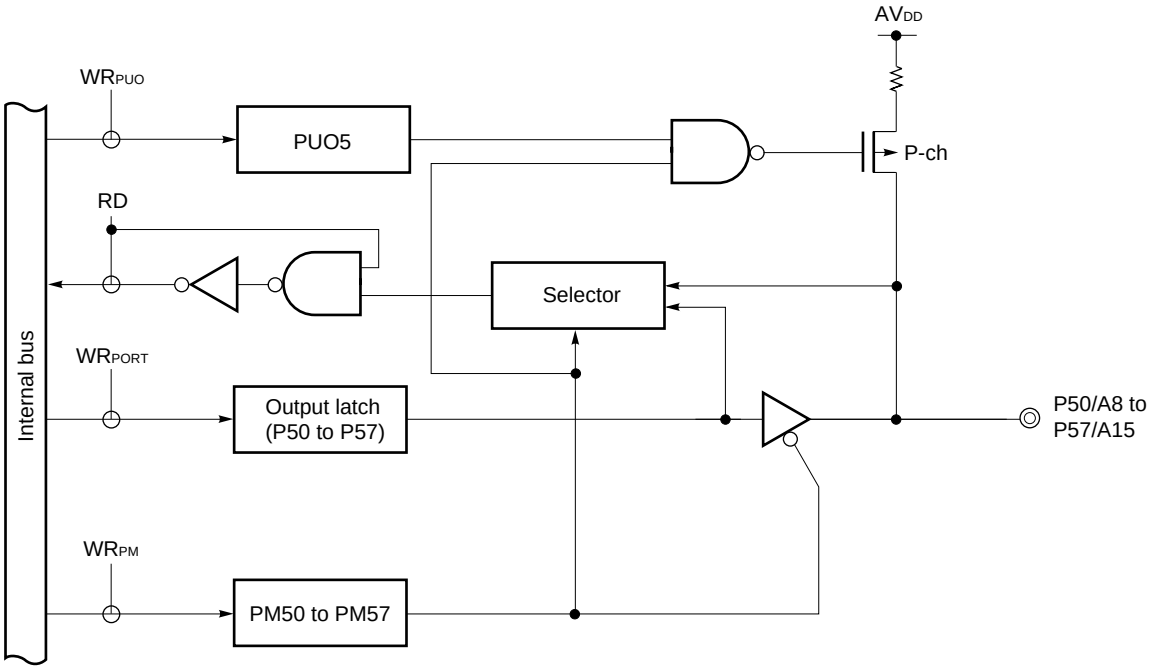
Port 5 can directly drive an LED.

These port pins are also multiplexed with an address bus that is used in the external memory extension mode.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figure 4-11 shows the block diagram of port 5.

Figure 4-11. Block Diagram of P50 to P57



PUO : pull-up resistor option register
PM : port mode register
RD : read signal of port 5
WR : write signal of port 5

4.2.7 Port 6

This is an 8-bit I/O port with output latch. P60 to P67 pins can be specified in the input or output mode in 1-bit units by using the port mode register 6 (PM6).

This port can be connected with pull-up resistors as described in the following table. The number of bits in units of which the pull-up resistor can be connected differs depending on whether the high- or low-order 4 bits of the port are involved.

Table 4-3. Pull-up Resistors in Port 6

	High-order 4 Bits (P64 to P67 pins)	Low-order 4 Bits (P60 to P63 pins)
Mask ROM version	Internal pull-up resistor can be connected in 4-bit units by PUO6	Internal pull-up resistor can be connected in 1-bit units by mask option

PUO6: bit 6 of pull-up resistor option register (PUO)

P60 to P63 pins can directly drive an LED.

P64 to P67 pins are also used to output control signals in the external memory extension mode.

This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

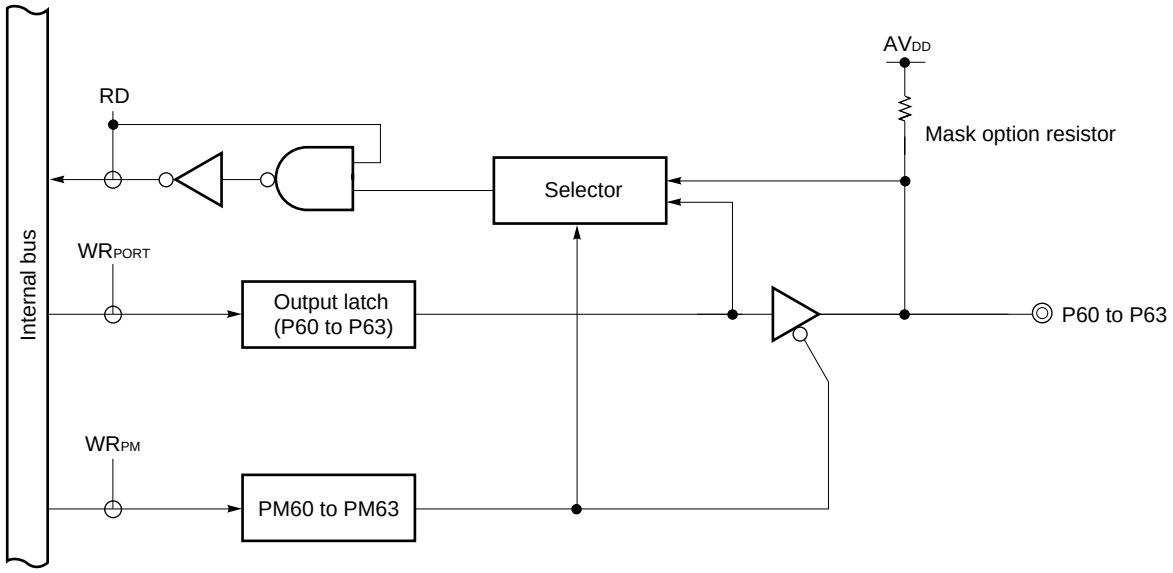
Figures 4-12 and 4-13 show the block diagrams of port 6.

Cautions 1. P66 can be used as an I/O port pins when no external wait state is used in the external memory extension mode.

2. The value of the low-level input leakage current flowing through P60 through P63 differs depending on the following conditions:

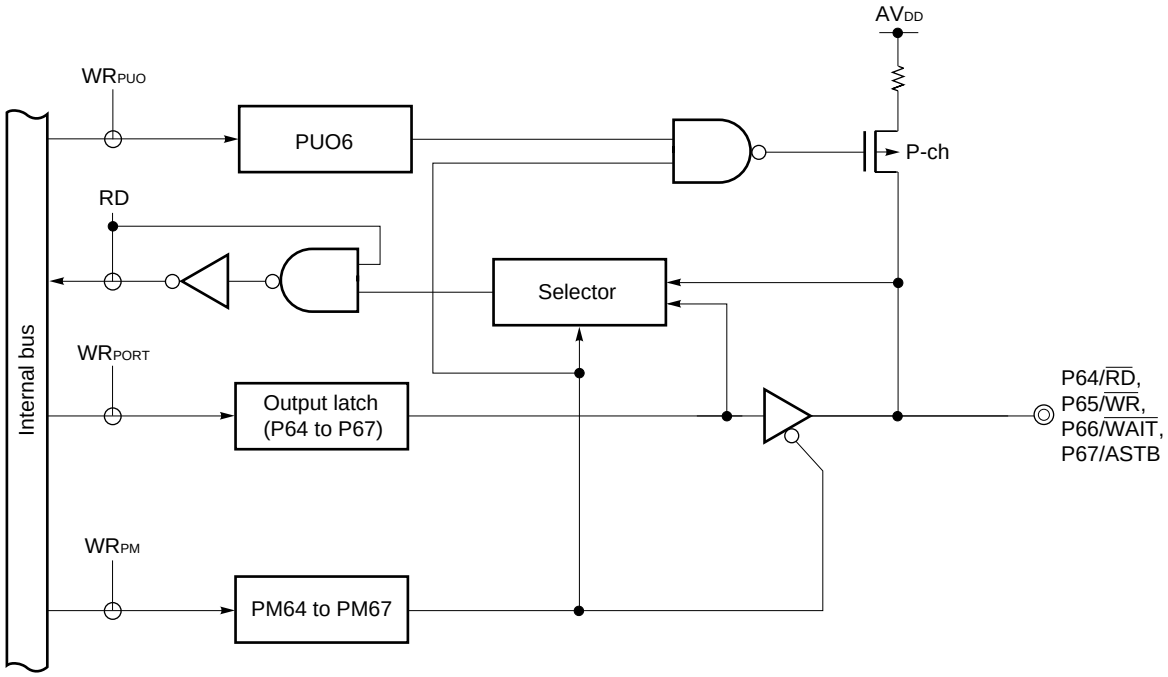
- When pull-up resistor is connected: always –3 μA (MAX.)
- When pull-up resistor is not connected
  - For duration of 3 clocks (without wait) when a read instruction is executed to port 6 (P6) or port mode register 6 (PM6) : –200 μA (MAX.)
  - Others : –3 μA (MAX.)

Figure 4-12. Block Diagram of P60 to P63



PM : port mode register
RD : read signal of port 6
WR : write signal of port 6

Figure 4-13. Block Diagram of P64 to P67



PUO : pull-up resistor option register
PM : port mode register
RD : read signal of port 6
WR : write signal of port 6

4.3 Registers Controlling Port Functions

The following four types of registers control the ports:

- Port mode registers (PM0, PM1, PM2, PM3, PM5, PM6)
- Pull-up resistor option register (PUO)
- Memory extension mode register (MM)
- Key return mode register (KRM)

(1) Port mode registers (PM0, PM1, PM2, PM3, PM5, PM6)

These registers set the corresponding ports in the input or output mode in 1-bit units.

PM0, PM1, PM2, PM3, PM5, and PM6 are manipulated by a 1-bit or 8-bit memory manipulation instruction.

When the $\overline{\text{RESET}}$ signal is input, the contents of PM0 are set to 1FH, and those of the other registers are set to FFH.

To use the multiplexed function of a port pin, set the port mode register corresponding to that pin and the output latch as shown in Table 4-4.

Cautions 1. P00 and P04 pins are input only pins.

2. P40 to P47 pins are specified in the input or output mode by the memory extension mode register (MM).

3. Because port 0 is multiplexed with external interrupt request input pins, interrupt request flags are set when the output mode of the port function is specified and the output level is changed. To use this port in the output mode, therefore, set 1 to the interrupt mask flags in advance.

Table 4-4. Setting of Port Mode Register and Output Latch when Multiplexed Function Is Used

Pin Name	Multiplexed Function		PM××	P××	Pin Name	Multiplexed Function		PM××	P××
	Name	I/O				Name	I/O		
P00	INTP0	Input	1 (fixed)	None	P36	BUZ	Output	0	0
	TI0	Input	1 (fixed)	None	P40 to P47	AD0 to AD7	I/O	×Note 2	
P01 to P03	INTP1 to INTP3	Input	1	×	P50 to P57	A8 to A15	Output	×Note 2	
P04Note 1	XT1	Input	1 (fixed)	None	P64	$\overline{RD}$	Output	×Note 2	
P10 to P17Note 1	ANI0 to ANI7	Input	1	×	P65	$\overline{WR}$	Output	×Note 2	
P30 to P32	TO0 to TO2	Output	0	0	P66	$\overline{WAIT}$	Input	×Note 2	
P33, P34	TI1, TI2	Input	1	×	P67	ASTB	Output	×Note 2	
P35	PCL	Output	0	0					

Notes 1. The contents of the read data are undefined if a read instruction is excuted to these ports while they are used as multiplexed function pins.

2. When using the multiplexed function of P40 to P47, P50 to P57, and P64 to P67 pins, set the function by using the memory extension mode register (MM).

Caution When using the pins of port 2 as serial interface pins, the input/output and output latch must be set in accordance with the function to be used. For the details of setting, refer to Figure 13-3. Format of Serial Operation Mode Register 0 and Figure 14-3. Format of Serial Operation Mode Register 1.

Remark × : don't care (need not to be set)
PM×× : port mode register
P×× : output latch of port

Figure 4-14. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	0	0	0	1	PM03	PM02	PM01	1	FF20H	1FH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W

PMmn	Selects I/O mode of Pmn pin (m = 0, 1, 2, 3, 5, 6 : n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(2) Pull-up resistor option register (PUO)

This register sets whether the internal pull-up resistor is connected to each port. The internal pull-up resistor can be connected only to the port pin which is specified by PUO to be connected to the internal pull-up resistor and the bit which is set in the input mode. The bit which is set in the output mode, and is used as the analog input pin of the A/D converter cannot be connected to the internal pull-up resistor, regardless of the setting of PUO.

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Cautions 1. P00 and P04 pins are not provided with an internal pull-up resistor.

2. When using the multiplexed functions of ports 1, 4, and 5, and P64 to P67 pins, the internal pull-up resistor cannot be used even when PUOm is set to 1 (m = 1, 4 to 6).

Figure 4-15. Format of Pull-up Resistor Option Register

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUO	0	PUO6	PUO5	PUO4	PUO3	PUO2	PUO1	PUO0	FFF7H	00H	R/W

PUOm	Selects internal pull-up resistor of Pm (m = 0, 1, 2, 3, 4, 5, 6)
0	Internal pull-up resistor is not used
1	Internal pull-up resistor is used

(3) Memory extension mode register (MM)

This register sets port 4 in the input or output mode.

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

This register is set to 10H when the $\overline{\text{RESET}}$ signal is input.

Remark MM also has functions to set the number of wait states and an external extension area, in addition to the function to set the input/output mode of port 4.

Figure 4-16. Format of Memory Extension Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
MM	0	0	PW1	PW0	0	MM2	MM1	MM0	FFF8H	10H	R/W

MM2	MM1	MM0	Selects single-chip/ memory extension mode		P40 to P47, P50 to P57, P64 to P67 pin status				
					P40 to P47	P50 to P53	P54, P55	P56, P57	P64 to P67
0	0	0	Single-chip mode		Port mode: Input	Port mode			
0	0	1			Port mode: Output				
0	1	1	Memory extension mode	256-byte mode	AD0 to AD7	Port mode			P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = $\overline{\text{WAIT}}$ P67 = ASTB
1	0	0		4-Kbyte mode		A8 to A11	Port mode		
1	0	1		16-Kbyte mode			A12, A13	Port mode	
1	1	1		Full address mode[Note]				A14, A15	
Others			Setting prohibited						

PW1	PW0	Wait control
0	0	No wait
0	1	Wait (1 wait state inserted)
1	0	Setting prohibited
1	1	Wait control by external wait pin

Note The full address mode is a mode in which all the areas of the 64-Kbyte address space, except the internal ROM, RAM, SFR, and prohibited areas, can be externally extended.

Remark P60 to P63 pins are set in the port mode regardless of whether the single-chip mode or memory extension mode is specified.

(4) Key return mode register (KRM)

This register enables/disables releasing the standby mode by key return signals (detection of the falling edge of port 4).

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

This register is set to 02H when the $\overline{\text{RESET}}$ signal is input.

Figure 4-17. Format of Key Return Mode Register

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
KRM	0	0	0	0	0	0	KRMK	KRIF	FFF6H	02H	R/W

KRIF	Key return signal detection flag
0	Falling edge of port 4 not detected
1	Falling edge of port 4 detected

KRMK	Controls standby mode by key return signal
0	Enables releasing standby mode
1	Disables releasing standby mode

Caution Be sure to clear KRIF to 0 to detect the falling edge of port 4 (KRIF is not cleared to 0 automatically).

4.4 Operation of Port Functions

The operation of a port differs depending on whether the port is set in the input or output mode, as described below.

4.4.1 Writing to I/O port

(1) In output mode

A value can be written to the output latch of a port by using a transfer instruction. The contents of the output latch can be output from the pins of the port.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. However, the status of the port pin is not changed because the output buffer is OFF.

The data once written to the output latch is retained until new data is written to the output latch.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set in the input mode and not subject to manipulation become undefined.

4.4.2 Reading from I/O port

(1) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch are not changed.

(2) In input mode

The status of a pin can be read by using a transfer instruction. The contents of the output latch are not changed.

4.4.3 Arithmetic operation of I/O port

(1) In output mode

An arithmetic operation can be performed with the contents of the output latch. The result of the operation is written to the output latch. The contents of the output latch are output from the port pins.

The data once written to the output latch is retained until new data is written to the output latch.

(2) In input mode

The contents of the output latch become undefined. However, the status of the pin is not changed because the output buffer is OFF.

Caution A 1-bit memory manipulation instruction is executed to manipulate 1 bit of a port. However, this instruction accesses the port in 8-bit units. When this instruction is executed to manipulate a bit of an input/output port, therefore, the contents of the output latch of the pin that is set in the input mode and not subject to manipulation become undefined.

4.5 Mask Option

The P60 to P63 pins can be connected to an internal pull-up resistor in 1-bit units by mask option.

CHAPTER 5 CLOCK GENERATION CIRCUIT

5.1 Function of Clock Generation Circuit

The clock generation circuit generates the clock to be supplied to the CPU and peripheral hardware.
The following two types of system clock oscillation circuits are available.

(1) Main system clock oscillation circuit

This circuit oscillates a frequency of 1.0 to 10.0 MHz. Oscillation can be stopped by executing the STOP instruction or by setting the processor clock control register (PCC).

(2) Subsystem clock oscillation circuit

This circuit oscillates a frequency of 32.768 kHz. Oscillation cannot be stopped. When the subsystem clock oscillation circuit is not used, it can be set by using the processor clock control register (PCC) that the internal feedback resistor is not used, so that the power dissipation in the STOP mode can be reduced.

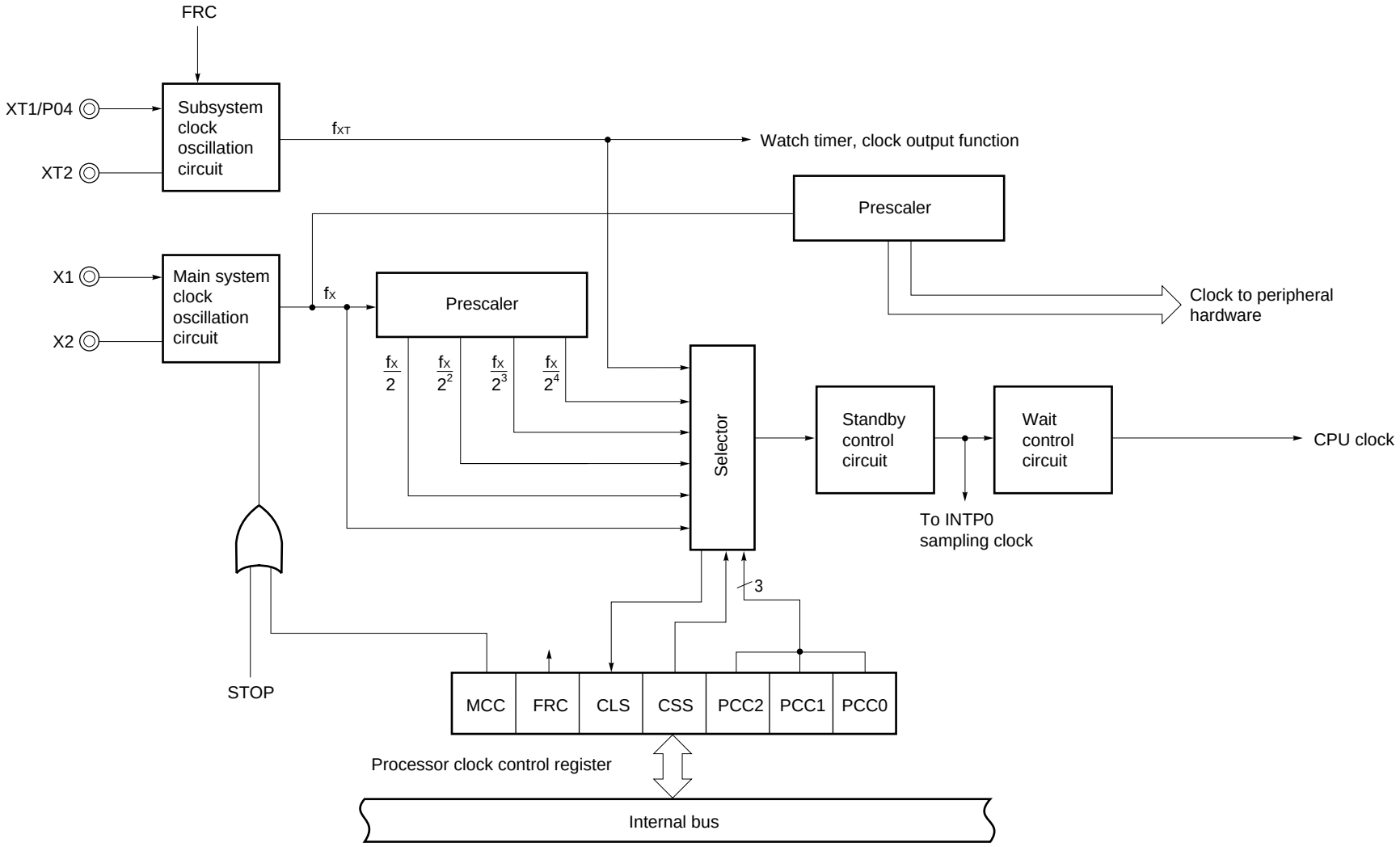
5.2 Configuration of Clock Generation Circuit

The clock generation circuit consists of the following hardware:

Table 5-1. Configuration of Clock Generation Circuit

Item	Configuration
Control register	Processor clock control register (PCC)
Oscillation circuit	Main system clock oscillation circuit Subsystem clock oscillation circuit

Figure 5-1. Block Diagram of Clock Generation Circuit



5.3 Register Controlling Clock Generation Circuit

The clock generation circuit is controlled by the processor clock control register (PCC). This register selects the CPU clock and division ratio, starts/stops the operation of the main system clock oscillation circuit, and sets whether the internal feedback resistor of the subsystem clock oscillation circuit is used or not.

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

This register is set to 04H when the $\overline{\text{RESET}}$ signal is input.

Figure 5-2. Feedback Resistor of Subsystem Clock

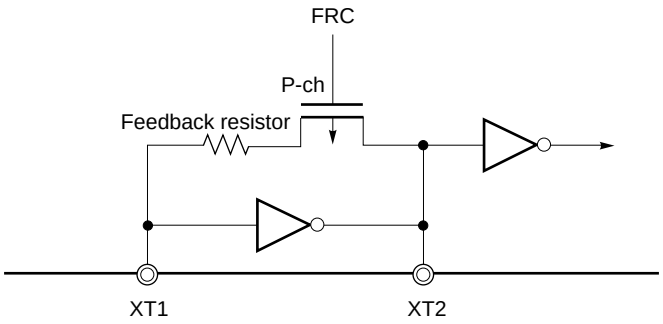


Figure 5-3. Format of Processor Clock Control Register

Symbol	<7>	<6>	<5>	<4>	3	2	1	0	Address	After reset	R/W
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0	FFFBH	04H	R/W[Note 1]

R/W

CSS	PCC2	PCC1	PCC0	Selects CPU clock (f_{CPU})
0	0	0	0	f_X
	0	0	1	$f_X/2$
	0	1	0	$f_X/2^2$
	0	1	1	$f_X/2^3$
	1	0	0	$f_X/2^4$
1	0	0	0	f_{XT}
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Others				Setting prohibited

R

CLS	Status of CPU clock
0	Main system clock
1	Subsystem clock

R/W

FRC	Selects feedback resistor of subsystem clock
0	Uses internal feedback resistor
1	Does not use internal feedback resistor

R/W

MCC	Controls oscillation of main system clock[Note 2]
0	Enables oscillation
1	Stops oscillation

Notes 1. Bit 5 is a read-only bit.

2. To stop oscillation of the main system clock while the CPU operates on the subsystem clock, use MCC. Do not use the STOP instruction.

Caution Be sure to clear bit 3 to 0.

Remarks 1. f_X : main system clock oscillation frequency

2. f_{XT} : subsystem clock oscillation frequency

The fastest instruction of the μPD78014H Subseries is executed by in four CPU clocks. The relationship between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in Table 5-2.

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

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $4/f_{CPU}$
f_X	0.4 μs
$f_X/2$	0.8 μs
$f_X/2^2$	1.6 μs
$f_X/2^3$	3.2 μs
$f_X/2^4$	6.4 μs
f_{XT}	122 μs

f_X = 10.0 MHz, f_{XT} = 32.768 kHz

f_X : Main system clock oscillation frequency

f_{XT} : Subsystem clock oscillation frequency

5.4 System Clock Oscillation Circuits

5.4.1 Main system clock oscillation circuit

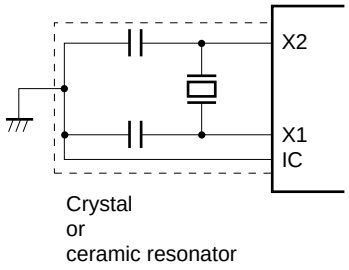
The main system clock oscillation circuit is oscillated by the crystal or ceramic resonator (10.0 MHz TYP.) connected across the X1 and X2 pins.

An external clock can also be input to the circuit. In this case, input the clock signal to the X1 pin, and input the reversed signal to the X2 pin.

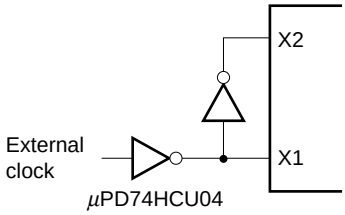
Figure 5-4 shows the external circuit of the main system clock oscillation circuit.

Figure 5-4. External Circuit of Main System Clock Oscillation Circuit

(a) Crystal or ceramic oscillation



(b) External clock



Caution Do not execute STOP instruction nor set MCC (bit 7 of processor clock control register (PCC)) to 1. If the STOP instruction is executed or MCC is set to 1, X-in system clock is stopped and the X2 pin is pulled up to V_{DD}.

5.4.2 Subsystem clock oscillation circuit

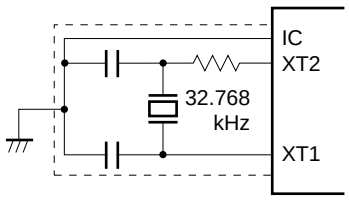
The subsystem clock oscillation circuit is oscillated by the crystal resonator connected across the XT1 and XT2 pins (32.768 kHz TYP.).

An external clock can also be input to the circuit. In this case, input the clock signal to the XT1 pin, and input the reversed signal to the XT2 pin.

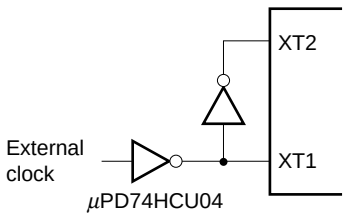
Figure 5-5 shows the external circuit of the subsystem clock oscillation circuit.

Figure 5-5. External Circuit of Subsystem Clock Oscillation Circuit

(a) Crystal oscillation



(b) External clock



Refer to **Cautions** on the following pages.

Caution 1. When using the main system clock or subsystem clock oscillator circuit, to avoid influence of wiring capacity, etc. wire the portion enclosed by broken line in Figures 5-4 and 5-5 as follows:

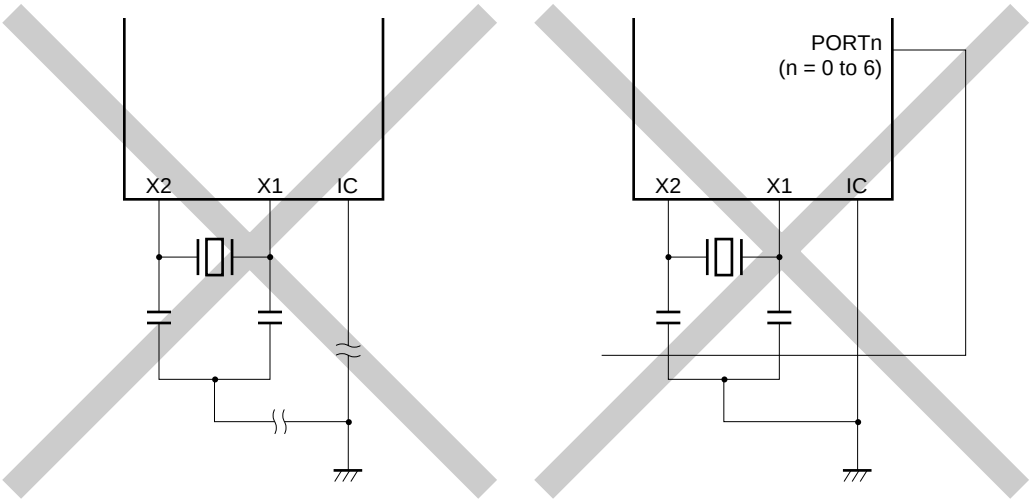
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with any other signal lines. Do not route the wiring in the vicinity of a line through which a high alternating current flows.**
- **Always keep the ground of the capacitor of the oscillation circuit at the same potential as V_{SS}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not extract a signal from the oscillation circuit.**

Note that the amplification factor of the subsystem clock oscillation circuit is kept low to reduce the current dissipation.

Figure 5-6 shows incorrect examples of resonator connection.

Figure 5-6. Incorrect Examples of Resonator Connection (1/2)

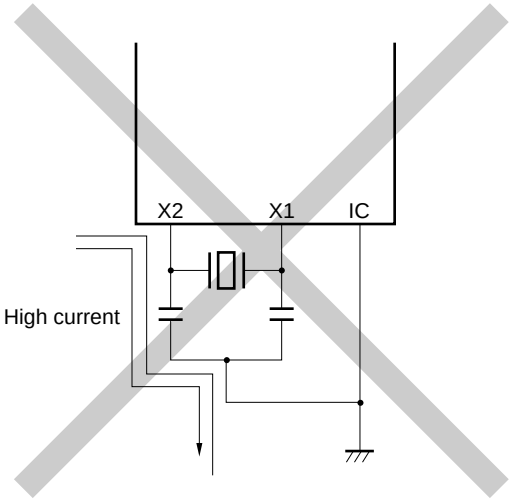
(a) Too long wiring **(b) Crossed signal line**



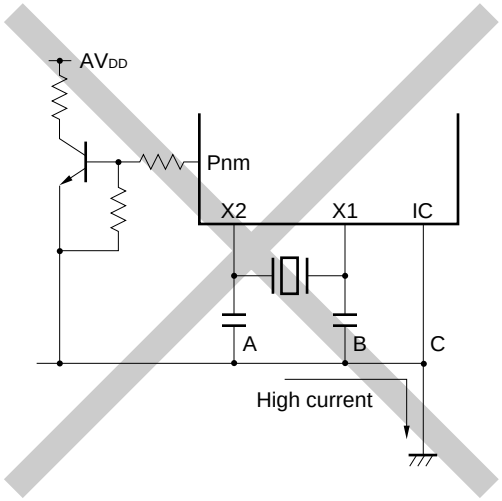
Remark X1 and X2 in this figure should be XT1 and XT2 when the subsystem clock is used. Connect a resistor to XT2 in series.

Figure 5-6. Incorrect Examples of Resonator Connection (2/2)

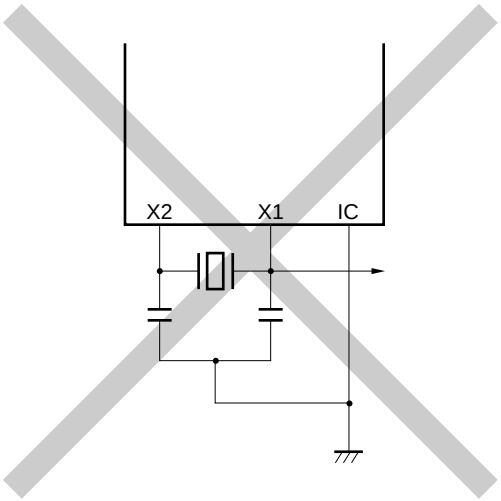
(c) Wiring near high alternating current



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



(e) Signal are fetched



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

Caution 2. When XT2 and X1 are wired in parallel, the cross-talk noise of X1 may increase with XT2, resulting in malfunctioning. To prevent that from occurring, it is recommended to avoid wiring XT2 and X1 in parallel and to connect the IC pin between XT2 and X1 directly to V_{SS}.

5.4.3 Divider circuit

The divider circuit divides the output of the main system clock oscillation circuit (f_X) to generate various clocks.

5.4.4 When subsystem clock is not used

When the subsystem clock is not necessary for a power-saving operation or watch operation, handle the XT1 and XT2 pins as follows:

XT1: Connect to V_{DD}
XT2: Leave open

In this status, however, a tiny amount of current leaks via the internal feedback resistor of the subsystem clock oscillation circuit when the main system clock is stopped. To suppress this leakage current, it is possible to remove the above internal feedback resistor, by using the bit 6 (FRC) of the processor clock control register (PCC). At this time, process the XT1 and XT2 pins in the same manner as above.

5.5 Operation of Clock Generation Circuit

The clock generation circuit generates the following clocks and control the operation modes of the CPU, such as the standby mode:

- Main system clock f_X
- Subsystem clock f_{XT}
- CPU clock f_{CPU}
- Clock to peripheral hardware

The operation of the clock generation circuit is determined by the processor clock control register (PCC), as follows:

(a) The slowest mode (6.4 μs: at 10.0-MHz operation) of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 04H). While a low level is input to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Five types of CPU clocks (0.4 μs, 0.8 μs, 1.6 μs, 3.2 μs, and 6.4 μs: at 10.0-MHz operation) can be selected by the PCC setting with the main system clock selected.

(c) Two standby modes, STOP and HALT, can be used when the main system clock is selected. In a system where the subsystem clock is not used, the current consumption in the STOP mode can be further reduced by specifying not to use the internal feedback resistor by using the bit 6 (FRC) of PCC.

(d) The subsystem clock can be selected by PCC and the microcontroller can operate with a low current consumption (122 μs: at 32.768-kHz operation).

(e) Oscillation of the main system clock can be stopped by PCC with the subsystem clock selected. Moreover, the HALT mode can be used. However, the STOP mode cannot be used (oscillation of the subsystem clock cannot be stopped).

(f) The clock to the peripheral hardware is supplied by dividing the main system clock. However, the subsystem clock is supplied to the watch timer and clock output function only. Therefore, the watch function and clock output function can be continuously used even in the standby status. The other peripheral hardware is stopped when the main system clock is stopped because the peripheral hardware operates on the main system clock (except, however, the external clock input operation).

5.5.1 Operation of main system clock

When the main system clock is used (when bit 5 (CLS) of the processor clock control register (PCC) is 0), the following operations are performed by the PCC setting:

(a) Because the operation guaranteeing instruction execution speed differs depending on the supply voltage, the minimum instruction execution time can be changed by using the bits 0 to 2 (PCC0 to PCC2) of PCC.

(b) Oscillation of the main system clock is not stopped even when bit 7 (MCC) of PCC is set to 1 when the microcontroller operates on the main system clock. When bit 4 (CSS) of PCC is later set to 1 and then subsystem clock is selected (CLS = 1), oscillation of the main system clock is stopped (refer to **Figure 5-7**).

Figure 5-7. Stopping Main System Clock (1/2)

(a) When CSS is set and then MCC is set during main system clock operation

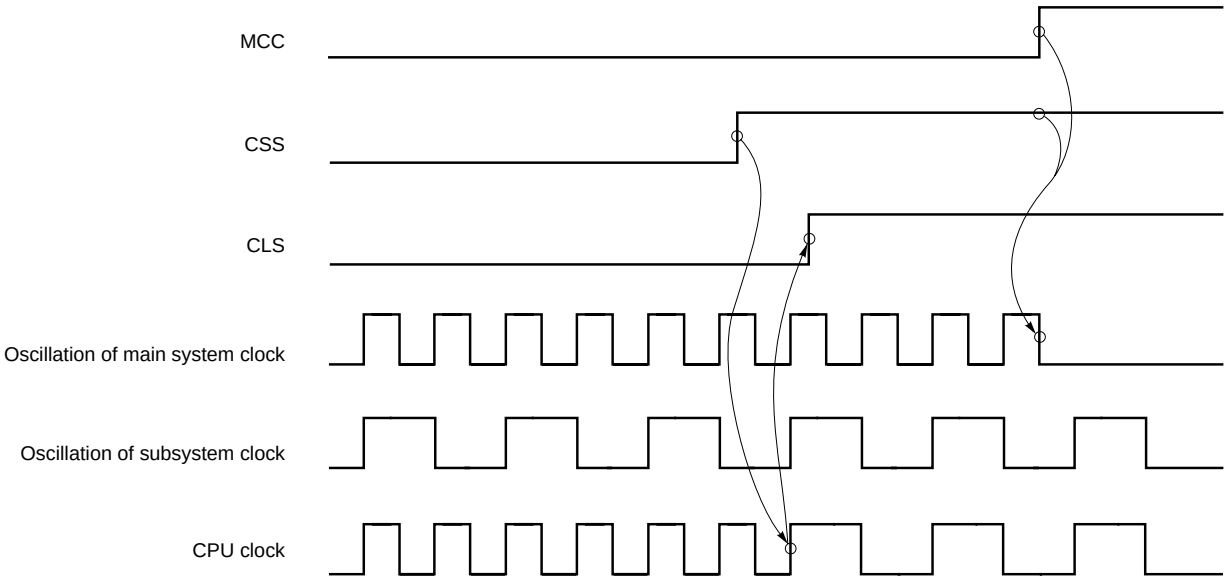
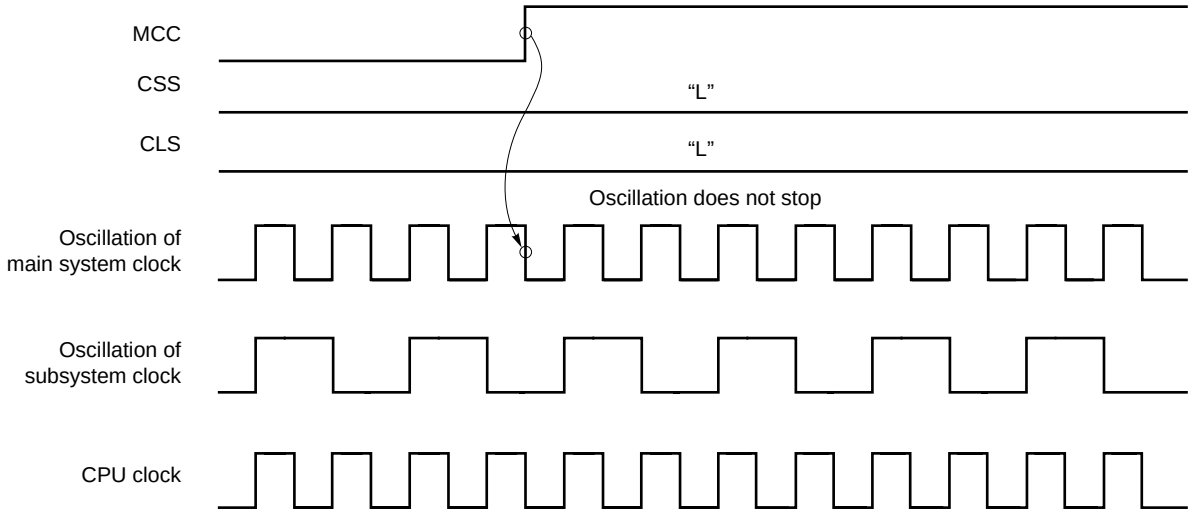
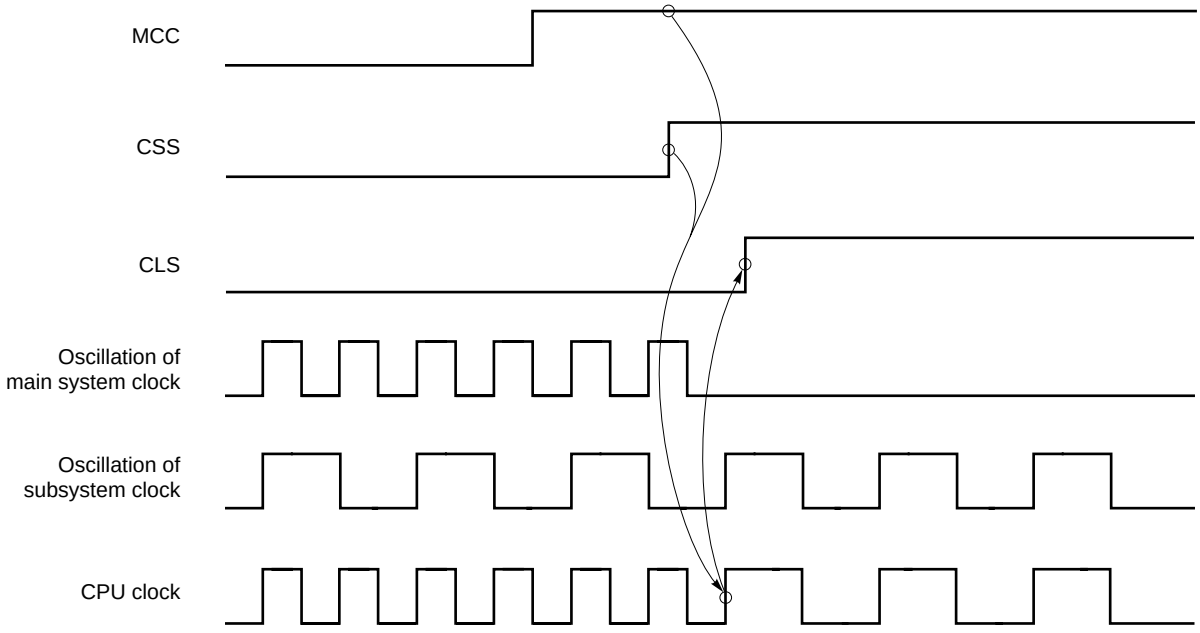


Figure 5-7. Stopping Main System Clock (2/2)

(b) When MCC is set during main system clock operation



(c) When MCC is set and then CSS is set during main system clock oscillation



5.5.2 Operation of subsystem clock

When the subsystem clock is used (when bit 5 (CLS) of the processor clock control register (PCC) is 1), the following operations are performed:

(a) Minimum instruction execution time is held constant (122 μs: at 32.768-kHz operation) regardless of the setting of bits 0 to 2 (PCC0 to PCC2) of PCC.

(b) The watchdog timer stops counting.

Caution Do not execute the STOP instruction during subsystem clock operation.

5.6 Changing Setting of System Clock and CPU Clock

5.6.1 Time required for switching between system clock and CPU clock

The system clock or CPU clock can be selected by using bits 0 to 2 (PCC0 to PCC2) and bit 4 (CSS) of the processor clock control register (PCC).

Actually, the specified clock is not selected immediately after the setting of PCC has been changed, and the old clock is used for the duration of several instructions after that (refer to **Table 5-3**).

Whether the system operates on the main system clock or subsystem clock can be learned by using bit 5 (CLS) of PCC.

Table 5-3. Maximum Time Required for Switching CPU Clock

Set Value before Switching				Set Value after Switching																							
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 instructions				16 instructions				16 instructions				16 instructions				$f_X/4f_{XT}$ instructions (77 instructions)			
	0	0	1	8 instructions								8 instructions				8 instructions				8 instructions				$f_X/8f_{XT}$ instructions (39 instructions)			
	0	1	0	4 instructions				4 instructions								4 instructions				4 instructions				$f_X/16f_{XT}$ instructions (20 instructions)			
	0	1	1	2 instructions				2 instructions				2 instructions								2 instructions				$f_X/32f_{XT}$ instructions (10 instructions)			
	1	0	0	1 instruction				1 instruction				1 instruction				1 instruction								$f_X/64f_{XT}$ instructions (5 instructions)			
1	×	×	×	1 instruction				1 instruction				1 instruction				1 instruction				1 instruction							

Caution Do not select the division ratio of the CPU clock (PCC0 to PCC2) and switch the main system clock to the subsystem clock (CSS 0 → 1) at the same time.
However, the division ratio of the CPU clock (PCC0 to PCC2) can be selected and the subsystem clock can be switched to the main system clock at the same time (CSS 1 → 0).

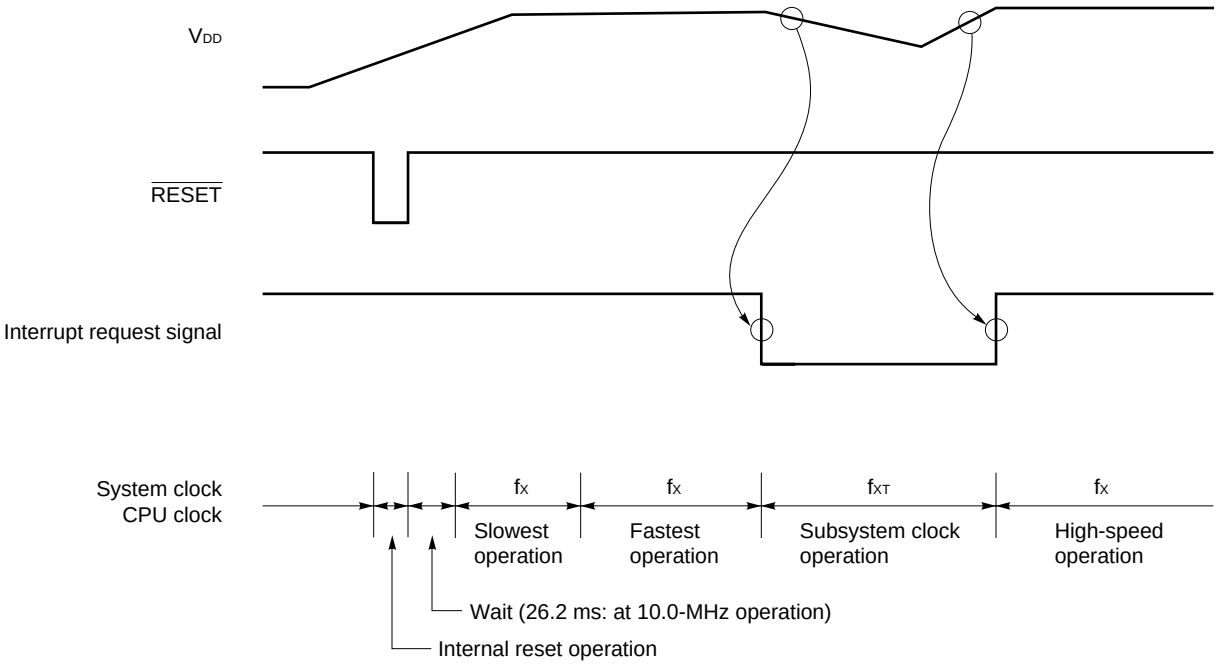
Remarks 1. One instruction is the minimum instruction execution time of the CPU clock before switching.

2. ( ): f_X = 10.0 MHz or f_{XT} = 32.768 kHz

5.6.2 Switching between system clock and CPU clock

The following figure illustrates how the system clock is switched to the CPU clock or vice versa.

Figure 5-8. Switching between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. The effect of resetting is released when the $\overline{\text{RESET}}$ pin is later made high, and the main system clock starts oscillating. At this time, the time during which oscillation stabilizes ($2^{18}/f_X$) is automatically secured.
After that, the CPU starts instruction execution at the slowest speed of the main system clock (6.4 μs: at 10.0-MHz operation).

<2> After the time during which the V_{DD} voltage rises to the level at which the CPU can operate at the highest speed has elapsed, processor clock control register (PCC) is rewritten so that the highest speed can be selected.

<3> A drop of the V_{DD} voltage is detected by using an interrupt request signal. If this happens, the subsystem clock is selected (at this time, the subsystem clock must be in the oscillation stabilization status).

<4> The recovery of V_{DD} voltage to the original level is detected by using an interrupt request signal, 0 is set to bit 7 of PCC (MCC), and oscillation of the main system clock is started. After the time required for oscillation to stabilize has elapsed, PCC is rewritten, so that the highest speed can be selected.

Caution To select the main system clock again when the system operates on the subsystem clock with the main system clock stopped, be sure to secure the oscillation stabilization time by program, and then select the main system clock.

[MEMO]

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

6.1 Overview of μPD78014H Subseries On-chip Timer

This chapter explains the 16-bit timer/event counter. The following is an overview of timers and related functions that are provided in the μPD78014H Subseries.

(1) 16-bit timer/event counter (TM0)

This timer can be used as an interval timer, for PWM output, for pulse width measurement (infrared remote controller signal reception function), as an external event counter, and for output of square waves of any frequency.

(2) 8-bit timer/event counters (TM1 and TM2)

These counters can be used as interval timers, external event counters, and for output of square waves of any frequency. Moreover, the two 8-bit timer/event counters can be used in combination as a 16-bit time/event counter (refer to **CHAPTER 7 8-BIT TIMER/EVENT COUNTER**).

(3) Watch timer (TM3)

This timer can be used to set a flag every 0.5 second or to generate an interrupt request at any time intervals set in advance at the same time (refer to **CHAPTER 8 WATCH TIMER**).

(4) Watchdog timer (WDTM)

The watchdog timer can also be used to generate a non-maskable interrupt request, maskable interrupt request, or $\overline{\text{RESET}}$ signal at any time intervals set in advance (refer to **CHAPTER 9 WATCHDOG TIMER**).

(5) Clock output control circuit

This circuit supplies a clock obtained by dividing the main system clock, and the subsystem clock to other devices (refer to **CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT**).

(6) Buzzer output control circuit

This circuit outputs a buzzer frequency that is obtained by dividing the main system clock (refer to **CHAPTER 11 BUZZER OUTPUT CONTROL CIRCUIT**).

Table 6-1. Operations of Timer/Event Counters

		16-bit Timer/ Event Counter	8-bit Timer/ Event Counter	Watch Timer	Watchdog Timer
Operation mode	Interval timer	1 channel	2 channels	1 channel[Note 1]	1 channel[Note 2]
	External event counter	○	○	–	–
Function	Timer output	○	○	–	–
	PWM output	○	–	–	–
	Pulse width measurement	○	–	–	–
	Square wave output	○	○	–	–
	Interrupt request	○	○	○	○
	Test input	–	–	○	–

Notes 1. Watch timer can be used as a watch timer and an interval timer at the same time.

2. Watchdog timer has a watchdog timer and interval timer functions. Select one of them.

6.2 Functions of 16-Bit Timer/Event Counter

The 16-bit timer/event counter (TM0) has the following functions:

- Interval timer
- PWM output
- Pulse width measurement
- External event counter
- Square wave output

The PWM output and pulse width measurement functions can be used at the same time.

(1) Interval timer

When the 16-bit timer/event counter is used as an interval timer, it generates an interrupt request at any time intervals set in advance.

Table 6-2. Interval Time of 16-Bit Timer/Event Counter

Minimum Interval Time	Maximum Interval Time	Resolution
2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
$2^2 \times 1/f_X$ (400 ns)	$2^{17} \times 1/f_X$ (13.1 ms)	$2 \times 1/f_X$ (200 ns)
$2^3 \times 1/f_X$ (800 ns)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^2 \times 1/f_X$ (400 ns)
$2^4 \times 1/f_X$ (1.6 μs)	$2^{19} \times 1/f_X$ (52.4 ms)	$2^3 \times 1/f_X$ (800 ns)

Remarks 1. f_X : main system clock oscillation frequency

2. ( ): f_X = 10.0-MHz operation

(2) PWM output

The 16-bit timer/event counter can be used for PWM output with a resolution of 14 bits.

(3) Pulse width measurement

The 16-bit timer/event counter can be used to measure the pulse width of an externally input signal.

(4) External event counter

The number of pulses of an externally input signal can be measured.

(5) Square wave output

A square wave of any frequency can be output.

Table 6-3. Square Wave Output Range of 16-Bit Timer/Event Counter

Minimum Pulse Width	Maximum Pulse Width	Resolution
2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
$2^2 \times 1/f_X$ (400 ns)	$2^{17} \times 1/f_X$ (13.1 ms)	$2 \times 1/f_X$ (200 ns)
$2^3 \times 1/f_X$ (800 ns)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^2 \times 1/f_X$ (400 ns)
$2^4 \times 1/f_X$ (1.6 μs)	$2^{19} \times 1/f_X$ (52.4 ms)	$2^3 \times 1/f_X$ (800 ns)

Remarks 1. f_X : main system clock oscillation frequency

2. ( ): f_X = 10.0-MHz operation

6.3 Configuration of 16-Bit Timer/Event Counter

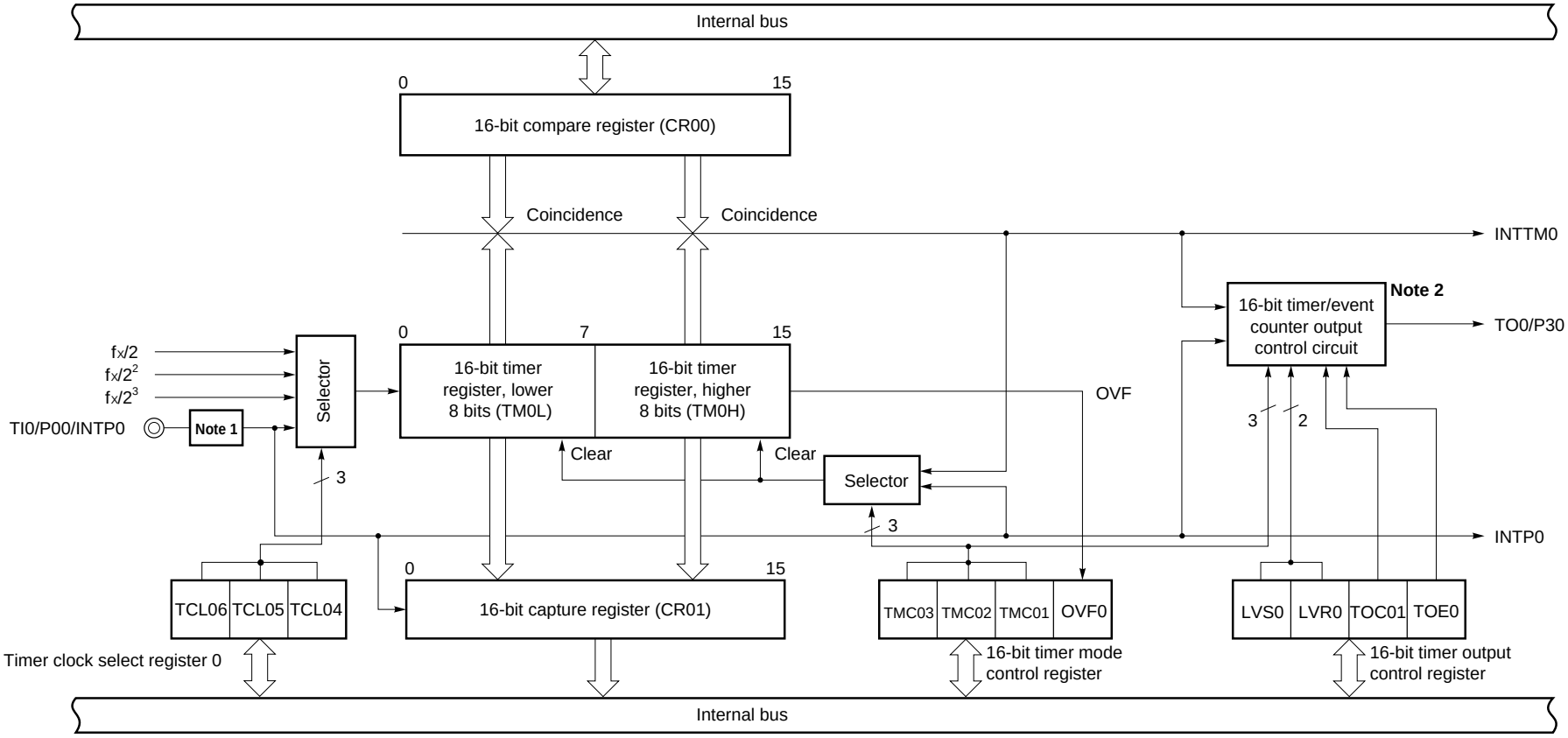
The 16-bit timer/event counter consists of the following hardware:

Table 6-4. Configuration of 16-Bit Timer/Event Counter

Item	Configuration
Timer register	16 bits × 1 (TM0)
Register	Compare register : 16 bits × 1 (CR00) Capture register : 16 bits × 1 (CR01)
Timer output	1 (TO0)
Control register	Timer clock select register 0 (TCL0) 16-bit timer mode control register (TMC0) 16-bit timer output control register (TOC0) Port mode register 3 (PM3) External interrupt mode register (INTM0) Sampling clock select register (SCS)[Note]

Note Refer to **Figure 15-1 Basic Configuration of Interrupt Function**.

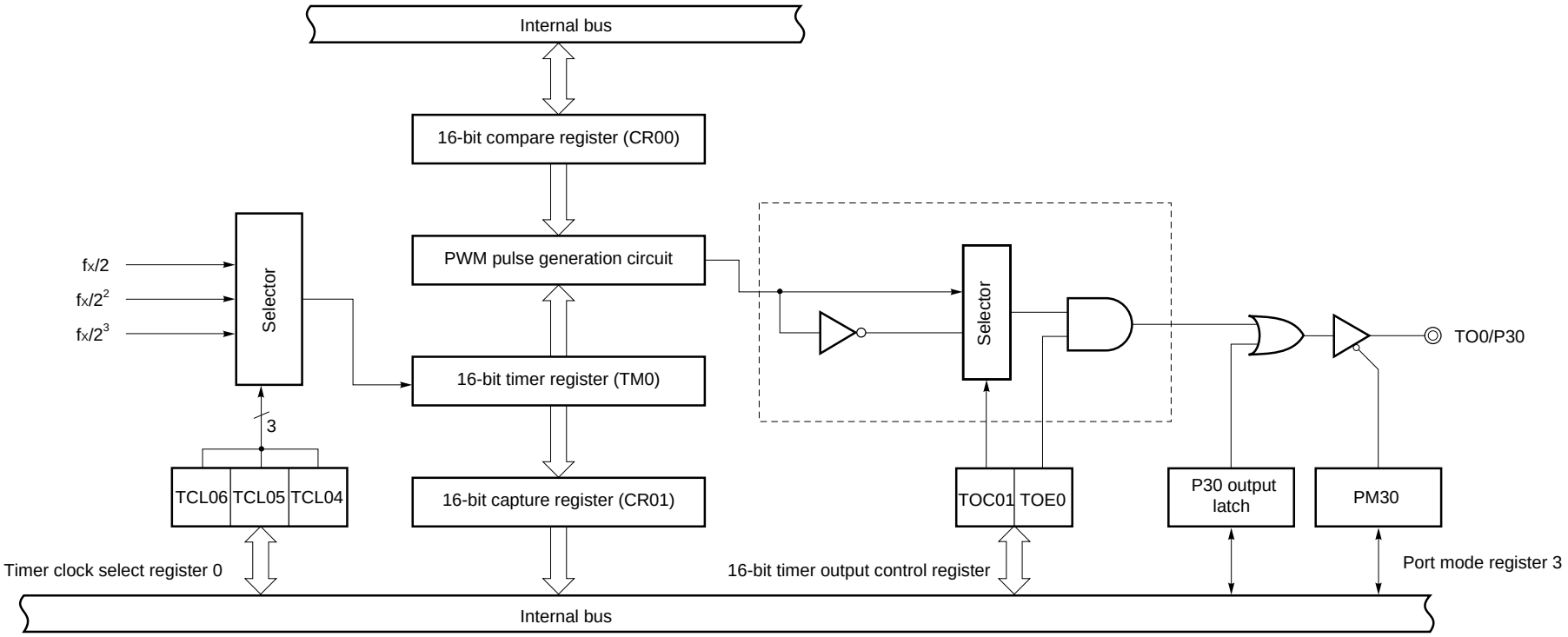
Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter (Timer Mode)



Notes **1.** Edge detection circuit

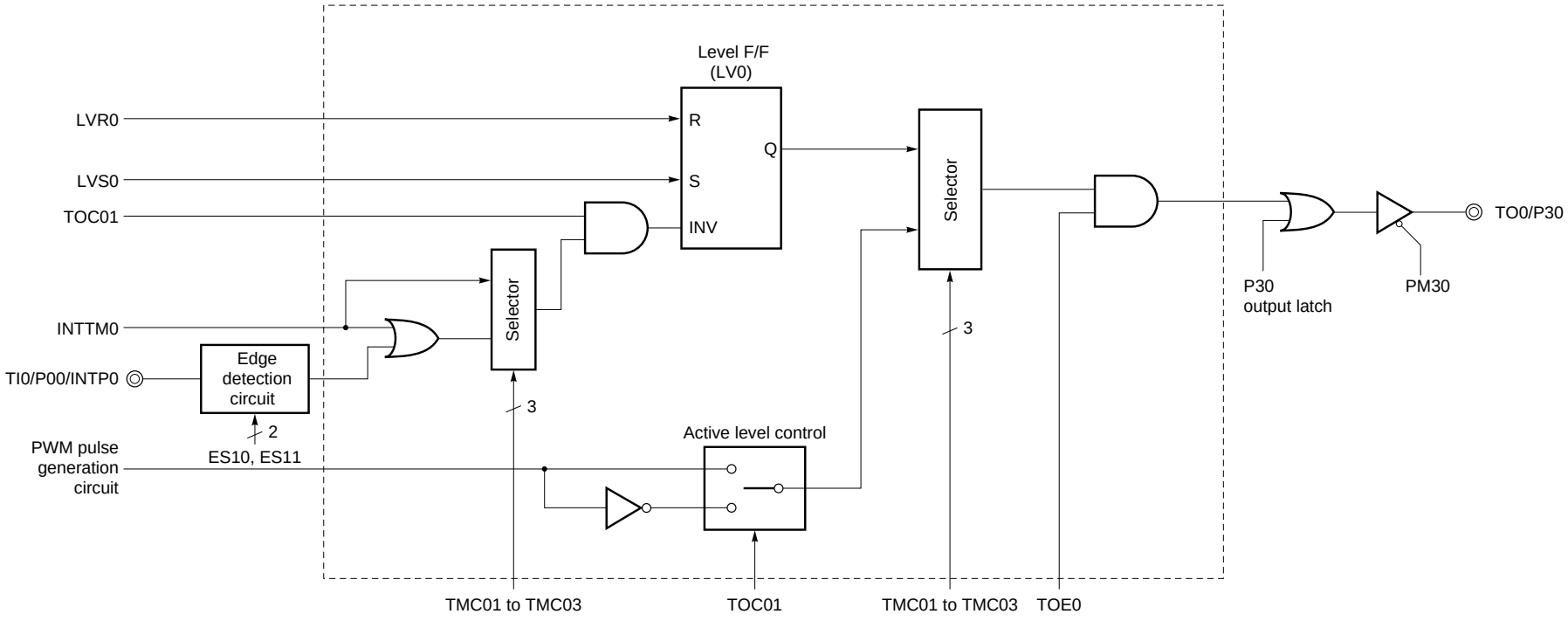
2. For the configuration of the output control circuit of the 16-bit timer/event counter, refer to **Figure 6-3**.

Figure 6-2. Block Diagram of 16-Bit Timer/Event Counter (PWM Mode)



Remark The output control circuit is shown enclosed by dotted line.

Figure 6-3. Block Diagram of 16-Bit Timer/Event Counter Output Control Circuit



Remark The output control circuit is shown enclosed by dotted line.

(1) 16-bit compare register (CR00)

This register always compares its set value in CR00 with the count value of the 16-bit timer register (TM0). When the two values coincide, an interrupt request (INTTM0) is generated.

When TM0 is set as an interval timer, this register is used to hold interval time. When TM0 is set for PWM output, this register is used to set a pulse width.

CR00 is set by a 16-bit memory manipulation instruction in a range of 0001H to FFFFH.

The contents of this register become undefined when the $\overline{\text{RESET}}$ signal is input.

Cautions 1. Set data of PWM (14 bits) to the higher 14 bits of CR00. At this time, set the lower 2 bits to 00.

2. Set CR00 to any value other than 0000H. Thus, one pulse will not be counted when the timer is used as an event counter.

3. If the new value of CR00 is less than the value of the 16-bit timer register (TM0), TM0 continues counting, overflows, and counts again from 0. If the new value of CR00 is less than the previous value, it is necessary to restart the timer.

(2) 16-bit capture register (CR01)

This 16-bit register captures the contents of the 16-bit timer register (TM0).

The capture trigger is the valid edge input to the INTP0/TI0 pin. The valid edge of INTP0 is set by the external interrupt mode register (INTM0).

CR01 is read by a 16-bit memory manipulation instruction.

The contents of this register become undefined when the $\overline{\text{RESET}}$ signal is input.

Caution If the valid edge of the TI0/P00 pin is input while CR01 is read, CR01 does not perform the capture operation but holds data. However, the interrupt request flag (RIF0) is set by detection of the valid edge.

(3) 16-bit timer register (TM0)

This 16-bit register counts the number of count pulses.

The value of TM0 can be read by a 16-bit memory manipulation instruction.

This register is initialized to 0000H when the $\overline{\text{RESET}}$ signal is input.

Caution Because the value of TM0 is read via CR01, the value of CR01 is destroyed.

6.4 Registers Controlling 16-Bit Timer/Event Counter

The following six types of registers control the 16-bit timer/event counter:

- Timer clock select register 0 (TCL0)
- 16-bit timer mode control register (TMC0)
- 16-bit timer output control register (TOC0)
- Port mode register 3 (PM3)
- External interrupt mode register (INTM0)
- Sampling clock select register (SCS)

(1) Timer clock select register 0 (TCL0) (refer to Figure 6-4)

This register sets the count clock of the 16-bit timer register.

TCL0 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Remark TCL0 also has a function to set the clock for PCL output, in addition to the function to set the count clock of the 16-bit timer register.

(2) 16-bit timer mode control register (TMC0) (refer to Figure 6-5)

This register sets an operation mode of the 16-bit timer, clear mode of the 16-bit timer register, and output timing, and detects an overflow.

TMC0 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Caution The 16-bit timer register (TM0) starts operation when a value other than 0, 0, 0 is set to TMC01 to TMC03 (operation stop mode). To stop the operation, set TMC01 to TMC03 to 0, 0, 0.

Figure 6-4. Format of Timer Clock Select Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TCL0	CLOE	TCL06	TCL05	TCL04	TCL03	TCL02	TCL01	TCL00	FF40H	00H	R/W

TCL03	TCL02	TCL01	TCL00	Selects clock of PCL output
0	0	0	0	f_{XT} (32.768 kHz)
0	1	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	0	$f_X/2^4$ (625 kHz)
1	0	0	1	$f_X/2^5$ (313 kHz)
1	0	1	0	$f_X/2^6$ (156 kHz)
1	0	1	1	$f_X/2^7$ (78.1 kHz)
1	1	0	0	$f_X/2^8$ (39.1 kHz)
Others				Setting prohibited

TCL06	TCL05	TCL04	Selects count clock of 16-bit timer register
0	0	0	TI0 (valid edge can be specified)
0	1	0	$f_X/2$ (5.0 MHz)
0	1	1	$f_X/2^2$ (2.5 MHz)
0	0	0	$f_X/2^3$ (1.25 MHz)
Others			Setting prohibited

CLOE	Controls PCL output
0	Disables output
1	Enables output

Cautions 1. The valid edge of the TI0/INTP0 pin is set by the external interrupt mode register (INTM0). The frequency of a sampling clock is selected by the sampling clock select register (SCS).

2. To enable PCL output, set TCL00 to TCL03, and then set CLOE to 1 by using a 1-bit memory manipulation instruction.

3. Read the count value from TM0, not from the 16-bit capture register (CR01), when TI0 is used as the count clock of TM0.

4. To write data other than that already written to TCL0, stop the timer operation once.

Remarks 1. f_X : main system clock oscillation frequency

2. f_{XT} : subsystem clock oscillation frequency

3. TI0 : input pin of 16-bit timer/event counter

4. TM0 : 16-bit timer register

5. ( ) : at f_X = 10.0-MHz or f_{XT} = 32.768-kHz operation

6. For PCL, refer to **CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT**.

Figure 6-5. Format of 16-Bit Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC0	0	0	0	0	TMC03	TMC02	TMC01	OVF0	FF48H	00H	R/W

OVF0	Detects overflow in 16-bit timer register
0	Overflow does not occur
1	Overflow occurs

TMC03	TMC02	TMC01	Selects operation mode or clear mode	Selects timing of TO0 output	Generates interrupt request
0	0	0	Stops operation (TM0 is cleared to 0)	Not affected	Not generated
0	0	1	PWM mode (free running)	PWM pulse output	Generated when TM0 coincides with CR00
0	1	0	Free running mode	Coincidence between TM0 and CR00	
0	1	1		Coincidence between TM0 and CR00 or valid edge of TI0	
1	0	0	Clear and start at valid edge of TI0	Coincidence between TM0 and CR00	
1	0	1		Coincidence between TM0 and CR00 or valid edge of TI0	
1	1	0	Clear and start at coincidence between TM0 and CR00	Coincidence between TM0 and CR00	
1	1	1		Coincidence between TM0 and CR00 or valid edge of TI0	

Cautions 1. Before changing the clear mode and output timing of TO0, stop the timer operation (set TMC01 through TMC03 to 0, 0, 0).

2. The valid edge of the TI0/INTP0 pin is set by the external interrupt mode register (INTM0). The frequency of the sampling clock is selected by the sampling clock select register (SCS).

3. When using the PWM mode, set the PWM mode and then set data to CR00.

4. When a mode in which the 16-bit timer is cleared and started on coincidence between TM0 and CR00 is selected, the OVF0 flag is set to 1 when the value of TM0 changes from FFFFH to 0000H with FFFFH set to CR00.

Remark TO0 : output pin of 16-bit timer/event counter
TI0 : input pin of 16-bit timer/event counter
TM0 : 16-bit timer register
CR00 : 16-bit compare register

(3) 16-bit timer output control register (TOC0)

This register controls the operation of the 16-bit timer/event counter output control circuit. It sets/resets an R-S flip-flop (LV0), sets an active level in the PWM mode, enables/disables inversion of the output in a mode other than PWM mode, and sets a data output mode.

TOC0 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 6-6. Format of 16-Bit Timer Output Control Register

Symbol	7	6	5	4	<3>	<2>	1	<0>	Address	After reset	R/W
TOC0	0	0	0	0	LVS0	LVR0	TOC01	TOE0	FF4EH	00H	R/W

TOE0	Controls output of 16-bit timer/event counter
0	Disables output (port mode)
1	Enables output

TOC01	PWM mode	Other than PWM mode
	Selects active level	Controls timer output F/F
0	High active	Disables inversion
1	Low active	Enables inversion

LVS0	LVR0	Sets status of timer output F/F of 16-bit timer/event counter
0	0	Not affected
0	1	Resets timer output F/F to 0
1	0	Sets timer output F/F to 1
1	1	Setting prohibited

Cautions 1. Be sure to stop the timer operation before setting TOC0.

2. 0 is read from LVS0 and LVR0 after data has been set to these bits.

(4) Port mode register 3 (PM3)

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

When the P30/TO0 pin is used as a timer output pin, set 0 to the PM30 bit of this register and the output latch of the P30 pin.

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

This register is set to FFH when the $\overline{\text{RESET}}$ signal is input.

Figure 6-7. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	Selects I/O mode of P3n pin (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(5) External interrupt mode register (INTM0)

This register sets the valid edges of the INTP0 to INTP2 pins.

INTM0 is set by an 8-bit memory manipulation register.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Remarks 1. The INTP0 pin is multiplexed with TI0/P00.

2. INTP3 is fixed to the falling edge.

Figure 6-8. Format of External Interrupt Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES31	ES30	ES21	ES20	ES11	ES10	0	0	FFECH	00H	R/W

ES11	ES10	Selects valid edge of INTP0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES21	ES20	Selects valid edge of INTP1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	Selects valid edge of INTP2
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Caution Before setting the valid edge of the INP0/TI0/P00 pin, clear the bits 1 through 3 (TMC01 through TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0, and stop the timer operation.

(6) Sampling clock select register (SCS)

This register sets the clock with which the valid edge input to INTP0 is to be sampled. When a remote controller signal is eliminated by using INTP0, digital noise is eliminated by sampling clock.

SCS is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 6-9. Format of Sampling Clock Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	Selects sampling clock of INTP0
0	0	$f_X/2^{N+1}$
0	1	Setting prohibited
1	0	$f_X/2^6$ (156 kHz)
1	1	$f_X/2^7$ (78.1 kHz)

Caution $f_X/2^{N+1}$ is the clock supplied to the CPU, and $f_X/2^6$ and $f_X/2^7$ are the clocks supplied to the hardware. $f_X/2^{N+1}$ is stopped in the HALT mode.

Remarks 1. N : value (N = 0 to 4) set to bits 0 to 2 (PCC0 to PCC2) of processor clock control register (PCC)

2. f_X : main system clock oscillation frequency

3. ( ): at f_X = 10.0-MHz operation

6.5 Operation of 16-Bit Timer/Event Counter

6.5.1 Operation as interval timer

The 16-bit timer/event counter operates as an interval timer when bits 2 and 3 (TMC02 and TMC03) of the 16-bit timer mode control register (TMC0) are set to 1, 1, and repeatedly generates an interrupt request at time intervals specified by the count value set to the 16-bit compare register (CR00) in advance.

When the count value of the 16-bit timer register (TM0) coincides with the value set to CR00, the value of TM0 is cleared to 0 and TM0 continues counting. At the same time, an interrupt request signal (INTTM0) is generated.

The count clock of the 16-bit timer/event counter can be selected by bits 4 to 6 (TCL04 to TCL06) of the timer clock select register 0 (TCL0).

For the operation when the value of compare register is changed during timer counter operation, refer to **6.6 (3) Operation after changing value of compare register during timer count operation**.

Figure 6-10. Configuration of Interval Timer

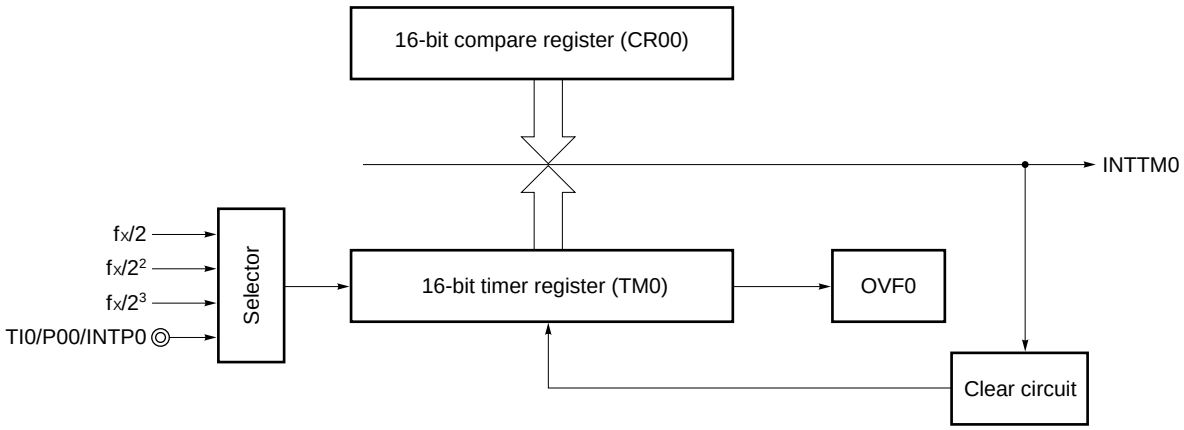
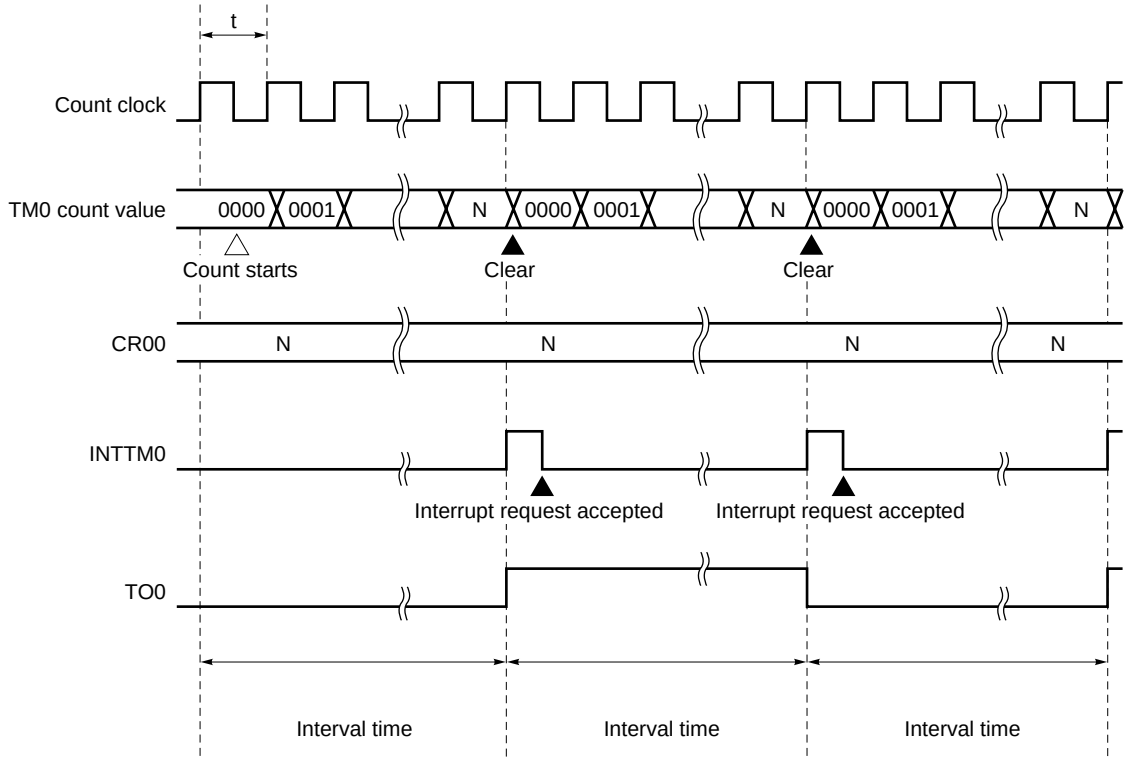


Figure 6-11. Interval Timer Operation Timing



Remark Interval time = (N + 1) x t: N = 0001H to FFFFH

Table 6-5. Interval Time of 16-Bit Timer/Event Counter

TCL06	TCL05	TCL04	Minimum Interval Time	Maximum Interval time	Resolution
0	0	0	$2 \times$ TI0 input cycle	$2^{16} \times$ TI0 input cycle	TI0 input edge cycle
0	1	0	$2^2 \times 1/fx$ (400 ns)	$2^{17} \times 1/fx$ (13.1 ms)	$2 \times 1/fx$ (200 ns)
0	1	1	$2^3 \times 1/fx$ (800 ns)	$2^{18} \times 1/fx$ (26.2 ms)	$2^2 \times 1/fx$ (400 ns)
1	0	0	$2^4 \times 1/fx$ (1.6 μs)	$2^{19} \times 1/fx$ (52.4 ms)	$2^3 \times 1/fx$ (800 ns)
Others			Setting prohibited		

Remarks **1.** f_X : main system clock oscillation frequency

2. TCL04 to TCL06 : bits 4 to 6 of timer clock selection register (TCL0).

3. ( ) : at f_X = 10.0-MHz operation

6.5.2 Operation as PWM output

The 16-bit timer/event counter performs PWM output when bits 1 to 3 (TMC01 to 03) of the 16-bit timer mode control register (TMC0) are set to 1, 0, 0, and outputs a pulse whose duty ratio is determined by the value set to the 16-bit compare register (CR00), from the TO0/P30 pin.

Set the active level width of the PWM pulse to the higher 14 bits of CR00. Select the active level by the bit 1 (TOC01) of the 16-bit timer output control register (TOC0).

The PWM pulse has a resolution of 14 bits. It can be converted into an analog voltage when integrated by an external low-pass filter (LPF). This pulse is created by using the basic cycle determined by $2^8/\Phi$ and subcycle determined by $2^{14}/\Phi$ in combination, and is designed to shorten the time constant of the external LPF. Count clock Φ can be selected by bits 4 to 6 (TCL04 to TCL06) of the timer clock select register 0 (TCL0).

PWM output can be enabled or disabled by the bit 0 (TOE0) of TOC0.

Cautions 1. Set CR00 after selecting the PWM operation mode.

2. Be sure to write 0 to the bits 0 and 1 of CR00.

3. When an external clock is input from the TI0/P00/INTP0 pin, do not select the PWM operation mode.

By integrating the PWM pulse with a 14-bit resolution by using an external low-pass filter, the pulse can be converted into an analog voltage which can be used for electronic tuning and D/A conversion.

The analog output voltage (V_{AN}) used for D/A conversion, whose configuration is shown in Figure 6-12, can be calculated by the following expression:

$$V_{AN} = V_{REF} \times \frac{\text{Value of compare register (CR00)}}{2^{16}}$$

where,

V_{REF}: reference voltage of external switching circuit

Figure 6-12. Example of Configuration of D/A Converter Using PWM Output

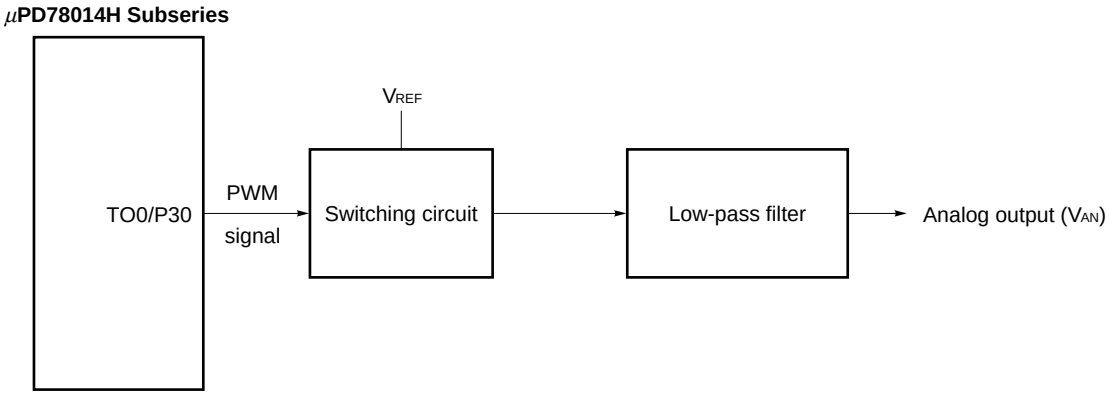
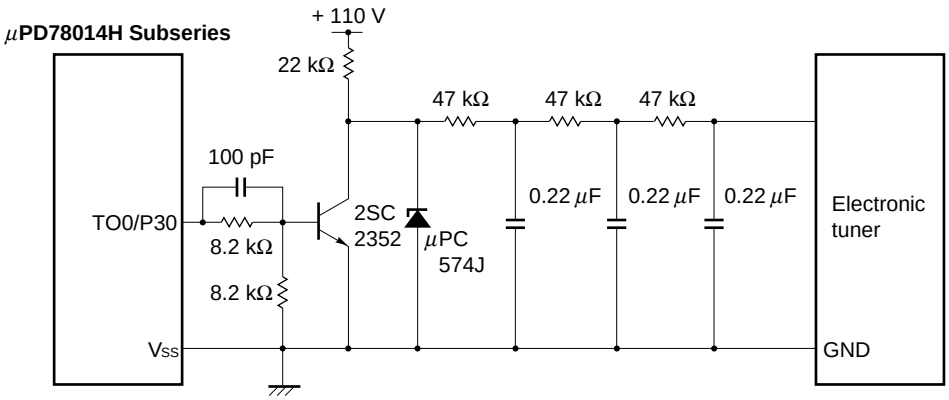


Figure 6-13 shows an example where the PWM output is converted into an analog voltage and applied for a TV tuner of voltage synthesizer type.

Figure 6-13. Example of Application Circuit (TV Tuner)



6.5.3 Operation as pulse width measurement

The 16-bit timer register (TM0) can be used to measure the pulse width of the signal input to the TI0/P00 pin.

Measurement is carried out in two ways. One is to measure the pulse width with the TM0 in the free running status, and the other is to measure the pulse width by restarting the timer in synchronization with the valid edge of the signal input to the TI0/P00 pin.

(1) Free running pulse width measurement

If an edge specified by the external interrupt mode register (INTM0) is input to the TI0/P00 pin while the 16-bit timer register (TM0) operates, the value of TM0 is captured to the 16-bit capture register (CR01), and an external interrupt request signal (INTP0) is set.

Three types of edges can be selected by bits 2 and 3 (ES10 and ES11) of INTM0: rising, falling, and both rising and falling edges.

To detect the valid edge, sampling is performed at the cycle selected by the sampling clock select register (SCS). The value of TM0 is not captured until the valid edge is detected two times. Consequently, noise that may be superimposed on a pulse with a short pulse width can be eliminated.

Figure 6-14. Configuration of Pulse Width Measurement by Free Running

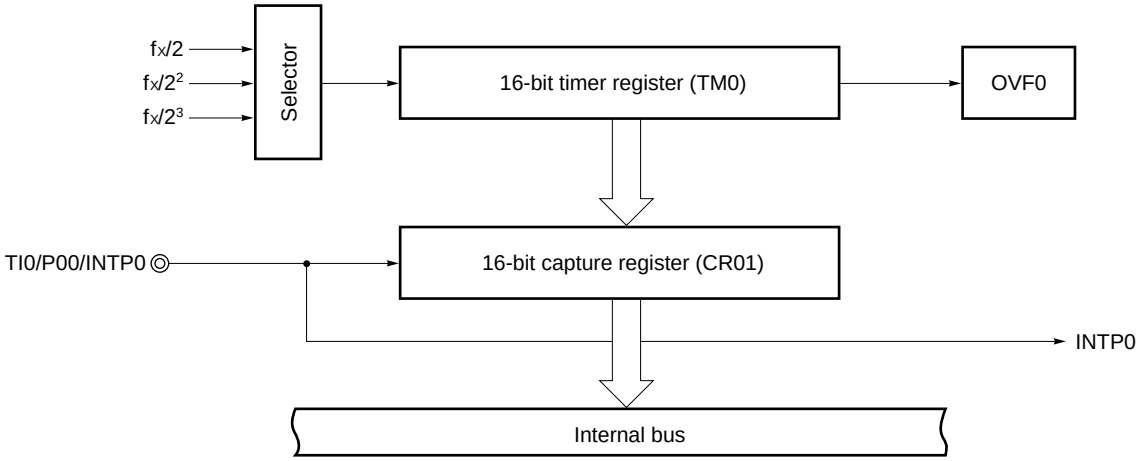
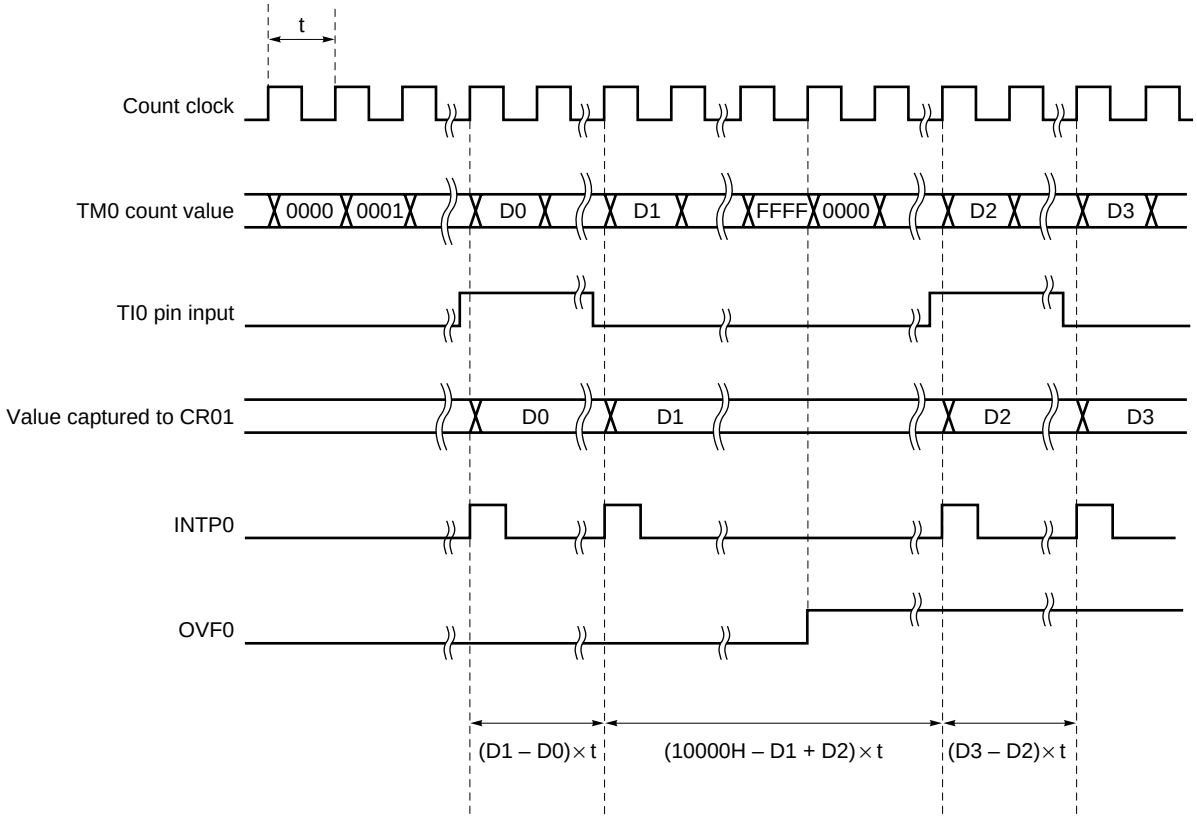


Figure 6-15. Pulse Width Measurement Timing by Free Running (with both rising and falling edges specified)



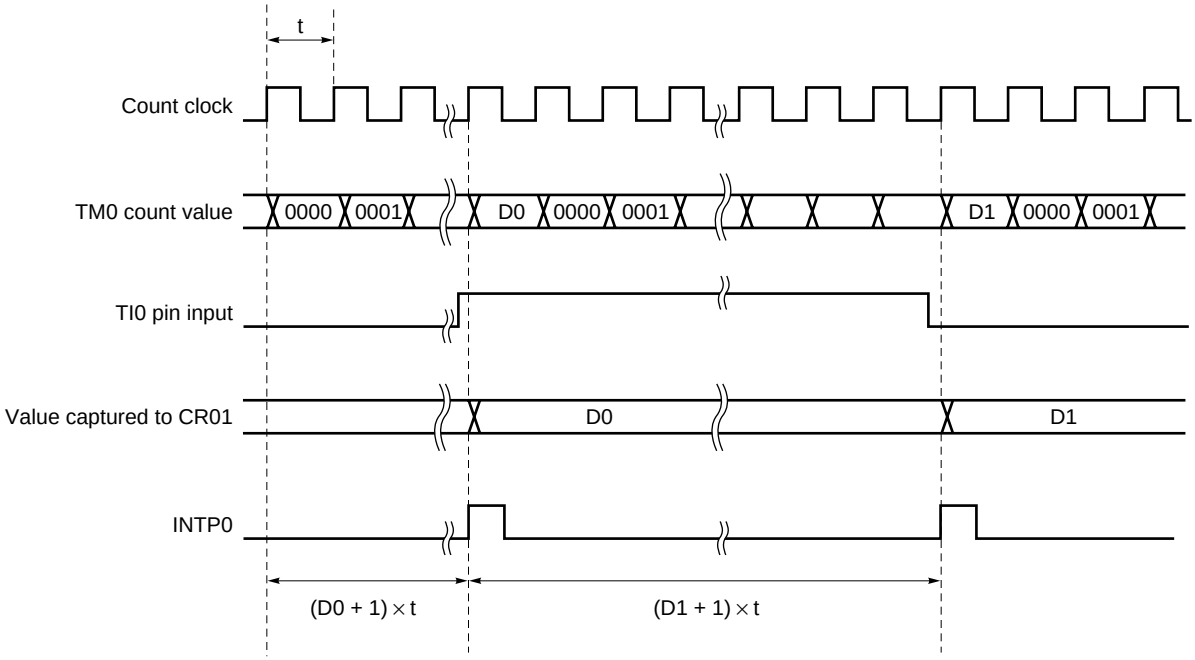
(2) Pulse width measurement by restarting timer

The pulse width of the signal input to the TI0/P00 pin is measured by clearing TM0 and restarting counting after the count value of the 16-bit timer register (TM0) has been captured to the 16-bit capture register (CR01), when the valid edge is detected on the TI0/P00 pin.

Three types of edges can be selected by bits 2 and 3 (ES10 and ES11) of external interrupt mode register (INTM0): rising, falling, and both rising and falling edges.

To detect the valid edge, sampling is performed at the cycle selected by the sampling clock select register (SCS). The value of TM0 is not captured until the valid edge is detected two times. Consequently, noise that may be superimposed on a pulse with a short pulse width can be eliminated.

Figure 6-16. Pulse Width Measurement Timing by Restarting Timer (with both rising and falling edges specified)



6.5.4 Operation as external event counter

The external event counter counts the number of clock pulses externally input to the TI0/P00 pin by using the 16-bit timer register (TM0).

Each time the valid edge specified by the external interrupt mode register (INTM0) is input, the value of TM0 is incremented.

When the measured value of TM0 coincides with the value of the 16-bit compare register (CR00), TM0 is cleared to 0, and an interrupt request signal (INTTM0) is generated.

★ Set CR00 to a value other than 0000H (one-pulse count cannot be performed).

Three types of edges can be selected by bits 2 and 3 (ES10 and ES11) of INTM0: rising, falling, and both rising and falling edges.

To detect the valid edge, sampling is performed at the cycle selected by the sampling clock select register (SCS). The value of TM0 is not captured until the valid edge is detected two times. Consequently, noise that may be superimposed on a pulse with a short pulse width can be eliminated.

Figure 6-17. Configuration of External Event Counter

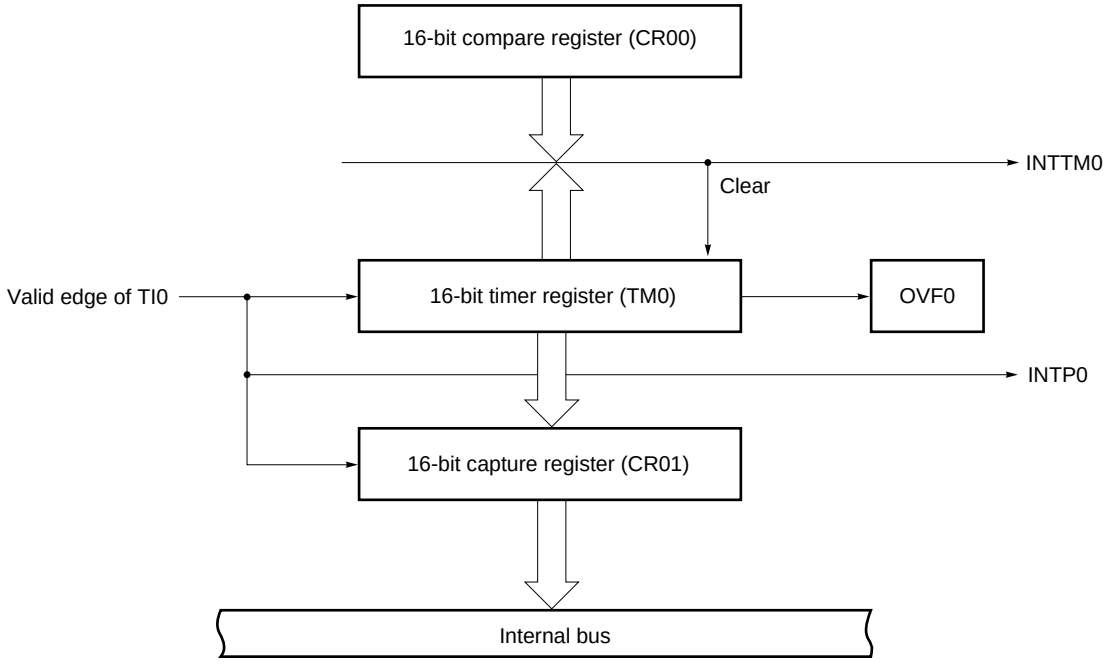
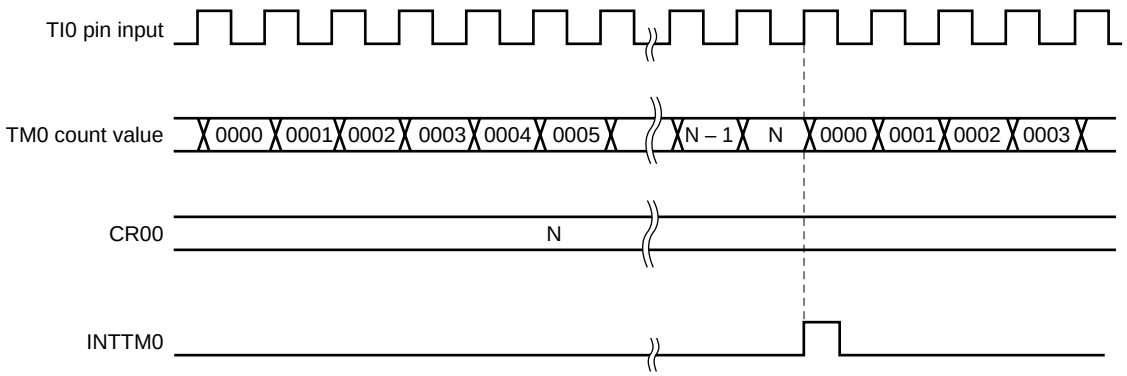


Figure 6-18. External Event Counter Operation Timing (with rising edge specified)



6.5.5 Operation as square wave output

The 16-bit timer/event counter can be used to output square waves of any frequency at time intervals specified by the count value set to the 16-bit compare register (CR00) in advance.

When the bits 0 and 1 (TOE0 and TOC01) of the 16-bit timer output control register (TOC0) are set to 1, the output status of the TO0/P30 pin is inverted at time intervals specified by the count value set to CR00 in advance. In this way, square waves of any frequency can be output.

Table 6-6. Square Wave Output Range of 16-Bit Timer/Event Counter

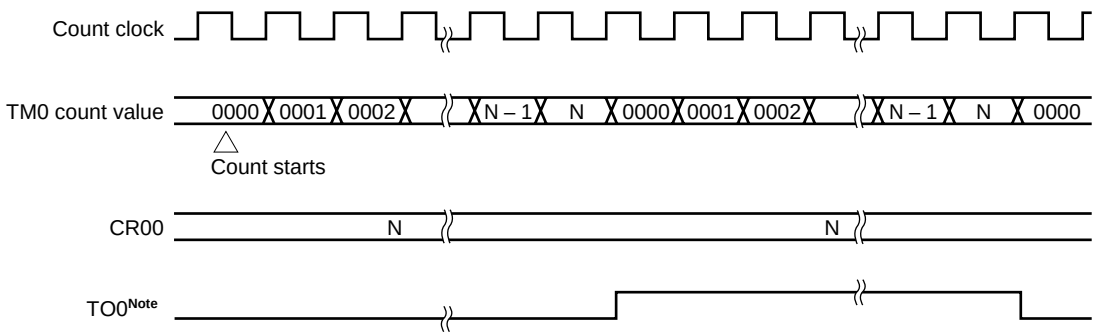
TCL06	TCL05	TCL04	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
0	1	0	2^2 × 1/fx (400 ns)	2^{17} × 1/fx (13.1 ms)	2 × 1/fx (200 ns)
0	1	1	2^3 × 1/fx (800 ns)	2^{18} × 1/fx (26.2 ms)	2^2 × 1/fx (400 ns)
1	0	0	2^4 × 1/fx (1.6 μs)	2^{19} × 1/fx (52.4 ms)	2^3 × 1/fx (800 ns)

Remarks 1. fx : main system clock oscillation frequency

2. TCL04 to TCL06 : bits 4 through 6 of timer clock select register 0 (TCL0)

3. ( ) : at fx = 10.0-MHz operation

Figure 6-19. Square Wave Output Timing



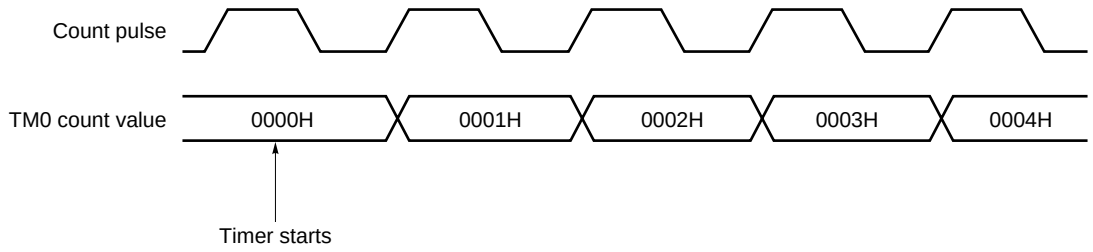
Note The initial value of TO0 output can be set by bits 2 and 3 (LVR0 and LVS0) of 16-bit timer output control register (TOC0).

6.6 Notes on Using 16-Bit Timer/Event Counter

(1) Error on starting timer

An error of up to 1 clock occurs after the timer has been started until a coincidence signal is generated. This is because the 16-bit timer register (TM0) is started in asynchronization with the count pulse.

Figure 6-20. Start Timing of 16-Bit Timer Register



(2) Setting of 16-bit compare register

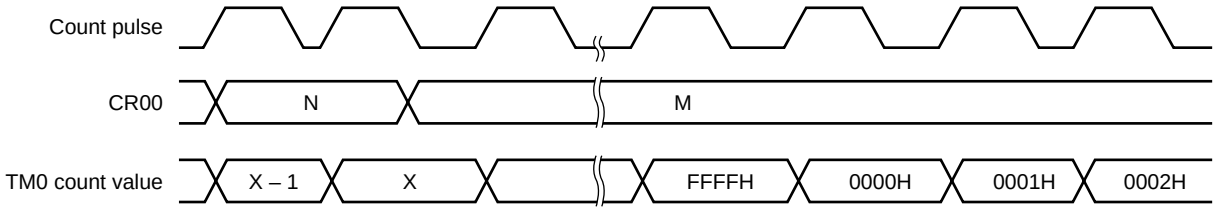
Set a value other than 0000H to the 16-bit compare register (CR00).

Therefore, one pulse cannot be counted when the 16-bit timer/event counter operates as an event counter.

(3) Operation after changing value of compare register during timer count operation

If a new value of the 16-bit compare register (CR00) is less than the value of the 16-bit timer register (TM0), TM0 continues counting, overflows, and restarts counting from 0. Therefore, if the new value of CR00 (M) is less than its old value (N), it is necessary to restart the timer after changing the value of CR00.

Figure 6-21. Timing after Changing Value of Compare Register during Timer Count Operation

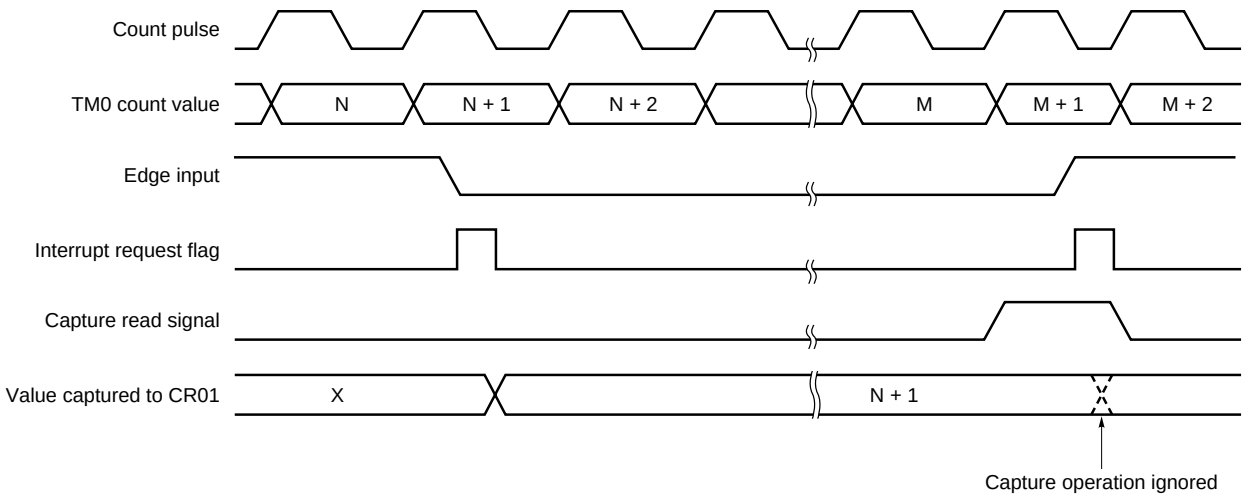


Remark N > X > M

(4) Data hold timing of capture register

When the valid edge is input to the TI0/P00 pin while data is read from the 16-bit capture register (CR01), CR01 does not perform the capture operation, but holds the data. However, the interrupt request flag (PIF0) is set when the valid edge is detected.

Figure 6-22. Data Hold Timing of Capture Register



(5) Setting valid edge

Set the valid edge of the TI0/P00/INTP0 pin after setting the bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0 and then stopping the timer operation. The valid edge is set by using bits 2 and 3 (ES10 and ES11) of the external interrupt mode register (INTM0).

(6) Operation of OVF0 flag

The OVF0 flag is set to 1 in the following case:

Mode in which the timer is cleared and started on coincidence between TM0 and CR00 is selected

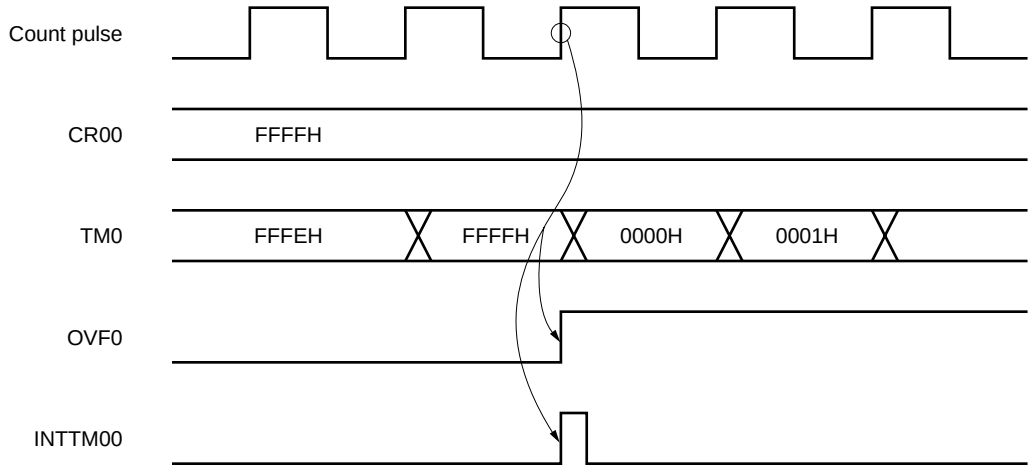
↓

CR00 is set to FFFFH

↓

TM0 counts up from FFFFH to 0000H

Figure 6-23. Operation Timing of OVF0 Flag



[MEMO]

CHAPTER 7 8-BIT TIMER/EVENT COUNTER

7.1 Function of 8-Bit Timer/Event Counter

The μPD78014H Subseries is provided with 8-bit timer/event counters which can be used in the following two modes:

- 8-bit timer/event counter mode : Two channels of 8-bit timer/event counters are individually used.
- 16-bit timer/event counter mode : Two channels of 8-bit timer/event counters are used in combination as a 16-bit timer/event counter.

7.1.1 8-bit timer/event counter mode

The 8-bit timer/event counters 1 and 2 (TM1 and TM2) have the following functions:

- Interval timer
- External event counter
- Square wave output

(1) 8-bit interval timer

When an 8-bit timer/event counter is used as an interval timer, it generates an interrupt request at any time intervals set in advance.

Table 7-1. Interval Time of 8-Bit Timer/Event Counter

Minimum Interval Time	Maximum Interval Time	Resolution
$2^2 \times 1/f_X$ (400 ns)	$2^{10} \times 1/f_X$ (102.4 μs)	$2^2 \times 1/f_X$ (400 ns)
$2^3 \times 1/f_X$ (800 ns)	$2^{11} \times 1/f_X$ (204.8 μs)	$2^3 \times 1/f_X$ (800 ns)
$2^4 \times 1/f_X$ (1.6 μs)	$2^{12} \times 1/f_X$ (409.6 μs)	$2^4 \times 1/f_X$ (1.6 μs)
$2^5 \times 1/f_X$ (3.2 μs)	$2^{13} \times 1/f_X$ (819.2 μs)	$2^5 \times 1/f_X$ (3.2 μs)
$2^6 \times 1/f_X$ (6.4 μs)	$2^{14} \times 1/f_X$ (1.64 ms)	$2^6 \times 1/f_X$ (6.4 μs)
$2^7 \times 1/f_X$ (12.8 μs)	$2^{15} \times 1/f_X$ (3.28 ms)	$2^7 \times 1/f_X$ (12.8 μs)
$2^8 \times 1/f_X$ (25.6 μs)	$2^{16} \times 1/f_X$ (6.55 ms)	$2^8 \times 1/f_X$ (25.6 μs)
$2^9 \times 1/f_X$ (51.2 μs)	$2^{17} \times 1/f_X$ (13.1 ms)	$2^9 \times 1/f_X$ (51.2 μs)
$2^{10} \times 1/f_X$ (102.4 μs)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^{10} \times 1/f_X$ (102.4 μs)
$2^{12} \times 1/f_X$ (409.6 μs)	$2^{20} \times 1/f_X$ (104.9 ms)	$2^{12} \times 1/f_X$ (409.6 μs)

Remarks 1. f_X : main system clock oscillation frequency

2. ( ) : at f_X = 10.0-MHz operation

(2) External event counter

The number of pulses of an externally input signal can be measured.

(3) Square wave output

A square wave of any frequency can be output.

Table 7-2. Square Wave Output Range of 8-Bit Timer/Event Counter

Minimum Pulse Width	Maximum Pulse Width	Resolution
$2^2 \times 1/f_X$ (400 ns)	$2^{10} \times 1/f_X$ (102.4 μs)	$2^2 \times 1/f_X$ (400 ns)
$2^3 \times 1/f_X$ (800 ns)	$2^{11} \times 1/f_X$ (204.8 μs)	$2^3 \times 1/f_X$ (800 ns)
$2^4 \times 1/f_X$ (1.6 μs)	$2^{12} \times 1/f_X$ (409.6 μs)	$2^4 \times 1/f_X$ (1.6 μs)
$2^5 \times 1/f_X$ (3.2 μs)	$2^{13} \times 1/f_X$ (819.2 μs)	$2^5 \times 1/f_X$ (3.2 μs)
$2^6 \times 1/f_X$ (6.4 μs)	$2^{14} \times 1/f_X$ (1.64 ms)	$2^6 \times 1/f_X$ (6.4 μs)
$2^7 \times 1/f_X$ (12.8 μs)	$2^{15} \times 1/f_X$ (3.28 ms)	$2^7 \times 1/f_X$ (12.8 μs)
$2^8 \times 1/f_X$ (25.6 μs)	$2^{16} \times 1/f_X$ (6.55 ms)	$2^8 \times 1/f_X$ (25.6 μs)
$2^9 \times 1/f_X$ (51.2 μs)	$2^{17} \times 1/f_X$ (13.1 ms)	$2^9 \times 1/f_X$ (51.2 μs)
$2^{10} \times 1/f_X$ (102.4 μs)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^{10} \times 1/f_X$ (102.4 μs)
$2^{12} \times 1/f_X$ (409.6 μs)	$2^{20} \times 1/f_X$ (104.9 ms)	$2^{12} \times 1/f_X$ (409.6 μs)

Remarks 1. f_X : main system clock oscillation frequency

2. ( ) : at f_X = 10.0-MHz operation

7.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

When two 8-bit timer/event counters are used in combination as a 16-bit interval timer, it generates an interrupt request at any time intervals set in advance.

Table 7-3. Interval Time of 8-Bit Timer/Event Counters Used as 16-Bit Timer/Event Counter

Minimum Interval Time	Maximum Interval Time	Resolution
$2^2 \times 1/f_X$ (400 ns)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^2 \times 1/f_X$ (400 ns)
$2^3 \times 1/f_X$ (800 ns)	$2^{19} \times 1/f_X$ (52.4 ms)	$2^3 \times 1/f_X$ (800 ns)
$2^4 \times 1/f_X$ (1.6 μs)	$2^{20} \times 1/f_X$ (104.9 ms)	$2^4 \times 1/f_X$ (1.6 μs)
$2^5 \times 1/f_X$ (3.2 μs)	$2^{21} \times 1/f_X$ (209.7 ms)	$2^5 \times 1/f_X$ (3.2 μs)
$2^6 \times 1/f_X$ (6.4 μs)	$2^{22} \times 1/f_X$ (419.4 ms)	$2^6 \times 1/f_X$ (6.4 μs)
$2^7 \times 1/f_X$ (12.8 μs)	$2^{23} \times 1/f_X$ (838.9 ms)	$2^7 \times 1/f_X$ (12.8 μs)
$2^8 \times 1/f_X$ (25.6 μs)	$2^{24} \times 1/f_X$ (1.7 s)	$2^8 \times 1/f_X$ (25.6 μs)
$2^9 \times 1/f_X$ (51.2 μs)	$2^{25} \times 1/f_X$ (3.4 s)	$2^9 \times 1/f_X$ (51.2 μs)
$2^{10} \times 1/f_X$ (102.4 μs)	$2^{26} \times 1/f_X$ (6.7 s)	$2^{10} \times 1/f_X$ (102.4 μs)
$2^{12} \times 1/f_X$ (409.6 μs)	$2^{28} \times 1/f_X$ (26.8 s)	$2^{12} \times 1/f_X$ (409.6 μs)

Remarks 1. f_X : main system clock oscillation frequency

2. ( ) : at f_X = 10.0-MHz operation

(2) External event counter

The number of pulses of an externally input signal can be measured.

(3) Square wave output

A square wave of any frequency can be output.

Table 7-4. Square Wave Output Range of 8-Bit Timer/Event Counters Used as 16-Bit Timer/Event Counter

Minimum Pulse Width	Maximum Pulse Width	Resolution
$2^2 \times 1/f_X$ (400 ns)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^2 \times 1/f_X$ (400 ns)
$2^3 \times 1/f_X$ (800 ns)	$2^{19} \times 1/f_X$ (52.4 ms)	$2^3 \times 1/f_X$ (800 ns)
$2^4 \times 1/f_X$ (1.6 μs)	$2^{20} \times 1/f_X$ (104.9 ms)	$2^4 \times 1/f_X$ (1.6 μs)
$2^5 \times 1/f_X$ (3.2 μs)	$2^{21} \times 1/f_X$ (209.7 ms)	$2^5 \times 1/f_X$ (3.2 μs)
$2^6 \times 1/f_X$ (6.4 μs)	$2^{22} \times 1/f_X$ (419.4 ms)	$2^6 \times 1/f_X$ (6.4 μs)
$2^7 \times 1/f_X$ (12.8 μs)	$2^{23} \times 1/f_X$ (838.9 ms)	$2^7 \times 1/f_X$ (12.8 μs)
$2^8 \times 1/f_X$ (25.6 μs)	$2^{24} \times 1/f_X$ (1.7 s)	$2^8 \times 1/f_X$ (25.6 μs)
$2^9 \times 1/f_X$ (51.2 μs)	$2^{25} \times 1/f_X$ (3.4 s)	$2^9 \times 1/f_X$ (51.2 μs)
$2^{10} \times 1/f_X$ (102.4 μs)	$2^{26} \times 1/f_X$ (6.7 s)	$2^{10} \times 1/f_X$ (102.4 μs)
$2^{12} \times 1/f_X$ (409.6 μs)	$2^{28} \times 1/f_X$ (26.8 s)	$2^{12} \times 1/f_X$ (409.6 μs)

Remarks 1. f_X : main system clock oscillation frequency

2. ( ): at f_X = 10.0-MHz operation

7.2 Configuration of 8-Bit Timer/Event Counter

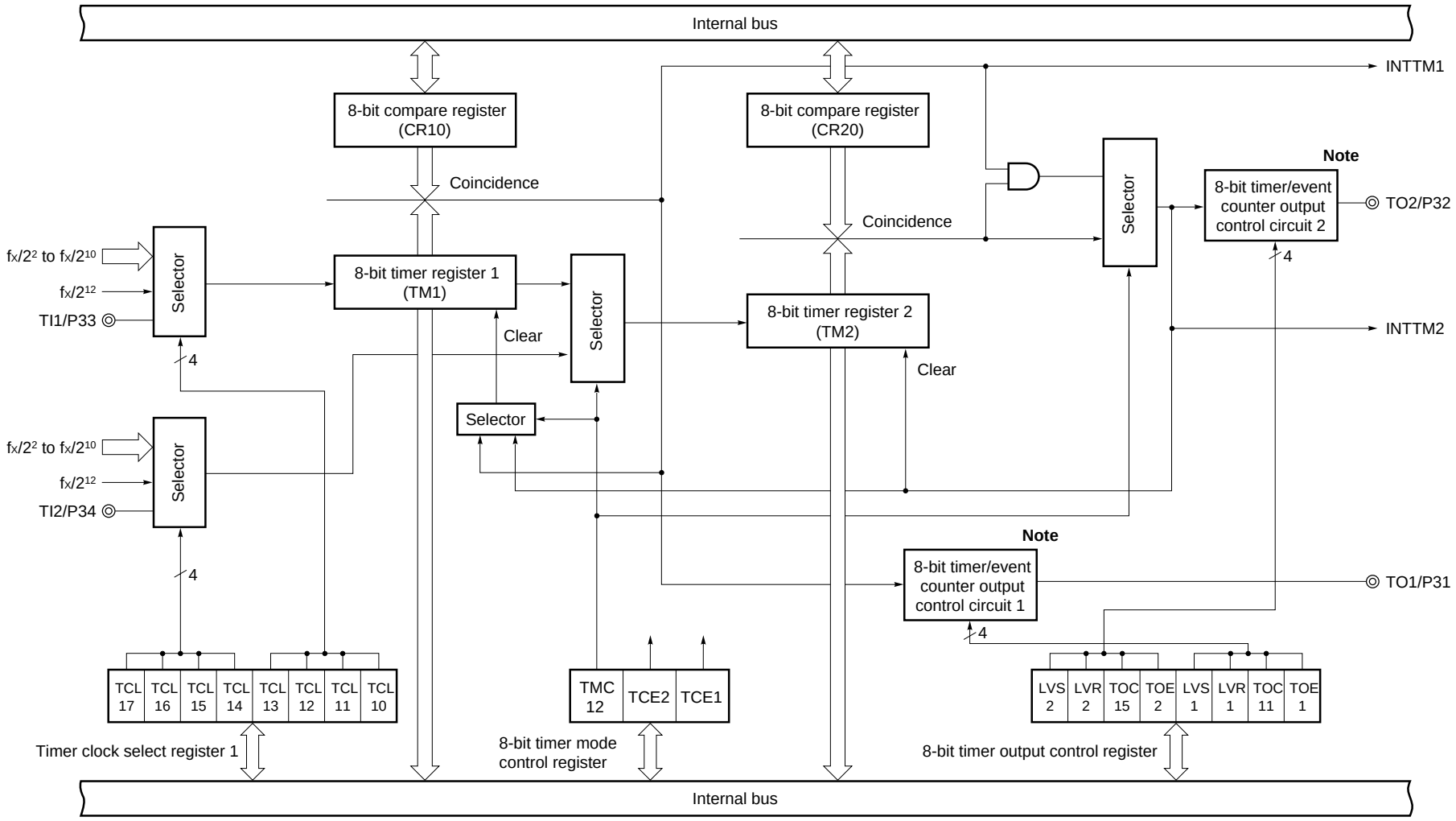
An 8-bit timer/event counter consists of the following hardware:

Table 7-5. Configuration of 8-Bit Timer/Event Counter

Item	Configuration
Timer register	8 bits × 2 (TM1, TM2)
Register	Compare register: 8 bits × 2 (CR10, CR20)
Timer output	2 (TO1, TO2)
Control register	Timer clock select register 1 (TCL1) 8-bit timer mode control register (TMC1) 8-bit timer output control register (TOC1) Port mode register 3 (PM3)[Note] Port 3 (P3)

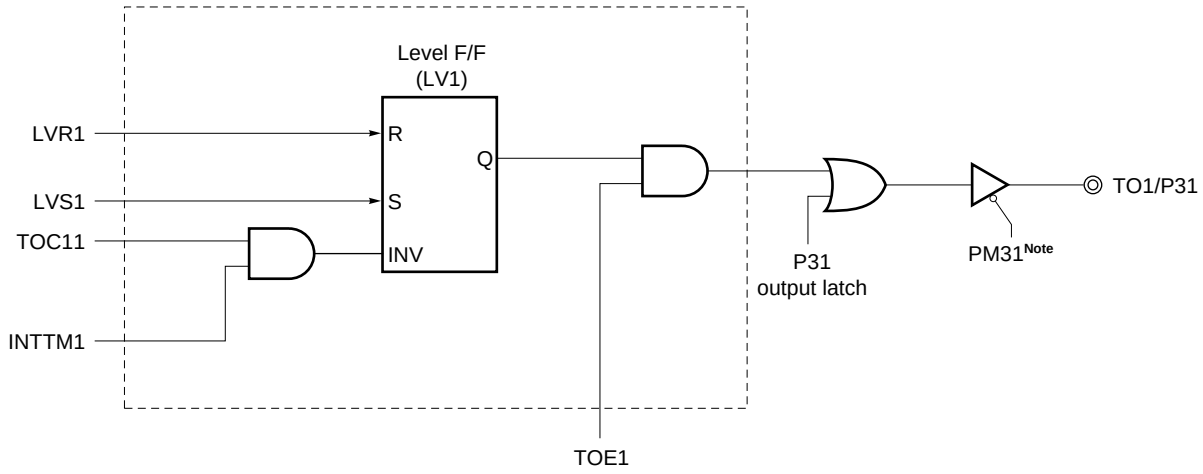
Note Refer to **Figure 4-8. Block Diagram of P30 to P37**.

Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter



Note For the configuration of the 8-bit timer/event counter output control circuits 1 and 2, refer to **Figures 7-2** and **7-3**.

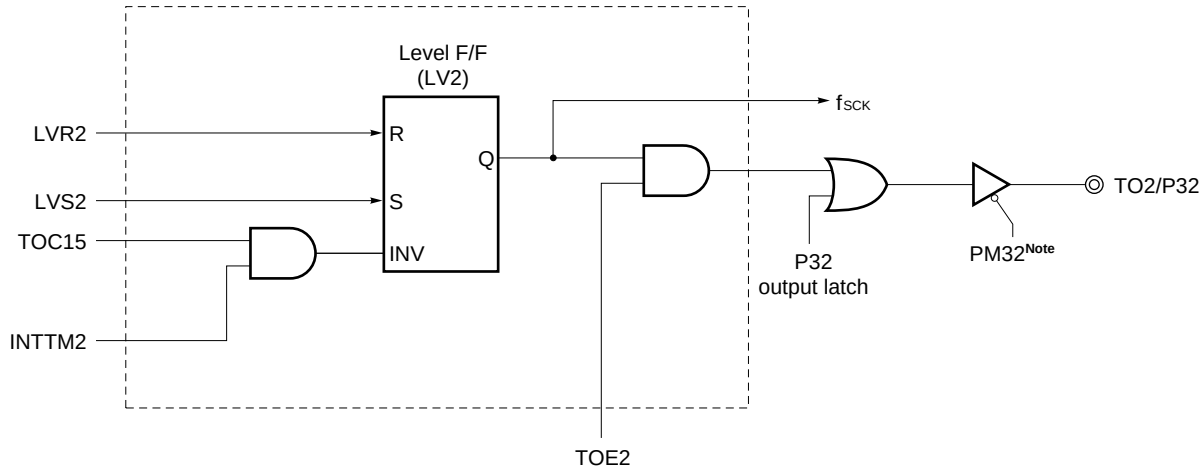
Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 1



Caution Bit 1 of port mode register 3 (PM3)

Remark The output control circuit is shown enclosed by broken line.

Figure 7-3. Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 2



Caution Bit 2 of port mode register 3 (PM3)

Remarks 1. The output control circuit is shown enclosed by broken line.

2. f_{SCK}: serial clock frequency

(1) 8-bit compare registers (CR10 and CR20)

These 8-bit registers always compare their set values with the count values of the 8-bit timer registers 1 and 2 (TM1 and TM2). CR10 compares its set value with TM1, while CR20 compares its set value with TM2. When the value of a compare register coincides with the count value of the corresponding timer register, the compare register generates an interrupt request (INTTM1 or INTTM2).

When TM1 or TM2 is used as an interval timer, the corresponding compare register can also be used to hold interval time.

CR10 and CR20 are set by an 8-bit memory manipulation instruction, and cannot be set by a 16-bit memory manipulation instruction. When TM1 and TM2 are used as 8-bit timer/event counters, a value in a range of 00H to FFH can be set to the corresponding compare registers. When the two 8-bit timer/event counters are used in combination as a 16-bit timer/event counter, a value in a range of 0000H to FFFFH can be set to the two compare registers.

The contents of these registers become undefined when the $\overline{\text{RESET}}$ signal is input.

Cautions 1. Set data to these registers when the two 8-bit timer/event counters are used as a 16-bit timer/event counter after stopping the operation of the timer.

2. When the values of CR10 and CR20 after changing are less than the values of 8-bit timer register (TM1 and TM2), TM1 and TM2 continue counting and start counting from 0 after overflow. For this reason, when the values of CR10 and CR20 after changing are less than the values before changing, restart the timer after changing CR10 and CR20.

(2) 8-bit timer registers 1 and 2 (TM1 and TM2)

These 8-bit registers count the number of count pulses.

When TM1 and TM2 are used individually, the value of each timer register can be read by an 8-bit memory manipulation instruction. When the two timer registers are used in combination as a 16-bit timer, the value of the 16-bit timer register (TMS) can be read by a 16-bit memory manipulation instruction.

These registers are initialized to 00H when the $\overline{\text{RESET}}$ signal is input.

7.3 Registers Controlling 8-Bit Timer/Event Counter

The following four types of registers control the 8-bit timer/event counters:

- Timer clock select register 1 (TCL1)
- 8-bit timer mode control register (TMC1)
- 8-bit timer output control register (TOC1)
- Port mode register 3 (PM3)

(1) Timer clock select register 1 (TCL1)

This register sets the count clocks of the 8-bit timer registers 1 and 2.

TCL1 is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 7-4. Format of Timer Clock Select Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL1	TCL17	TCL16	TCL15	TCL14	TCL13	TCL12	TCL11	TCL10	FF41H	00H	R/W

TCL13	TCL12	TCL11	TCL10	Selects count clock of 8-bit timer register 1
0	0	0	0	Falling edge of TI1
0	0	0	1	Rising edge of TI1
0	1	1	0	$f_X/2^2$ (2.5 MHz)
0	1	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	0	$f_X/2^4$ (625 kHz)
1	0	0	1	$f_X/2^5$ (313 kHz)
1	0	1	0	$f_X/2^6$ (156 kHz)
1	0	1	1	$f_X/2^7$ (78.1 kHz)
1	1	0	0	$f_X/2^8$ (39.1 kHz)
1	1	0	1	$f_X/2^9$ (19.5 kHz)
1	1	1	0	$f_X/2^{10}$ (9.8 kHz)
1	1	1	1	$f_X/2^{12}$ (2.4 kHz)
Others				Setting prohibited

TCL17	TCL16	TCL15	TCL14	Selects count clock of 8-bit timer register 2
0	0	0	0	Falling edge of TI2
0	0	0	1	Rising edge of TI2
0	1	1	0	$f_X/2^2$ (2.5 MHz)
0	1	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	0	$f_X/2^4$ (625 kHz)
1	0	0	1	$f_X/2^5$ (313 kHz)
1	0	1	0	$f_X/2^6$ (156 kHz)
1	0	1	1	$f_X/2^7$ (78.1 kHz)
1	1	0	0	$f_X/2^8$ (39.1 kHz)
1	1	0	1	$f_X/2^9$ (19.5 kHz)
1	1	1	0	$f_X/2^{10}$ (9.8 kHz)
1	1	1	1	$f_X/2^{12}$ (2.4 kHz)
Others				Setting prohibited

Caution To write data other than that already written to TCL1, stop the timer operation once.

Remarks 1. f_X : main system clock oscillation frequency

2. TI1 : input pin of 8-bit timer register 1

3. TI2 : input pin of 8-bit timer register 2

4. ( ) : at f_X = 10.0-MHz operation

(2) 8-bit timer mode control register (TMC1)

This register enables or disables the operations of 8-bit timer registers 1 and 2, and sets an operation mode of 8-bit timer register 1, 2.

TMC1 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 7-5. Format of 8-Bit Timer Mode Control Register

Symbol	7	6	5	4	3	2	<1>	<0>	Address	After reset	R/W
TMC1	0	0	0	0	0	TMC12	TCE2	TCE1	FF49H	00H	R/W

TCE1	Controls operation of 8-bit timer register 1
0	Stops operation (TM1 is cleared to 0)
1	Enables operation

TCE2	Controls operation of 8-bit timer register 2
0	Stops operation (TM2 is cleared to 0)
1	Enables operation

TMC12	Selects operation mode
0	8-bit timer register × 2 channel mode (TM1 , TM2)
1	16-bit timer register × 1 channel mode (TMS)

Cautions 1. When changing the operation mode, stop the timer operations beforehand.

2. When TM1 is used as a 16-bit timer register (TMS), enable or disable the operation of TMS by using the TCE1 bit of the 8-bit timer mode control register (TMC1).

(3) 8-bit timer output control register (TOC1)

This register controls the operations of the 8-bit timer/event counter output control circuits 1 and 2.

It sets/resets an R-S flip-flops (LV1, LV2), enables/disables inversion of the timer output F/F, and enables/disables the outputs of 8-bit timer registers 1 and 2.

TOC1 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 7-6. Format of 8-Bit Timer Output Control Register

Symbol	<7>	<6>	5	<4>	<3>	<2>	1	<0>	Address	After reset	R/W
TOC1	LVS2	LVR2	TOC15	TOE2	LVS1	LVR1	TOC11	TOE1	FF4FH	00H	R/W

TOE1	Controls output of 8-bit timer/event counter 1
0	Disables output (port mode)
1	Enables output

TOC11	Controls timer output F/F of 8-bit timer/event counter 1
0	Disables inversion
1	Enables inversion

LVS1	LVR1	Sets status of timer output F/F of 8-bit timer/event counter 1
0	0	Not affected
0	1	Resets timer output F/F to 0
1	0	Sets timer output F/F to 1
1	1	Setting prohibited

TOE2	Controls output of 8-bit timer/event counter 2
0	Disables output (port mode)
1	Enables output

TOC15	Controls timer output F/F of 8-bit timer/event counter 2
0	Disables inversion
1	Enables inversion

LVS2	LVR2	Sets status of timer output F/F of 8-bit timer/event counter 2
0	0	Not affected
0	1	Resets timer output F/F to 0
1	0	Sets timer output F/F to 1
1	1	Setting prohibited

Cautions 1. Be sure to stop the timer operation before setting TOC1.

2. 0 is read from LVS1, LVS2, LVR1, and LVR2 after data has been set to these bits.

(4) Port mode register 3 (PM3)

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

When the P31/TO1 and P32/TO2 pins are used as timer output pins, set 0 to the PM31 and PM32 bits of this register and the output latch of the P31 and P32 pins.

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

This register is set to FFH when the $\overline{\text{RESET}}$ signal is input.

Figure 7-7. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	Selects I/O mode of P3n pin (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

7.4 Operation of 8-Bit Timer/Event Counter

7.4.1 8-bit timer/event counter mode

(1) Operation as interval timer

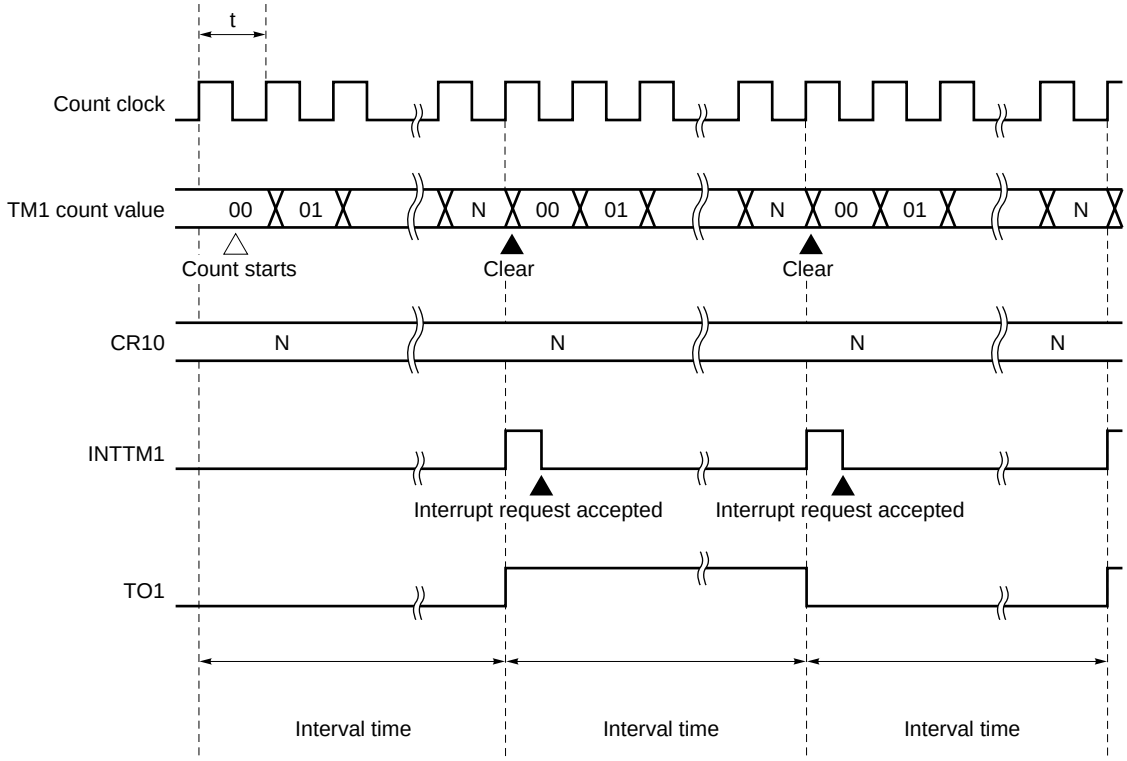
The 8-bit timer/event counters operate as interval timers and repeatedly generate an interrupt request at time intervals specified by the count values set to the corresponding 8-bit compare registers (CR10 and CR20) in advance.

When the count values of the 8-bit timer registers 1 and 2 (TM1 and TM2) coincide with the values set to the corresponding compare registers CR10 and CR20, the values of TM1 and TM2 are cleared to 0, TM1 and TM2 continue counting, and at the same time, interrupt request signals (INTTM1 and INTTM2) are generated.

The count clock of the TM1 can be selected by bits 0 to 3 (TCL10 to TCL13) of the timer clock select register 1 (TCL1), and the count clock of the TM2 can be selected by the bits 4 to 7 (TCL14 to TCL17) of TCL1.

For the operation when the value of compare register is changed during timer count operation, refer to **7.5 (3) Operation after changing value of compare register during timer count operation**.

Figure 7-8. Interval Timer Operation Timing



Remark Interval time = (N + 1) × t: N = 00H to FFH

Table 7-6. Interval Time of 8-Bit Timer/Event Counter 1

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI1 input cycle	$2^8 \times$ TI1 input cycle	TI1 input edge cycle
0	0	0	1	TI1 input cycle	$2^8 \times$ TI1 input cycle	TI1 input edge cycle
0	1	1	0	$2^2 \times 1/fx$ (400 ns)	$2^{10} \times 1/fx$ (102.4 μs)	$2^2 \times 1/fx$ (400 ns)
0	1	1	1	$2^3 \times 1/fx$ (800 ns)	$2^{11} \times 1/fx$ (204.8 μs)	$2^3 \times 1/fx$ (800 ns)
1	0	0	0	$2^4 \times 1/fx$ (1.6 μs)	$2^{12} \times 1/fx$ (409.6 μs)	$2^4 \times 1/fx$ (1.6 μs)
1	0	0	1	$2^5 \times 1/fx$ (3.2 μs)	$2^{13} \times 1/fx$ (819.2 μs)	$2^5 \times 1/fx$ (3.2 μs)
1	0	1	0	$2^6 \times 1/fx$ (6.4 μs)	$2^{14} \times 1/fx$ (1.64 ms)	$2^6 \times 1/fx$ (6.4 μs)
1	0	1	1	$2^7 \times 1/fx$ (12.8 μs)	$2^{15} \times 1/fx$ (3.28 ms)	$2^7 \times 1/fx$ (12.8 μs)
1	1	0	0	$2^8 \times 1/fx$ (25.6 μs)	$2^{16} \times 1/fx$ (6.55 ms)	$2^8 \times 1/fx$ (25.6 μs)
1	1	0	1	$2^9 \times 1/fx$ (51.2 μs)	$2^{17} \times 1/fx$ (13.1 ms)	$2^9 \times 1/fx$ (51.2 μs)
1	1	1	0	$2^{10} \times 1/fx$ (102.4 μs)	$2^{18} \times 1/fx$ (26.2 ms)	$2^{10} \times 1/fx$ (102.4 μs)
1	1	1	1	$2^{12} \times 1/fx$ (409.6 μs)	$2^{20} \times 1/fx$ (104.9 ms)	$2^{12} \times 1/fx$ (409.6 μs)
Others				Setting prohibited		

Remarks **1.** f_X : main system clock oscillation frequency

2. TCL10 to TCL13 : bits 0 to 3 of timer clock selection register 1 (TCL1)

3. ( ) : at f_X = 10.0-MHz operation

Table 7-7. Interval Time of 8-Bit Timer/Event Counter 2

TCL17	TCL16	TCL15	TCL14	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI2 input cycle	$2^8 \times$ TI2 input cycle	TI2 input edge cycle
0	0	0	1	TI2 input cycle	$2^8 \times$ TI2 input cycle	TI2 input edge cycle
0	1	1	0	$2^2 \times 1/f_X$ (400 ns)	$2^{10} \times 1/f_X$ (102.4 μs)	$2^2 \times 1/f_X$ (400 ns)
0	1	1	1	$2^3 \times 1/f_X$ (800 ns)	$2^{11} \times 1/f_X$ (204.8 μs)	$2^3 \times 1/f_X$ (800 ns)
1	0	0	0	$2^4 \times 1/f_X$ (1.6 μs)	$2^{12} \times 1/f_X$ (409.6 μs)	$2^4 \times 1/f_X$ (1.6 μs)
1	0	0	1	$2^5 \times 1/f_X$ (3.2 μs)	$2^{13} \times 1/f_X$ (819.2 μs)	$2^5 \times 1/f_X$ (3.2 μs)
1	0	1	0	$2^6 \times 1/f_X$ (6.4 μs)	$2^{14} \times 1/f_X$ (1.64 ms)	$2^6 \times 1/f_X$ (6.4 μs)
1	0	1	1	$2^7 \times 1/f_X$ (12.8 μs)	$2^{15} \times 1/f_X$ (3.28 ms)	$2^7 \times 1/f_X$ (12.8 μs)
1	1	0	0	$2^8 \times 1/f_X$ (25.6 μs)	$2^{16} \times 1/f_X$ (6.55 ms)	$2^8 \times 1/f_X$ (25.6 μs)
1	1	0	1	$2^9 \times 1/f_X$ (51.2 μs)	$2^{17} \times 1/f_X$ (13.1 ms)	$2^9 \times 1/f_X$ (51.2 μs)
1	1	1	0	$2^{10} \times 1/f_X$ (102.4 μs)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^{10} \times 1/f_X$ (102.4 μs)
1	1	1	1	$2^{12} \times 1/f_X$ (409.6 μs)	$2^{20} \times 1/f_X$ (104.9 ms)	$2^{12} \times 1/f_X$ (409.6 μs)
Others				Setting prohibited		

Remarks 1. f_X : main system clock oscillation frequency

2. TCL14 to TCL17 : bits 4 to 7 of timer clock selection register 1 (TCL1)

3. ( ) : at f_X = 10.0-MHz operation

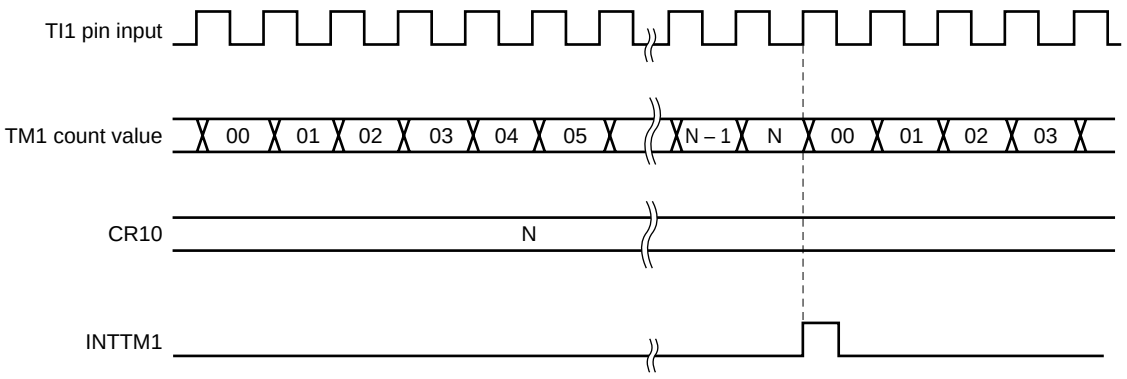
(2) Operation as external event counter

The external event counter counts the number of clock pulses externally input to the TI1/P33 and TI2/P34 pins by using the 8-bit timer registers 1 and 2 (TM1 and TM2).

Each time the valid edge specified by the timer clock select register 1 (TCL1) is input, the values of TM1 and TM2 are incremented. Either the rising edge or falling edge can be specified as the valid edge.

When the count values of TM1 and TM2 coincide with the values of the corresponding 8-bit compare registers (CR10 and CR20), TM1 and TM2 are cleared to 0, and interrupt request signals (INTTM1 and INTTM2) are generated.

Figure 7-9. External Event Counter Operation Timing (with rising edge specified)



Remark N = 00H to FFH

(3) Operation as square wave output

The 8-bit timer/event counters can be used to output square waves of any frequency at time intervals specified by the values set to the corresponding 8-bit compare registers (CR10 and CR20) in advance.

When bit 0 or 4 (TOE1 or TOE2) of the 8-bit timer output control register (TOC1) is set to 1, the output status of the TO1/P31 or TO2/P32 pin is inverted at time intervals specified by the values set to CR10 or CR20 in advance. In this way, square waves of any frequency can be output.

Table 7-8. Square Wave Output Range of 8-Bit Timer/Event Counters

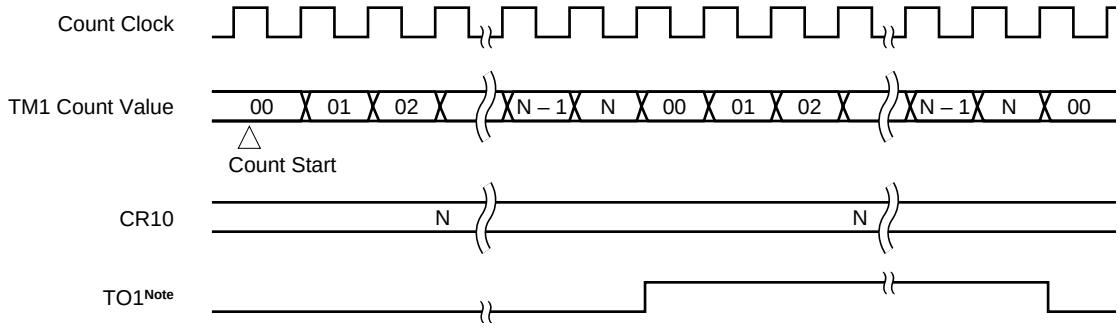
TCL13	TCL12	TCL11	TCL10	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	1	1	0	$2^2 \times 1/fx$ (400 ns)	$2^{10} \times 1/fx$ (102.4 μs)	$2^2 \times 1/fx$ (400 ns)
0	1	1	1	$2^3 \times 1/fx$ (800 ns)	$2^{11} \times 1/fx$ (204.8 μs)	$2^3 \times 1/fx$ (800 ns)
1	0	0	0	$2^4 \times 1/fx$ (1.6 μs)	$2^{12} \times 1/fx$ (409.6 μs)	$2^4 \times 1/fx$ (1.6 μs)
1	0	0	1	$2^5 \times 1/fx$ (3.2 μs)	$2^{13} \times 1/fx$ (819.2 μs)	$2^5 \times 1/fx$ (3.2 μs)
1	0	1	0	$2^6 \times 1/fx$ (6.4 μs)	$2^{14} \times 1/fx$ (1.64 ms)	$2^6 \times 1/fx$ (6.4 μs)
1	0	1	1	$2^7 \times 1/fx$ (12.8 μs)	$2^{15} \times 1/fx$ (3.28 ms)	$2^7 \times 1/fx$ (12.8 μs)
1	1	0	0	$2^8 \times 1/fx$ (25.6 μs)	$2^{16} \times 1/fx$ (6.55 ms)	$2^8 \times 1/fx$ (25.6 μs)
1	1	0	1	$2^9 \times 1/fx$ (51.2 μs)	$2^{17} \times 1/fx$ (13.1 ms)	$2^9 \times 1/fx$ (51.2 μs)
1	1	1	0	$2^{10} \times 1/fx$ (102.4 μs)	$2^{18} \times 1/fx$ (26.2 ms)	$2^{10} \times 1/fx$ (102.4 μs)
1	1	1	1	$2^{12} \times 1/fx$ (409.6 μs)	$2^{20} \times 1/fx$ (104.9 ms)	$2^{12} \times 1/fx$ (409.6 μs)

Remarks **1.** f_X : main system clock oscillation frequency

2. TCL10 to TCL13 : bits 0 to 3 of timer clock selection register 1 (TCL1)

3. ( ) : at f_X = 10.0-MHz operation

★ **Figure 7-10. Square Wave Output Operation Timings**



Note Initial value of TO1 output can be set at bits 2 and 3 (LVS1 and LVR1) of the 8-bit timer output control register (TOC1).

7.4.2 16-bit timer/event counter mode

When bit 2 (TMC12) of the 8-bit timer mode control register (TMC1) is set to 1, the 16-bit timer/event counter mode is selected.

The count clocks are selected with bits 0 to 3 (TCL10 to TCL13) of timer clock select register (TCL1). The overflow signal of 8-bit timer/event counter 1 (TM1) becomes a count clock of 8-bit timer/event counter 2 (TM2). Count operation enable/disable is selected with bit 0 (TCE1) of TMC1.

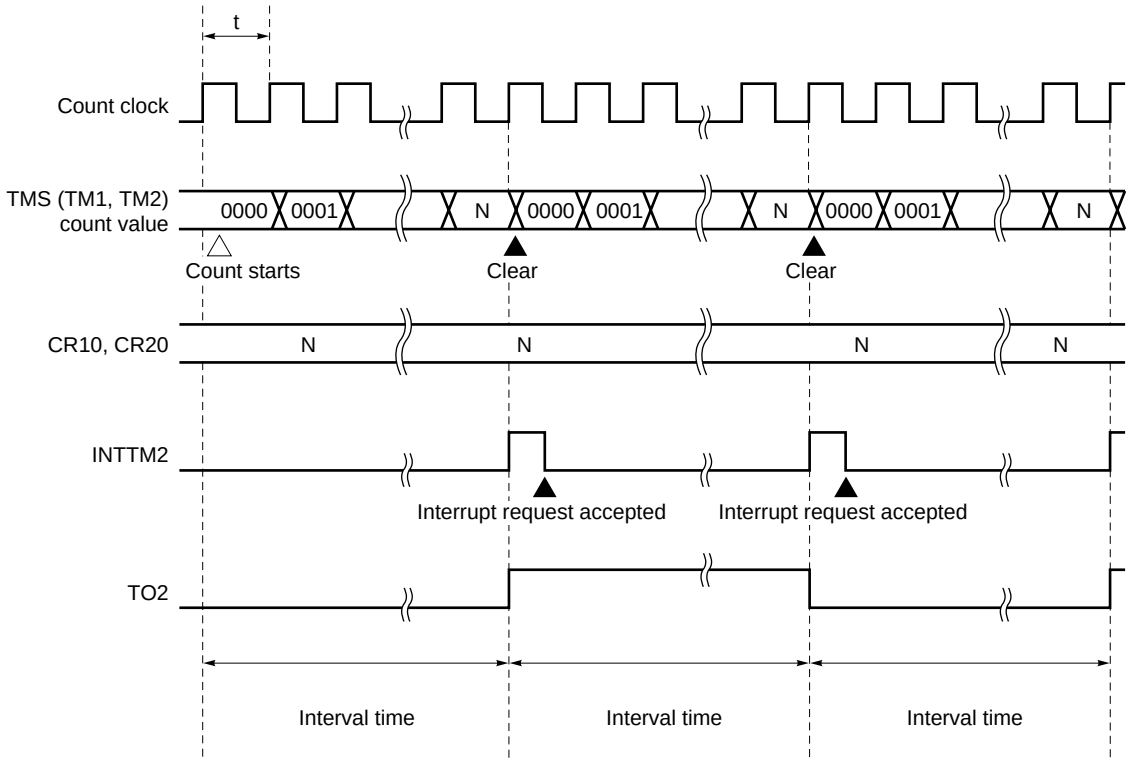
★ **(1) Interval timer operation**

The 8-bit timer/event counter operates as interval timer which generates interrupt requests repeatedly at intervals of the count value preset to 2-channel 8-bit compare registers (CR10 and CR20). When setting the count value, the upper 8-bit value is set as CR20 and the lower 8-bit value as CR10. For the count value (interval time) which can be set refer to **Table 7-9**.

When the 8-bit timer register 1 (TM1) and CR10 values match and the 8-bit timer register 2 (TM2) and CR20 values match, counting continues with the TM1 and TM2 values cleared to 0 and the interrupt request signal (INTTM2) is generated. For the operation timings of the interval timer, refer to **Figure 7-11**.

Count clock can be selected with bits 0 to 3 (TCL10 to TCL13) of the timer clock select register 1 (TCL1). The overflow signal of the TM1 becomes a count clock of the TM2.

Figure 7-11. Interval Timer Operation Timing



Remark Interval time = (N + 1) x t: N = 0000H to FFFFH

Caution Even when the two 8-bit timers are used in combination in a 16-bit timer/event counter mode, when the count value of TM1 coincides with the value of CR10, an interrupt request (INTTM1) is generated, and the F/F of the 8-bit timer/event counter output control circuit 1 is inverted. When using the 8-bit timers as a 16-bit interval timer, set mask flag TMMK1, which disables accepting INTTM1, to 1.

To read the count value of the 16-bit timer register (TMS), use a 16-bit memory manipulation instruction.

Table 7-9. Interval Time when Two 8-Bit Timer/Event Counters (TM1 and TM2) Are Used as One 16-Bit Timer/Event Counter

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	0	TI1 input cycle	$2^8 \times$ TI1 input cycle	TI1 input edge cycle
0	0	0	1	TI1 input cycle	$2^8 \times$ TI1 input cycle	TI1 input edge cycle
0	1	1	0	$2^2 \times 1/fx$ (400 ns)	$2^{18} \times 1/fx$ (26.2 ms)	$2^2 \times 1/fx$ (400 ns)
0	1	1	1	$2^3 \times 1/fx$ (800 ns)	$2^{19} \times 1/fx$ (52.4 ms)	$2^3 \times 1/fx$ (800 ns)
1	0	0	0	$2^4 \times 1/fx$ (1.6 μs)	$2^{20} \times 1/fx$ (104.9 ms)	$2^4 \times 1/fx$ (1.6 μs)
1	0	0	1	$2^5 \times 1/fx$ (3.2 μs)	$2^{21} \times 1/fx$ (209.7 ms)	$2^5 \times 1/fx$ (3.2 μs)
1	0	1	0	$2^6 \times 1/fx$ (6.4 μs)	$2^{22} \times 1/fx$ (419.4 ms)	$2^6 \times 1/fx$ (6.4 μs)
1	0	1	1	$2^7 \times 1/fx$ (12.8 μs)	$2^{23} \times 1/fx$ (838.9 ms)	$2^7 \times 1/fx$ (12.8 μs)
1	1	0	0	$2^8 \times 1/fx$ (25.6 μs)	$2^{24} \times 1/fx$ (1.7 s)	$2^8 \times 1/fx$ (25.6 μs)
1	1	0	1	$2^9 \times 1/fx$ (51.2 μs)	$2^{25} \times 1/fx$ (3.4 s)	$2^9 \times 1/fx$ (51.2 μs)
1	1	1	0	$2^{10} \times 1/fx$ (102.4 μs)	$2^{26} \times 1/fx$ (6.7 s)	$2^{10} \times 1/fx$ (102.4 μs)
1	1	1	1	$2^{12} \times 1/fx$ (409.6 μs)	$2^{28} \times 1/fx$ (26.8 s)	$2^{12} \times 1/fx$ (409.6 μs)
Others				Setting prohibited		

Remarks 1. fx : main system clock oscillation frequency

2. TCL10 to TCL13 : bits 0 to 3 of timer clock selection register 1 (TCL1)

3. ( ) : at fx = 10.0-MHz operation

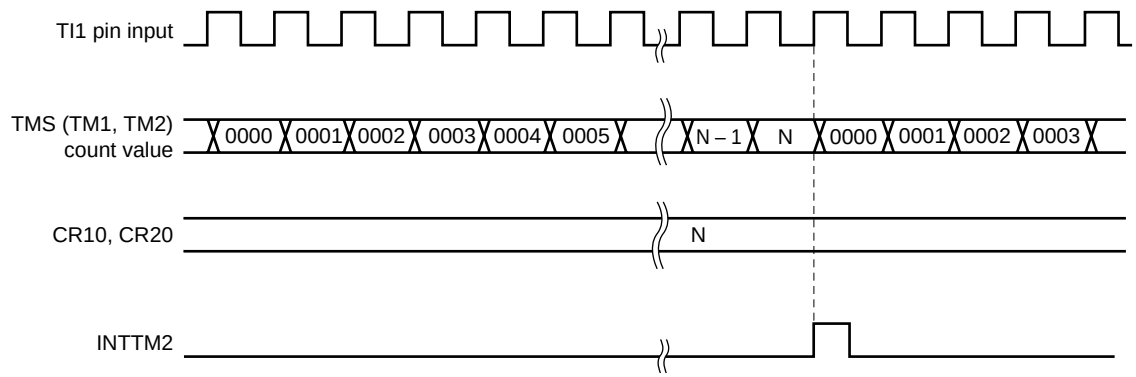
(2) Operation as external event counter

The external event counter counts the number of clock pulses externally input to the TI1/P33 pin by using the two channels of 8-bit timer registers 1 and 2 (TM1 and TM2).

★ Each time the valid edge specified by the timer clock select register 1 (TCL1) is input, TM1 is incremented. When TM1 overflows, TM2 is incremented based on the overflow signal. Either the rising edge or falling edge can be specified as the edge.

When the count values of TM1 and TM2 coincide with the values of the corresponding 8-bit compare registers (CR10 and CR20), TM1 and TM2 are cleared to 0, and an interrupt request signal (INTTM2) is generated.

Figure 7-12. External Event Counter Operation Timing (with rising edge specified)



Caution Even when the two 8-bit timers are used in combination in a 16-bit timer/event counter mode, when the count value of TM1 coincides with the value of CR10, an interrupt request (INTTM1) is generated, and the F/F of the 8-bit timer/event counter output control circuit 1 is inverted. When using the 8-bit timers as a 16-bit interval timer, set mask flag TMMK1, which disables accepting INTTM1, to 1.

To read the count value of the 16-bit timer register (TMS), use a 16-bit memory manipulation instruction.

(3) Operation as square wave output

The 8-bit timers/event counters can be used to output square waves of any frequency at time intervals specified by the values set to the corresponding 8-bit compare registers (CR10 and CR20) in advance. When setting count values, set the value of high order 8 bit to CR20 and low order 8 bit to CR10.

When the bit 4 (TOE2) of the 8-bit timer output control register (TOC1) is set to 1, the output status of the TO2/P32 pin is inverted at time intervals specified by the count values set to CR10 or CR20 in advance. In this way, square waves of any frequency can be output.

Table 7-10. Square Wave Output Range when Two 8-Bit Timer/Event Counters (TM1 and TM2) Are Used as One 16-Bit Timer/Event Counter

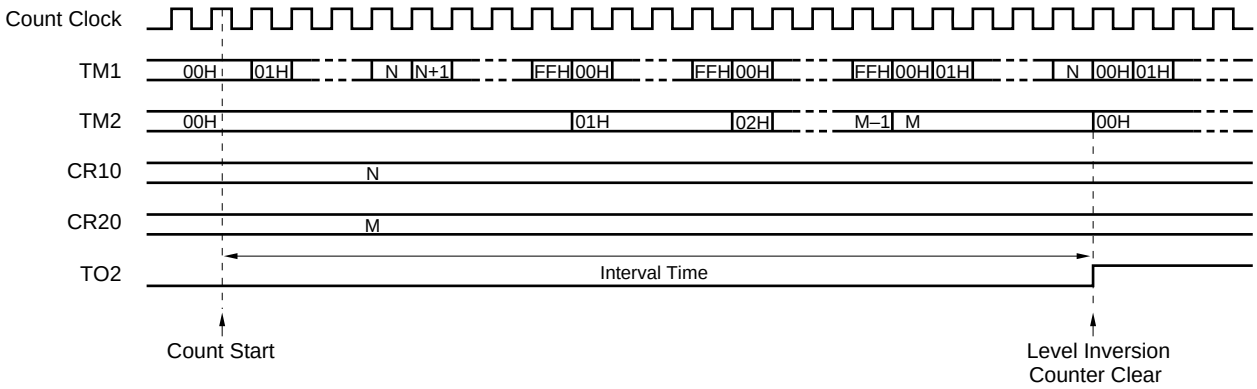
TCL13	TCL12	TCL11	TCL10	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	1	1	0	$2^2 \times 1/fx$ (400 ns)	$2^{18} \times 1/fx$ (26.2 ms)	$2^2 \times 1/fx$ (400 ns)
0	1	1	1	$2^3 \times 1/fx$ (800 ns)	$2^{19} \times 1/fx$ (52.4 ms)	$2^3 \times 1/fx$ (800 ns)
1	0	0	0	$2^4 \times 1/fx$ (1.6 μs)	$2^{20} \times 1/fx$ (104.9 ms)	$2^4 \times 1/fx$ (1.6 μs)
1	0	0	1	$2^5 \times 1/fx$ (3.2 μs)	$2^{21} \times 1/fx$ (209.7 ms)	$2^5 \times 1/fx$ (3.2 μs)
1	0	1	0	$2^6 \times 1/fx$ (6.4 μs)	$2^{22} \times 1/fx$ (419.4 ms)	$2^6 \times 1/fx$ (6.4 μs)
1	0	1	1	$2^7 \times 1/fx$ (12.8 μs)	$2^{23} \times 1/fx$ (838.9 ms)	$2^7 \times 1/fx$ (12.8 μs)
1	1	0	0	$2^8 \times 1/fx$ (25.6 μs)	$2^{24} \times 1/fx$ (1.7 s)	$2^8 \times 1/fx$ (25.6 μs)
1	1	0	1	$2^9 \times 1/fx$ (51.2 μs)	$2^{25} \times 1/fx$ (3.4 s)	$2^9 \times 1/fx$ (51.2 μs)
1	1	1	0	$2^{10} \times 1/fx$ (102.4 μs)	$2^{26} \times 1/fx$ (6.7 s)	$2^{10} \times 1/fx$ (102.4 μs)
1	1	1	1	$2^{12} \times 1/fx$ (409.6 μs)	$2^{28} \times 1/fx$ (26.8 s)	$2^{12} \times 1/fx$ (409.6 μs)

Remarks 1. f_X : main system clock oscillation frequency

2. TCL10 to TCL13 : bits 0 to 3 of timer clock selection register 1 (TCL1)

3. ( ) : at f_X = 10.0-MHz operation

★ **Figure 7-13. Square Wave Output Operation Timings**

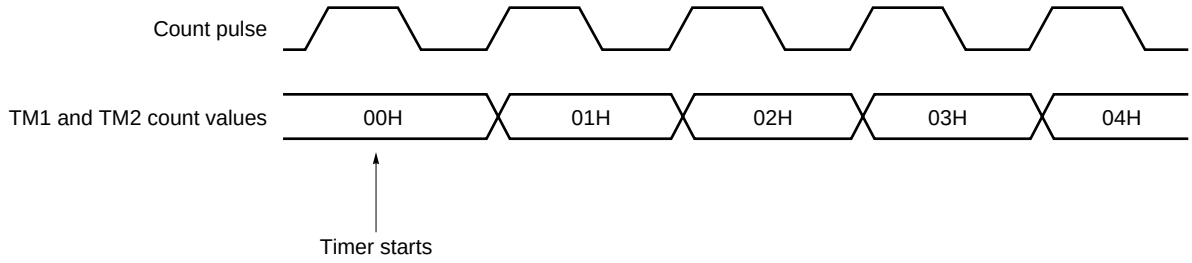


7.5 Notes on Using 8-Bit Timer/Event Counters

(1) Error on starting timer

An error of up to 1 clock occurs after the timer has been started until a coincidence signal is generated. This is because the 8-bit timer registers 1 and 2 (TM1 and TM2) are started in asynchronization with the count pulse.

Figure 7-14. Start Timing of 8-Bit Timer Register



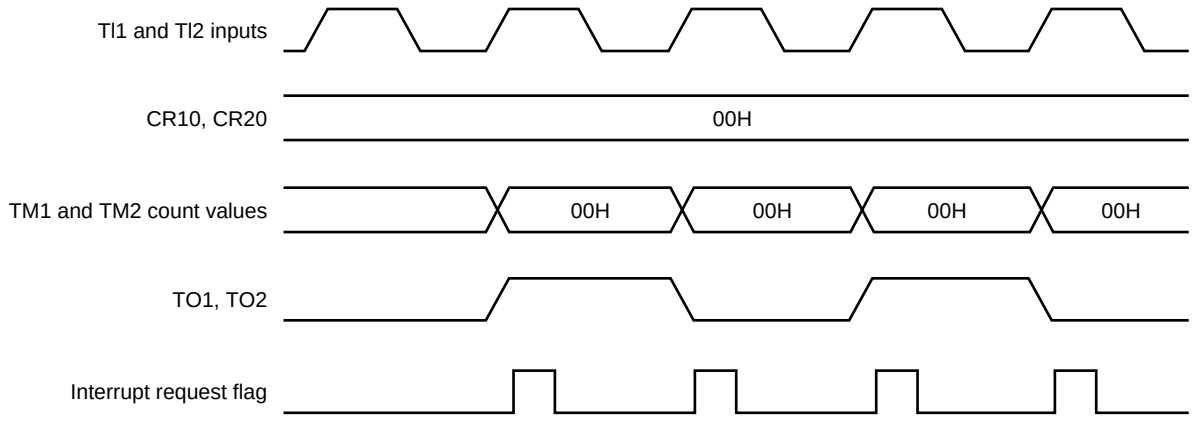
(2) Setting of 8-bit compare registers

The 8-bit compare registers (CR10 and CR20) can be set to 00H.

Therefore, one pulse can be counted when an 8-bit timer/event counter operates as an event counter.

When the two 8-bit timer/event counters are used together as a 16-bit timer/event counter, set bit 0 (TCE1) of the 8-bit timer mode control register (TMC1) to 0 and stop the timers, in order to write values to CR10 and CR20.

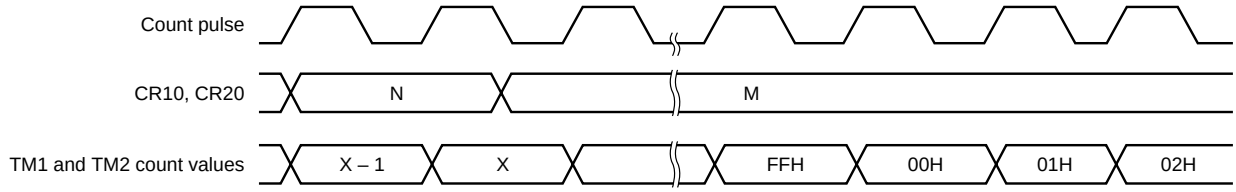
Figure 7-15. External Event Counter Operation Timing



(3) Operation after changing value of compare register during timer count operation

If a new value of an 8-bit compare register (CR10 or CR20) is less than the value of the corresponding 8-bit timer register (TM1 or TM2), TM1 and TM2 continue counting, overflow, and restart counting from 0. Therefore, if the new values of CR10 and CR20 (M) are less than their old values (N), it is necessary to restart the timers after changing the values of CR10 and CR20.

Figure 7-16. Timing after Changing Values of Compare Registers during Timer Count Operation



Remark N > X > M

[MEMO]

CHAPTER 8 WATCH TIMER

8.1 Functions of Watch Timer

The watch timer has the following functions:

- Watch timer
- Interval timer

The watch timer and interval timer can be used at the same time.

(1) Watch timer

The watch timer sets a flag (WTIF) at time intervals of 0.5 or 0.25 seconds by using the 32.768-kHz subsystem clock.

By using the 8.38-MHz main system clock, the flag (WTIF) is set at a time interval of 0.5 or 0.25 seconds. By using the 4.19-MHz (4.194304 MHz TYP.) main system clock, the flag (WTIF) is set at a time interval of 0.5 or 1 seconds. At the other frequencies, the flag is not set at a time interval of 0.5/0.25 or 0.5/1 seconds.

Caution When the 8.38-MHz or 4.19-MHz system clock is used, the time interval includes a slight error.

(2) Interval timer

When the watch timer is used as an interval timer, it generates an interrupt request (INTTM3) at time intervals set in advance.

Table 8-1. Interval Time of Interval Timer

Interval Time	At f_X = 10.0 MHz	At f_X = 8.38 MHz	At f_X = 4.19 MHz	At f_{XT} = 32.768 kHz
$2^4 \times 1/fw$	409.6 μs	489 μs	978 μs	488 μs
$2^5 \times 1/fw$	819.2 μs	978 μs	1.96 ms	977 μs
$2^6 \times 1/fw$	1.64 ms	1.96 ms	3.91 ms	1.95 ms
$2^7 \times 1/fw$	3.28 ms	3.91 ms	7.82 ms	3.91 ms
$2^8 \times 1/fw$	6.55 ms	7.82 ms	15.6 ms	7.81 ms
$2^9 \times 1/fw$	13.1 ms	15.6 ms	31.3 ms	15.6 ms

Remark f_X : main system clock oscillation frequency
f_{XT} : subsystem clock oscillation frequency
f_W : watch timer clock frequency

8.2 Configuration of Watch Timer

The watch timer consists of the following hardware:

Table 8-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits x 1
Control register	Timer clock select register 2 (TCL2) Watch timer mode control register (TMC2)

8.3 Registers Controlling Watch Timer

The following two registers control the watch timer:

- Timer clock select register 2 (TCL2)
- Watch timer mode control register (TMC2)

(1) Timer clock select register 2 (TCL2) (refer to **Figure 8-2.**)

This register sets the count clock of the watch timer.

TCL2 is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Remark TCL2 also has a function to set the count clock of the watchdog timer and the frequency of buzzer output, in addition to the function to set the count clock of the watch timer.

Figure 8-1. Block Diagram of Watch Timer

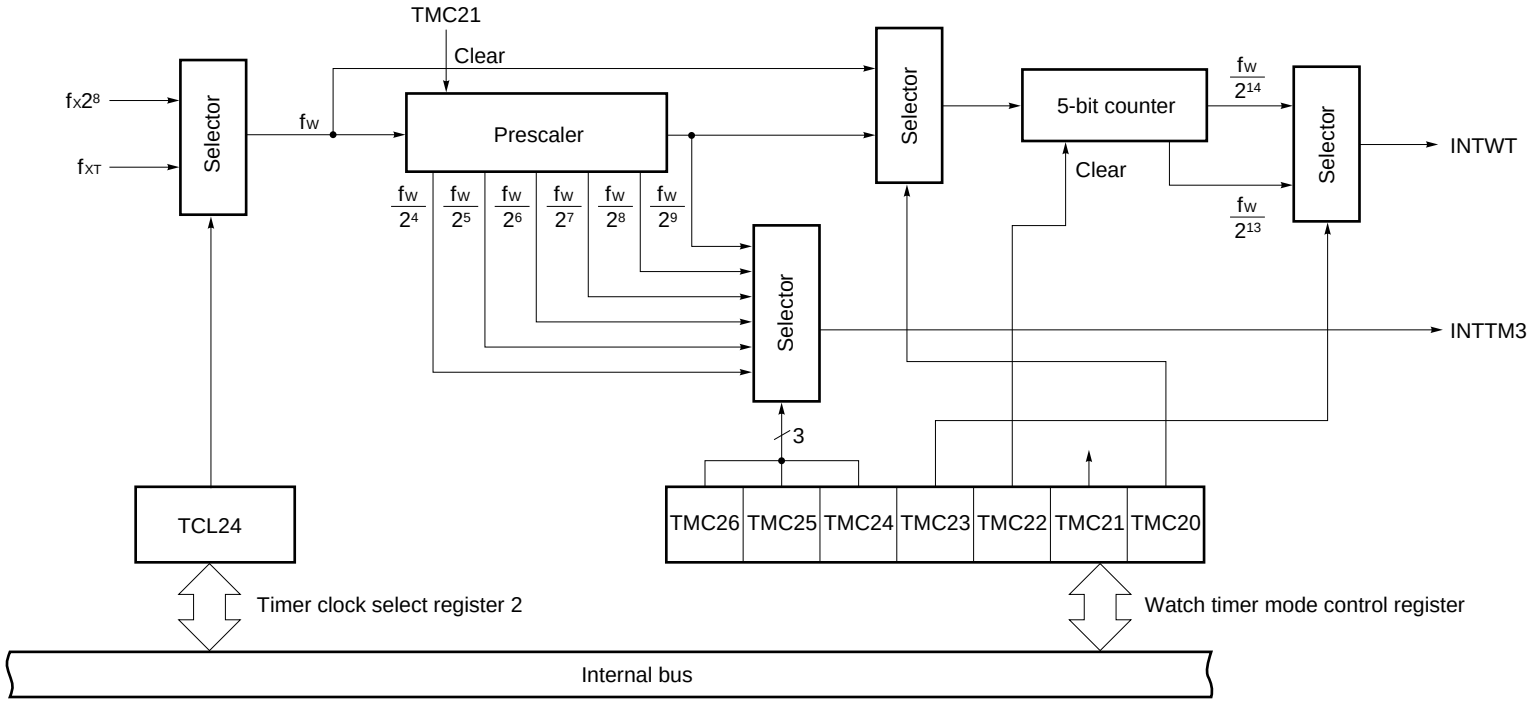


Figure 8-2. Format of Timer Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer
0	0	0	$f_X/2^4$ (625 kHz)
0	0	1	$f_X/2^5$ (313 kHz)
0	1	0	$f_X/2^6$ (156 kHz)
0	1	1	$f_X/2^7$ (78.1 kHz)
1	0	0	$f_X/2^8$ (39.1 kHz)
1	0	1	$f_X/2^9$ (19.5 kHz)
1	1	0	$f_X/2^{10}$ (9.8 kHz)
1	1	1	$f_X/2^{12}$ (2.4 kHz)

TCL24	Selects count clock of watch timer
0	$f_X/2^8$ (39.1 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Selects frequency of buzzer output
0	×	×	Disables buzzer output
1	0	0	$f_X/2^{10}$ (9.8 kHz)
1	0	1	$f_X/2^{11}$ (4.9 kHz)
1	1	0	$f_X/2^{12}$ (2.4 kHz)
1	1	1	Setting prohibited

Caution To write new data to TCL2, stop the timer operation once.

Remarks 1. f_X : main system clock oscillation frequency

2. f_{XT} : subsystem clock oscillation frequency

3. × : don't care

4. ( ) : at f_X = 10.0-MHz or f_{XT} = 32.768-kHz operation

(2) Watch timer mode control register (TMC2)

This register sets an operation mode of the watch timer, sets a set time of the watch flag, enables/disables the operation of the prescaler and 5-bit counter, and sets the interval time of the prescaler.

TMC2 is set by using a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 8-3. Format of Watch Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TMC2	0	TMC26	TMC25	TMC24	TMC23	TMC22	TMC21	TMC20	FF4AH	00H	R/W

TMC23	TMC20	Selects set time of watch flag
0	0	$2^{14}/f_W$ (0.5 s)
1		$2^{13}/f_W$ (0.25 s)
0	1	$2^5/f_W$ (977 μs)
1		$2^4/f_W$ (488 μs)

TMC21	Controls prescaler operation Note
0	Clears after operation stopped
1	Enables operation

TMC22	Controls 5-bit counter operation
0	Clears after operation stopped
1	Enables operation

TMC26	TMC25	TMC24	Selects interval time of prescaler
0	0	0	$2^4/f_W$ (488 μs)
0	0	1	$2^5/f_W$ (977 μs)
0	1	0	$2^6/f_W$ (1.95 ms)
0	1	1	$2^7/f_W$ (3.91 ms)
1	0	0	$2^8/f_W$ (7.81 ms)
1	0	1	$2^9/f_W$ (15.6 ms)
Others			Setting prohibited

Note Do not clear the prescaler frequently when using the watch timer.

Remarks **1.** f_W : watch timer clock frequency ($f_X/2^8$ or f_{XT})

2. ( ) : at f_W = 32.768-kHz operation

8.4 Operation of Watch Timer

8.4.1 Operation as watch timer

The watch timer operates at time intervals of 0.5 or 0.25 seconds when the 32.768-kHz subsystem clock or 8.38-MHz main system clock is used. When the 4.19-MHz main system clock is used, the watch timer can operate at time intervals of 0.5 or 1 seconds.

Caution When the 8.38-MHz or 4.19-MHz system clock is used, a slight error occurs.

When f_X = 8.38 MHz

$$\frac{2^8}{f_X} \times 2^{14} = \frac{2^{22}}{8.38 \times 10^6} = 0.5005136 \text{ ... (seconds)}$$

When f_X = 4.19 MHz

$$\frac{2^8}{f_X} \times 2^{13} = \frac{2^{21}}{4.19 \times 10^6} = 0.5005136 \text{ ... (seconds)}$$

When f_{XT} = 32.768 kHz

$$\frac{1}{f_{XT}} \times 2^{14} = \frac{2^{14}}{32.768 \times 10^3} = 0.50000 \text{ ... (seconds)}$$

When f_X = 10.0 MHz (this is not subject)

$$\frac{2^8}{f_X} \times 2^{14} = \frac{2^{22}}{10.0 \times 10^6} = 0.4194304 \text{ ... (seconds)}$$

The watch timer sets the test input flag (WTIF) to 1 at fixed time intervals. When WTIF is set to 1, the standby status (STOP/HALT mode) is released.

By setting bit 2 (TMC22) of the watch timer mode control register (TMC2) to 0, the 5-bit counter is cleared, and the count operation is stopped.

To operate the interval timer at the same time, set 0 to TMC22, so that the watch timer can be started from zero seconds (maximum error: 15.6-ms at 32.768-kHz operation).

8.4.2 Operation as interval timer

The watch timer also operates as an interval timer that repeatedly generates an interrupt request at time intervals specified by a count value set in advance.

The interval time can be selected by the bits 4 to 6 (TMC24 to TMC26) of the watch timer mode control register (TMC2).

Table 8-3. Interval Time of Interval Timer

TMC26	TMC25	TMC24	Interval Time	At f_X = 10.0 MHz	At f_X = 8.38 MHz	At f_X = 4.19 MHz	At f_{XT} = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	409.6 μs	489 μs	978 μs	488 μs
0	0	1	$2^5 \times 1/f_W$	819.2 μs	978 μs	1.96 ms	977 μs
0	1	0	$2^6 \times 1/f_W$	1.64 ms	1.96 ms	3.91 ms	1.95 ms
0	1	1	$2^7 \times 1/f_W$	3.28 ms	3.91 ms	7.82 ms	3.91 ms
1	0	0	$2^8 \times 1/f_W$	6.55 ms	7.82 ms	15.6 ms	7.81 ms
1	0	1	$2^9 \times 1/f_W$	13.1 ms	15.6 ms	31.3 ms	15.6 ms
Others			Setting prohibited				

Remark f_X : main system clock oscillation frequency

f_{XT} : subsystem clock oscillation frequency

f_W : watch timer clock frequency ($f_X/2^8$ or f_{XT})

TMC24 to TMC26 : bits 4 to 6 of watch timer mode control register (TMC2)

[MEMO]

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The watchdog timer has the following functions:

- Watchdog timer
- Interval timer

Caution Select the watchdog timer mode or interval timer mode by using the watchdog timer mode register (WDTM). (The watchdog timer and interval timer cannot be used simultaneously.)

(1) Watchdog timer mode

The watchdog timer is used to detect program runaway. When the runaway is detected, a non-maskable interrupt request or the $\overline{\text{RESET}}$ signal can be generated.

Table 9-1. Runaway Detection Time of Watchdog Timer

Runaway Detection Time	At f_X = 10.0 MHz	Runaway Detection Time	At f_X = 10.0 MHz
$2^{12} \times 1/f_X$	409.6 μs	$2^{16} \times 1/f_X$	6.55 ms
$2^{13} \times 1/f_X$	819.2 μs	$2^{17} \times 1/f_X$	13.1 ms
$2^{14} \times 1/f_X$	1.64 ms	$2^{18} \times 1/f_X$	26.2 ms
$2^{15} \times 1/f_X$	3.28 ms	$2^{20} \times 1/f_X$	104.9 ms

Remark f_X: main system clock oscillation frequency

(2) Interval timer mode

When the watchdog timer is used as an interval timer, it generates an interrupt request at time intervals set in advance.

Table 9-2. Interval Time

Interval Time	At f_X = 10.0 MHz	Interval Time	At f_X = 10.0 MHz
$2^{12} \times 1/f_X$	409.6 μs	$2^{16} \times 1/f_X$	6.55 ms
$2^{13} \times 1/f_X$	819.2 μs	$2^{17} \times 1/f_X$	13.1 ms
$2^{14} \times 1/f_X$	1.64 ms	$2^{18} \times 1/f_X$	26.2 ms
$2^{15} \times 1/f_X$	3.28 ms	$2^{20} \times 1/f_X$	104.9 ms

Remark f_X: main system clock oscillation frequency

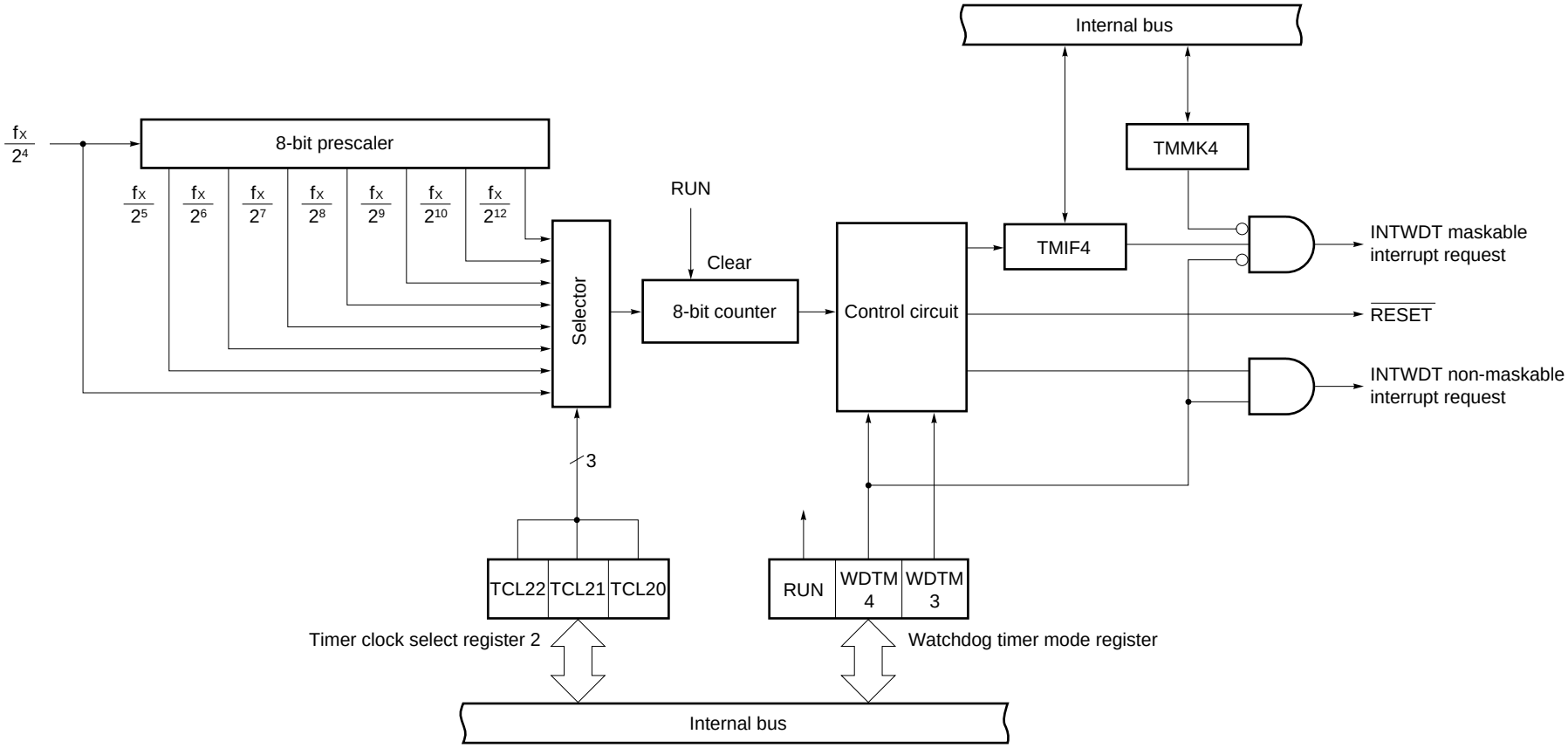
9.2 Configuration of Watchdog Timer

The watchdog timer consists of the following hardware:

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control register	Timer clock select register 2 (TCL2) Watchdog timer mode register (WDTM)

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Registers Controlling Watchdog Timer

The following two registers control the watchdog timer:

- Timer clock select register 2 (TCL2)
- Watchdog timer mode register (WDTM)

(1) Timer clock select register 2 (TCL2) (refer to Figure 9-2)

This register sets the count clock of the watchdog timer.

TCL2 is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Remark TCL2 also has a function to set the count clock of the watch timer and the frequency of buzzer output, in addition to the function to set the count clock of the watchdog timer.

Figure 9-2. Format of Timer Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer
0	0	0	$f_X/2^4$ (625 kHz)
0	0	1	$f_X/2^5$ (313 kHz)
0	1	0	$f_X/2^6$ (156 kHz)
0	1	1	$f_X/2^7$ (78.1 kHz)
1	0	0	$f_X/2^8$ (39.1 kHz)
1	0	1	$f_X/2^9$ (19.5 kHz)
1	1	0	$f_X/2^{10}$ (9.8 kHz)
1	1	1	$f_X/2^{12}$ (2.4 kHz)

TCL24	Selects count clock of watch timer
0	$f_X/2^8$ (39.1 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Selects frequency of buzzer output
0	×	×	Disables buzzer output
1	0	0	$f_X/2^{10}$ (9.8 kHz)
1	0	1	$f_X/2^{11}$ (4.9 kHz)
1	1	0	$f_X/2^{12}$ (2.4 kHz)
1	1	1	Setting prohibited

Caution To write new data to TCL2, stop the timer operation once.

Remarks 1. f_X : main system clock oscillation frequency

2. f_{XT} : subsystem clock oscillation frequency

3. × : don't care

4. ( ) : at f_X = 10.0-MHz or f_{XT} = 32.768-kHz operation

(2) Watchdog timer mode register (WDTM)

This register sets an operation mode of the watchdog timer, and enables/disables counting of the watchdog timer.

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 9-3. Format of Watchdog Timer Mode Register

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Selects operation mode of watchdog timer Note 1
0	×	Interval timer mode Note 2 (overflow and maskable interrupt request occur)
1	0	Watchdog timer mode 1 (overflow and non-maskable interrupt request occur)
1	1	Watchdog timer mode 2 (overflow occurs and reset operation started)

RUN	Selects operation of watchdog timer Note 3
0	Stops counting
1	Clears counter and starts counting

Notes 1. Once WDTM3 and WDTM4 have been set to 1, they cannot be cleared to 0 by software.

2. The watchdog timer starts operating as an interval timer as soon as the RUN bit has been set to 1.

3. Once RUN has been set to 1, it cannot be cleared to 0 by software. Therefore, when counting is started, it cannot be stopped by any means other than $\overline{\text{RESET}}$ input.

Cautions 1. When the watchdog timer is cleared by setting 1 to RUN, the actual overflow time is up to 0.5% shorter than the time set by the timer clock select register 2 (TCL2).

2. To use watchdog timer modes 1 and 2, set the WDTM4 bit to 1 after confirming that the interrupt request flag (TMIF4) is 0.

If WDTM4 is set when TMIF4 is 1, a non-maskable interrupt request occurs regardless of the contents of WDTM3.

Remark ×: don't care

9.4 Operation of Watchdog Timer

9.4.1 Operation as watchdog timer

The watchdog timer detects a runaway when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1.

The count clock (runaway detection time interval) of the watchdog timer can be selected by bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set runaway detection time interval after the watchdog timer has been started. By setting RUN to 1, the watchdog timer can be cleared and started counting. If RUN is not set to 1, and the runaway detection time is exceeded, the system is reset or a non-maskable interrupt request is generated by the value of bit 3 (WDTM3) of WDTM.

The watchdog timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set RUN to 1 before entering the STOP mode to clear the watchdog timer, and then execute the STOP instruction.

Cautions 1. The actual runaway detection time may be up to 0.5% shorter than the set time.

2. The count operation of the watchdog timer is stopped when the subsystem clock is selected as the CPU clock.

Table 9-4. Runaway Detection Time of Watchdog Timer

TCL22	TCL21	TCL20	Runaway Detection Time	At f_X = 10.0 MHz
0	0	0	$2^{12} \times 1/f_X$	409.6 μs
0	0	1	$2^{13} \times 1/f_X$	819.2 μs
0	1	0	$2^{14} \times 1/f_X$	1.64 ms
0	1	1	$2^{15} \times 1/f_X$	3.28 ms
1	0	0	$2^{16} \times 1/f_X$	6.55 ms
1	0	1	$2^{17} \times 1/f_X$	13.1 ms
1	1	0	$2^{18} \times 1/f_X$	26.2 ms
1	1	1	$2^{20} \times 1/f_X$	104.9 ms

Remark f_X : main system clock oscillation frequency

TCL20 to TCL22 : Bits 0 to 2 of timer clock selection register 2 (TCL2)

9.4.2 Operation as interval timer

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0, the watchdog timer also operates as an interval timer that repeatedly generates an interrupt request at time intervals specified by a count value set in advance.

Bits 0 through 2 (TCL20 through TCL22) of the timer clock select register 2 (TCL2) can be used to select a count clock (interval time). When bit 7 (RUN) of WDTM is set to 1, the watchdog timer starts operating as an interval timer.

In the interval timer mode, the interrupt mask flag (TMMK4) and priority specification flag (TMPR4) are valid, and a maskable interrupt request (INTWDT) can be generated. The default priority of INTWDT is set the highest of all the maskable interrupt requests.

The interval timer continues operation in the HALT mode, but stops in the STOP mode. Therefore, set bit 7 of WDTM (RUN) to 1 before entering the STOP mode to clear the interval timer, and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM has been set to 1 (when the watchdog timer mode is selected), the interval timer mode is not set, unless the $\overline{\text{RESET}}$ signal is input.

2. The interval time immediately after it has been set by WDTM may be up to 0.5% shorter than the set time.

3. The watchdog timer stops its counting operation when the subsystem clock is selected as the CPU clock.

Table 9-5. Interval Time of Interval Timer

TCL22	TCL21	TCL20	Interval Time	At f_X = 10.0 MHz
0	0	0	$2^{12} \times 1/f_X$	409.6 μs
0	0	1	$2^{13} \times 1/f_X$	819.2 μs
0	1	0	$2^{14} \times 1/f_X$	1.64 ms
0	1	1	$2^{15} \times 1/f_X$	3.28 ms
1	0	0	$2^{16} \times 1/f_X$	6.55 ms
1	0	1	$2^{17} \times 1/f_X$	13.1 ms
1	1	0	$2^{18} \times 1/f_X$	26.2 ms
1	1	1	$2^{20} \times 1/f_X$	104.9 ms

Remark f_X : main system clock oscillation frequency
TCL20 to TCL22 : Bits 0 to 2 of timer clock selection register 2 (TCL2)

CHAPTER 10 CLOCK OUTPUT CONTROL CIRCUIT

10.1 Function of Clock Output Control Circuit

The clock output control circuit outputs a carrier when a remote controller signal is transmitted, or a clock to be supplied to peripheral LSIs. It outputs the clock selected by the timer clock select register 0 (TCL0) from the PCL/P35 pin.

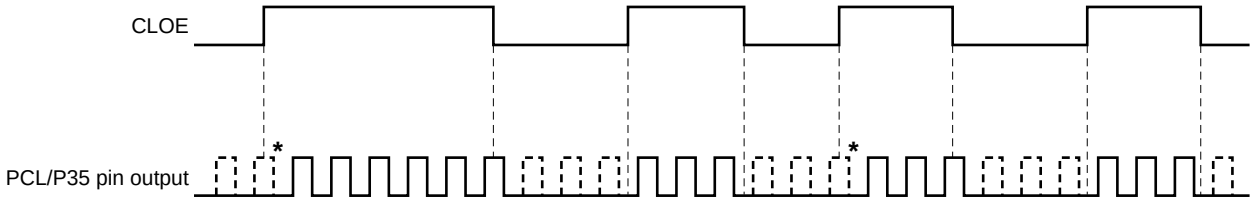
The clock pulse is output in the following procedure:

<1> Select the output frequency of the clock pulse by using bits 0 to 3 (TCL00 to TCL03) of TCL0 (output of the clock pulse is disabled).
<2> Set 0 to the output latch of the P35 pin.
<3> Set 0 to bit 5 (PM35) of the port mode register 3 (PM3) (to set the output mode).
<4> Set bit 7 (CLOE) of TCL0 to 1.

Caution When 1 is set to the output latch of the P35 pin, clock output cannot be used.

Remark The clock output control circuit is designed not to output a narrow pulse when clock output is enabled or disabled (refer to ***** in **Figure 10-1**).

Figure 10-1. Application Example of Remote Controller Output



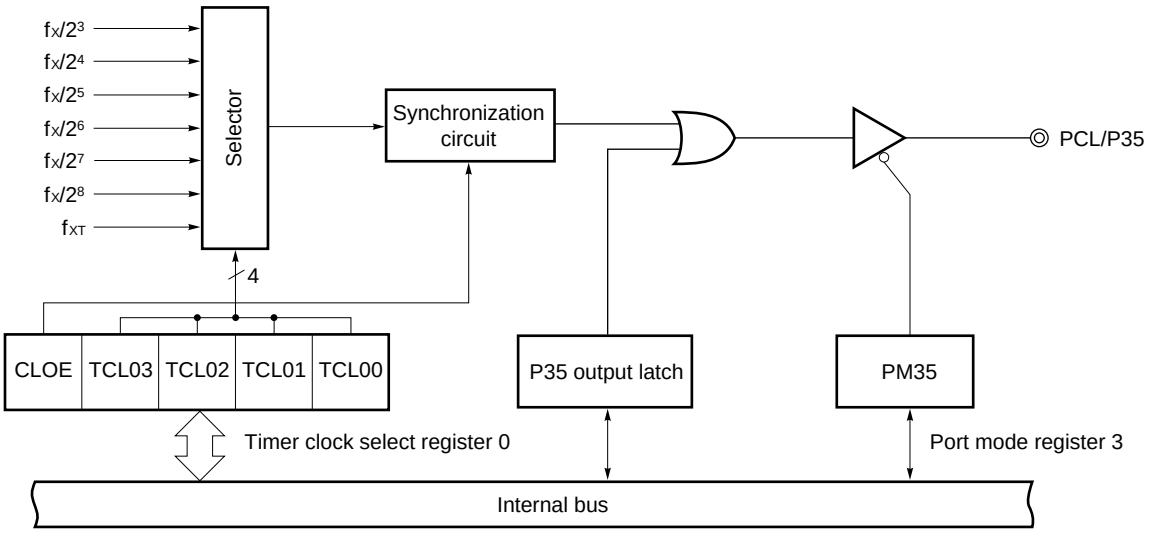
10.2 Configuration of Clock Output Control Circuit

The clock output control circuit consists of the following hardware:

Table 10-1. Configuration of Clock Output Control Circuit

Item	Configuration
Control register	Timer clock select register 0 (TCL0) Port mode register 3 (PM3) Port 3 (P3)

Figure 10-2. Block Diagram of Clock Output Control Circuit



10.3 Registers Controlling Clock Output Function

The following two registers control the clock output function:

- Timer clock select register 0 (TCL0)
- Port mode register 3 (PM3)

(1) Timer clock select register 0 (TCL0)

This register sets the clock for PCL output.

TCL0 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Remark TCL0 also has a function to set the count clock of the 16-bit timer register in addition to a function to set the clock for PCL output.

Figure 10-3. Format of Timer Clock Select Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TCL0	CLOE	TCL06	TCL05	TCL04	TCL03	TCL02	TCL01	TCL00	FF40H	00H	R/W

TCL03	TCL02	TCL01	TCL00	Selects clock of PCL output
0	0	0	0	f_{XT} (32.768 kHz)
0	1	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	0	$f_X/2^4$ (625 kHz)
1	0	0	1	$f_X/2^5$ (313 kHz)
1	0	1	0	$f_X/2^6$ (156 kHz)
1	0	1	1	$f_X/2^7$ (78.1 kHz)
1	1	0	0	$f_X/2^8$ (39.1 kHz)
Others				Setting prohibited

TCL06	TCL05	TCL04	Selects count clock of 16-bit timer register
0	0	0	TI0 (valid edge can be specified)
0	1	0	$f_X/2$ (5.0 MHz)
0	1	1	$f_X/2^2$ (2.5 MHz)
1	0	0	$f_X/2^3$ (1.25 MHz)
Others			Setting prohibited

CLOE	Controls PCL output
0	Disables output
1	Enables output

Cautions 1. The valid edge of the TI0/P00/INTP0 pin is set by the external interrupt mode register (INTM0). The frequency of a sampling clock is selected by the sampling clock select register (SCS).

2. To enable PCL output, set TCL00 to TCL03, and then set CLOE to 1 by using a 1-bit memory manipulation instruction.

3. Read the count value from TM0, not from the 16-bit capture register (CR01), when TI0 is used as the count clock of TM0.

4. To write data other than that already written to TCL0, stop the timer operation once.

Remarks 1. f_X : main system clock oscillation frequency

2. f_{XT} : subsystem clock oscillation frequency

3. TI0 : input pin of 16-bit timer/event counter

4. TM0 : 16-bit timer register

5. ( ) : at f_X = 10.0-MHz or f_{XT} = 32.768-kHz operation

(2) Port mode register 3 (PM3)

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

When the P35/PCL pin is used as a clock output function, set 0 to the PM35 bit of this register and the output latch of the P35 pin.

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

This register is set to FFH when the $\overline{\text{RESET}}$ signal is input.

Figure 10-4. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	Selects I/O mode of P3n pin (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

CHAPTER 11 BUZZER OUTPUT CONTROL CIRCUIT

11.1 Function of Buzzer Output Control Circuit

The buzzer output control circuit outputs a square wave with a frequency of 2.4 kHz, 4.9 kHz, or 9.8 kHz. The buzzer frequency selected by using the timer clock select register 2 (TCL2) is output from the BUZ/P36 pin.

The buzzer frequency is output in the following procedure:

<1> Select a buzzer output frequency by the bits 5 to 7 (TCL25 to TCL27) of TCL2.
<2> Set 0 to the output latch of the P36 pin.
<3> Set 0 to bit 6 (PM36) of port mode register 3 (PM3) (to set the output mode).

Caution When 1 is set to the output latch of the P36 pin, the buzzer output function cannot be used.

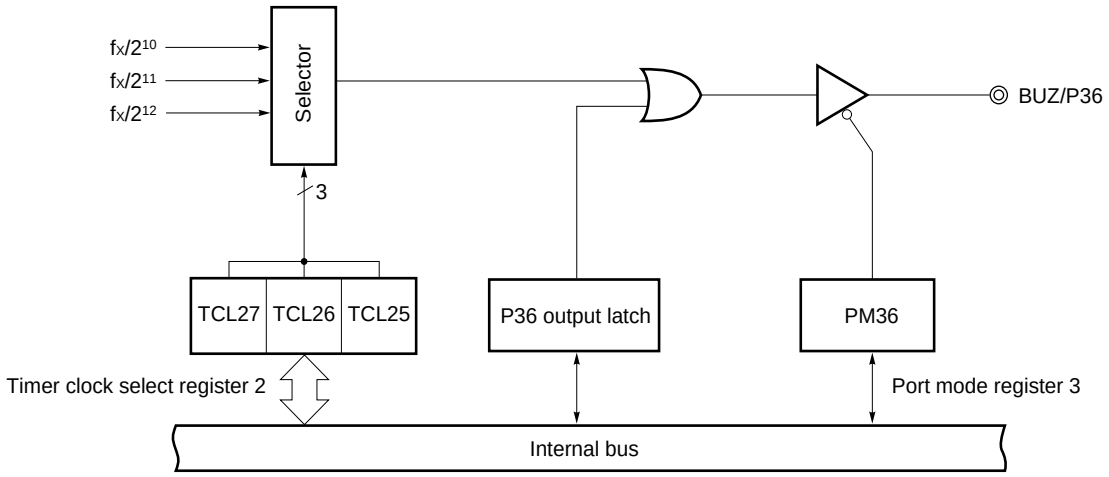
11.2 Configuration of Buzzer Output Control Circuit

The buzzer output control circuit consists of the following hardware:

Table 11-1. Configuration of Buzzer Output Control Circuit

Item	Configuration
Control register	Timer clock select register 2 (TCL2) Port mode register 3 (PM3) Port 3 (P3)

Figure 11-1. Block Diagram of Buzzer Output Control Circuit



11.3 Registers Controlling Buzzer Output Function

The following two types of registers control the buzzer output function:

- Timer clock select register 2 (TCL2)
- Port mode register 3 (PM3)

(1) Timer clock select register 2 (TCL2)

This register sets the frequency of buzzer output.

TCL2 is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Remark TCL2 also has a function to set the count clock of the watch timer and the count clock of the watchdog timer, in addition to the function to set the frequency of buzzer output.

Figure 11-2. Format of Timer Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Selects count clock of watchdog timer
0	0	0	$f_X/2^4$ (625 kHz)
0	0	1	$f_X/2^5$ (313 kHz)
0	1	0	$f_X/2^6$ (156 kHz)
0	1	1	$f_X/2^7$ (78.1 kHz)
1	0	0	$f_X/2^8$ (39.1 kHz)
1	0	1	$f_X/2^9$ (19.5 kHz)
1	1	0	$f_X/2^{10}$ (9.8 kHz)
1	1	1	$f_X/2^{12}$ (2.4 kHz)

TCL24	Selects count clock of watch timer
0	$f_X/2^8$ (39.1 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Selects frequency of buzzer output
0	×	×	Buzzer output disabled
1	0	0	$f_X/2^{10}$ (9.8 kHz)
1	0	1	$f_X/2^{11}$ (4.9 kHz)
1	1	0	$f_X/2^{12}$ (2.4 kHz)
1	1	1	Setting prohibited

Caution To write data other than that already written to TCL2, stop the timer operation once.

Remarks 1. f_X : main system clock oscillation frequency

2. f_{XT} : subsystem clock oscillation frequency

3. × : don't care

4. ( ) : at f_X = 10.0-MHz or f_{XT} = 32.768-kHz operation

(2) Port mode register 3 (PM3)

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

When the P36/BUZ pin is used as a buzzer output function, set 0 to the PM36 bit of this register and the output latch of the P36 pin.

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

This register is set to FFH when the $\overline{\text{RESET}}$ signal is input.

Figure 11-3. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

PM3n	Selects I/O mode of P3n pin (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

CHAPTER 12 A/D CONVERTER

12.1 Function of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of eight channels (ANI0 to ANI7) with a resolution of 8 bits.

This A/D converter is of successive approximation type, and the result of conversion is held by an 8-bit A/D conversion result register (ADCR).

A/D conversion can be started in the following two ways:

(1) Hardware start

Conversion is started by trigger input (INTP3).

(2) Software start

Conversion is started by setting the A/D conversion mode register (ADM).

Select one channel from ANI0 to ANI7 as an analog input, and start A/D conversion. In the case of hardware starting, an interrupt request (INTAD) is generated when an A/D conversion operation is completed, and the conversion operation stops. In the case of software starting, A/D conversion operations are repeated while generating an interrupt request (INTAD) each time one A/D conversion is completed.

Caution For the port pins having alternate (refer to 2.1 List of Pin Functions (1) Port pins), the following operations are prohibited. If these prohibitions are not observed, the operation within the total error rating during A/D conversion cannot be guaranteed.

<1> Changing the value of the output latch of the given port when it is used as a port

<2> Changing the output level of the pin used as an output even if it is not used as a port

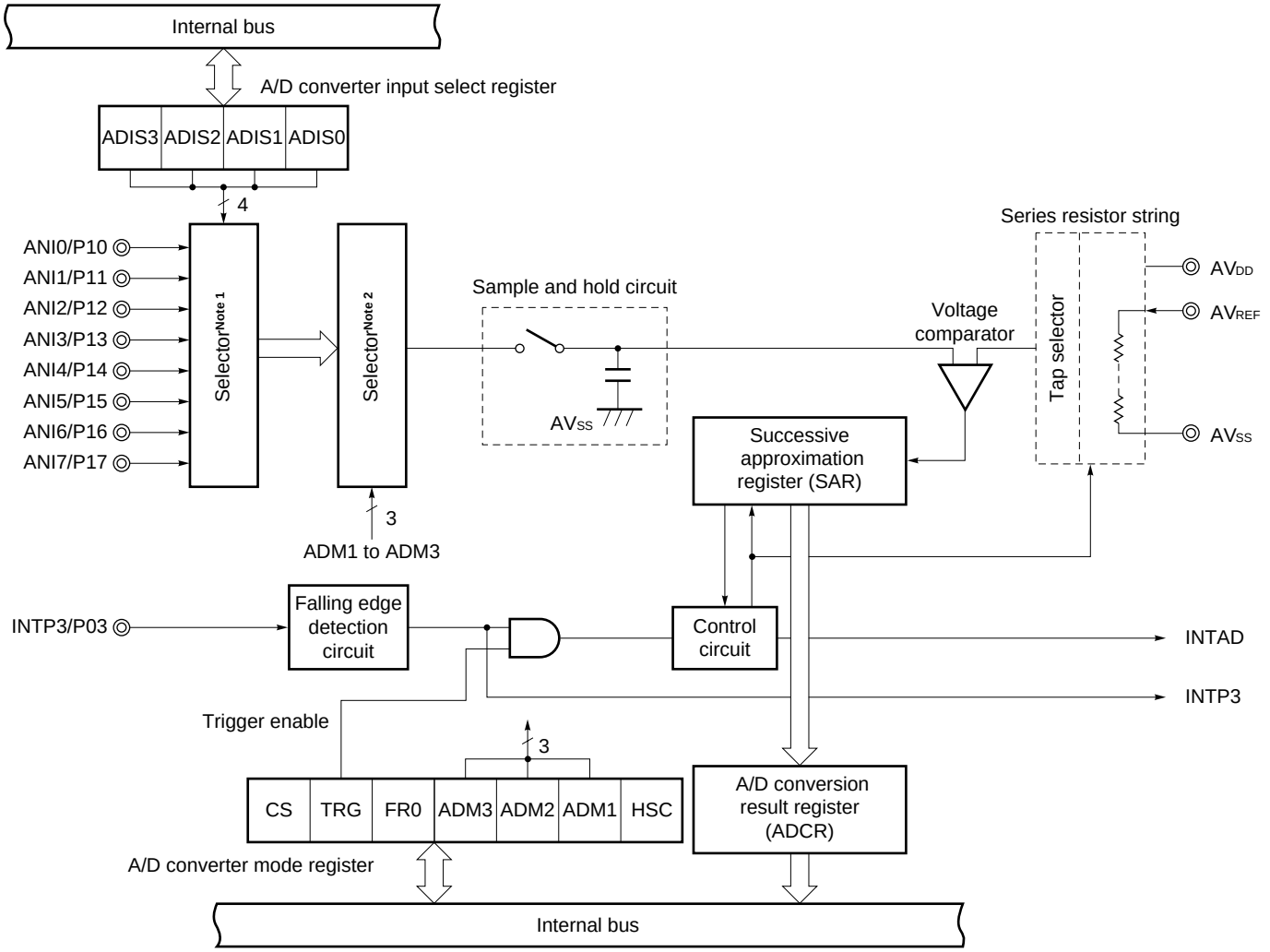
12.2 Configuration of A/D Converter

The A/D converter consists of the following hardware:

Table 12-1. Configuration of A/D Converter

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Control register	A/D converter mode register (ADM) A/D converter input select register (ADIS)
Register	Successive approximation register (SAR) A/D conversion result register (ADCR)

Figure 12-1. Block Diagram of A/D Converter



Notes **1.** Selector that selects the number of channels used for analog input.

2. Selector selecting a channel for A/D conversion.

(1) Successive approximation register (SAR)

This register compares the voltage value of analog input with the value of a voltage tap (compare voltage) from the series resistor string, and holds the result of the comparison starting from the most significant bit (MSB).

When the result is retained to the least significant bit (LSB) (end of A/D conversion), the contents of SAR are transferred to the A/D conversion result register (ADCR).

(2) A/D conversion result register (ADCR)

This register holds the result of A/D conversion result. Each time A/D conversion has been completed, the result of the conversion is loaded to this register from the successive approximation register (SAR).

ADCR can be read by an 8-bit memory manipulation instruction.

The contents of this register become undefined when the $\overline{\text{RESET}}$ signal is input.

(3) Sample and hold circuit

The sample and hold circuit samples analog input signals sequentially sent from the input circuit on a one-by-one basis, and sends the sampled signals to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input with the output voltage of the series resistor string.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS} and generates a voltage to be compared with an analog input.

(6) ANI0 to ANI7 pins

These are eight channels of analog input pins of the A/D converter. They input analog signals that are converted to digital values.

Pins other than those pins selected for analog input by A/D converter input select register (ADIS) can be used as I/O ports.

Cautions 1. Observe the specified input voltage range of ANI0 to ANI7. If a voltage of AV_{REF} or higher, or AV_{SS} or lower (even within the range of absolute maximum ratings) is applied to a channel, the converted value of that channel becomes undefined, or the converted value of the other channels may be affected.

2. ANI0 to ANI7 are also used as I/O port pins (PORT 1).

To use these pins as analog input pins, specify the input mode.

When A/D conversion is performed with any of ANI0 to ANI7 selected, do not execute the input instruction to PORT 1 while conversion is in progress; otherwise, the conversion resolution may be degraded.

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 under A/D conversion.

(7) AV_{REF} pin

This pin inputs a reference voltage to the A/D converter.

Based on the voltage applied between AV_{REF} and AV_{SS}, the signal input to ANI0 to ANI7 is converted into a digital signal.

In the standby mode, the current flowing through the series resistor string can be reduced by inputting a voltage of AV_{SS} level to the AV_{REF} pin.

Caution A series resistor string of about 10 kΩ is connected between the AV_{REF} and AV_{SS} pins. If the output impedance of the reference voltage source is high, therefore, the reference voltage error increases by connecting the impedance in parallel with the series resistor string between the AV_{REF} and AV_{SS} pins.

(8) AV_{SS} pin

This pin provides ground potential for the A/D converter and port circuitries. Be sure to use this pin at the same voltage as that on the V_{SS} pin always even when the A/D converter is not used.

(9) AV_{DD} pin

This pin supplies power to the A/D converter's analog unit and the port circuitries. Be sure to use this pin at the same voltage as that on the V_{DD} pin always even when the A/D converter is not used.

12.3 Registers Controlling A/D Converter

The following two registers control the A/D converter:

- A/D converter mode register (ADM)
- A/D converter input select register (ADIS)

(1) A/D converter mode register (ADM)

This register sets the channel of an analog input to be converted into a digital value, conversion time, starts/stops conversion operation, and sets an external trigger.

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

This register is set to 01H when the $\overline{\text{RESET}}$ signal is input.

Figure 12-2. Format of A/D Converter Mode Register

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Selects analog input channel
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

FR1	FR0	HSC	Selects A/D conversion time Note 1				
				At f_X = 10.0 MHz	At f_X = 8.38 MHz	At f_X = 5.0 MHz	At f_X = 4.19 MHz
0	0	1	160/f_X	Setting prohibited Note 2	19.1 μs	32.0 μs	38.1 μs
0	1	1	80/f_X	Setting prohibited Note 2	Setting prohibited Note 2	Setting prohibited Note 2	19.1 μs
1	0	0	100/f_X	Setting prohibited Note 2	Setting prohibited Note 2	20.0 μs	23.9 μs
1	0	1	200/f_X	20.0 μs	23.9 μs	40.0 μs	47.7 μs
Others			Setting prohibited				

TRG	Selects external trigger
0	No external trigger (software start mode)
1	Conversion started by external trigger (hardware start mode)

CS	Controls A/D conversion operation
0	Stops operation
1	Starts operation

Notes 1. Set these bits so that the A/D conversion time is 19.1 μs or more.

2. This setting is prohibited because the A/D conversion time is less than 19.1 μs.

Cautions 1. To reduce the power dissipation of the A/D converter in the standby mode, clear bit 7 (CS) to 0 to stop A/D conversion, and then execute the HALT or STOP instruction.

2. To resume A/D conversion that has been once stopped, clear the interrupt request flag (ADIF) to 0 and then start the A/D conversion.

Remark f_X: main system clock oscillation frequency

(2) A/D converter input select register (ADIS)

This register sets whether the ANI0/P10 to ANI7/P17 pins are used as the analog input channels of the A/D converter or as port pins. Pins not selected as analog input channels can be used as I/O port pins.

ADIS is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Cautions 1. Set an analog input channel in the following procedure:

<1> Set the number of analog input channels by ADIS.

<2> Select one channel for A/D conversion by using the A/D converter mode register (ADM) from the channels set as analog inputs by ADIS.

2. The channel set as an analog input by ADIS is not connected to the internal pull-up resistor regardless of the value of bit 1 (PUO1) of the pull-up resistor option register (PUO).

Figure 12-3. Format of A/D Converter Input Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	Selects number of analog input channels
0	0	0	0	None (P10 to P17)
0	0	0	1	1 channel (ANI0, P11 to P17)
0	0	1	0	2 channels (ANI0, ANI1, P12 to P17)
0	0	1	1	3 channels (ANI0 to ANI2, P13 to P17)
0	1	0	0	4 channels (ANI0 to ANI3, P14 to P17)
0	1	0	1	5 channels (ANI0 to ANI4, P15 to P17)
0	1	1	0	6 channels (ANI0 to ANI5, P16 to P17)
0	1	1	1	7 channels (ANI0 to ANI6, P17)
1	0	0	0	8 channels (ANI0 to ANI7)
Others				Setting prohibited

12.4 Operation of A/D Converter

12.4.1 Basic operation of A/D converter

<1> Set the number of analog input channels by using the A/D converter input select register (ADIS).

<2> Select one channel for A/D conversion by using the A/D converter mode register (ADM) from the channels set as analog inputs by ADIS.

<3> The voltage input to the selected analog input channel is sampled by the sample and hold circuit.

<4> When the voltage has been sampled for a specific time, the sample and hold circuit enters the hold status, and holds the input analog voltage until A/D conversion is completed.

<5> Bit 7 of the successive approximation register (SAR). The tap selector selects (1/2)AV_{REF} as the voltage tap of the series resistor string.

<6> The voltage difference between the voltage tap of the series resistor string and the analog input is compared by the voltage comparator. If the analog input is higher than (1/2)AV_{REF}, the MSB of SAR remains set. If it is less than (1/2)AV_{REF}, the MSB is reset.

<7> Next, bit 6 of SAR is automatically set, and the next voltage difference is compared. Here the voltage tap of the series resistor string is selected as follows, according to the value of bit 7 to which the result of the first comparison has been already set.

- Bit 7 = 1 : (3/4)AV_{REF}
- Bit 7 = 0 : (1/4)AV_{REF}

This voltage tap and analog input voltage are compared, and bit 6 of SAR is manipulated as follows, according to the result of the comparison:

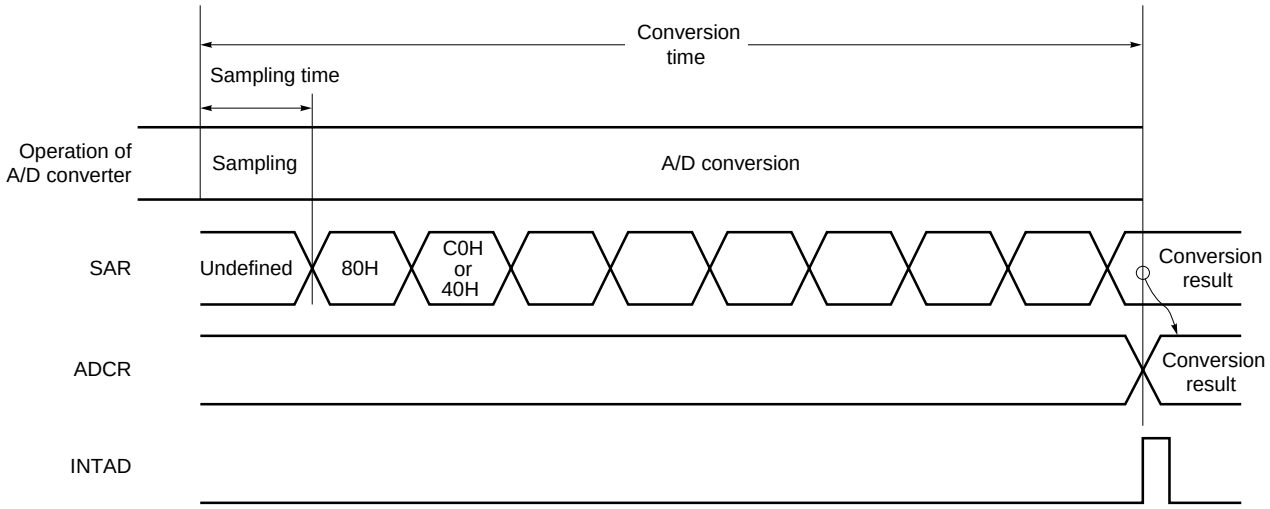
- Analog input voltage ≥ voltage tap : bit 6 = 1
- ★ • Analog input voltage < voltage tap : bit 6 = 0

<8> In this way, all the bits of SAR, including bit 0, are compared.

<9> When all the 8 bits of SAR have been compared, SAR holds the valid digital result whose values are transferred and latched to the A/D conversion result register (ADCR).

At the same time, an A/D conversion end interrupt request (INTAD) can be generated.

Figure 12-4. Basic Operation of A/D Converter



The A/D conversion is performed continuously, until bit 7 (CS) of the A/D converter mode register (ADM) is reset to 0 by software.

If the data of the ADM is rewritten during the A/D conversion, the conversion is initialized. If the CS bit is set to 1 at this time, conversion is performed again from the start.

The contents of the ADCR become undefined when the $\overline{\text{RESET}}$ signal is input.

12.4.2 Input voltage and conversion result

The relation between the analog voltage input to the analog input pins (ANI0 to ANI7) and A/D conversion result (value stored to A/D conversion result register (ADCR)) is as follows:

$$ADCR = INT\left(\frac{V_{IN}}{AV_{REF}} \times 256 + 0.5\right)$$

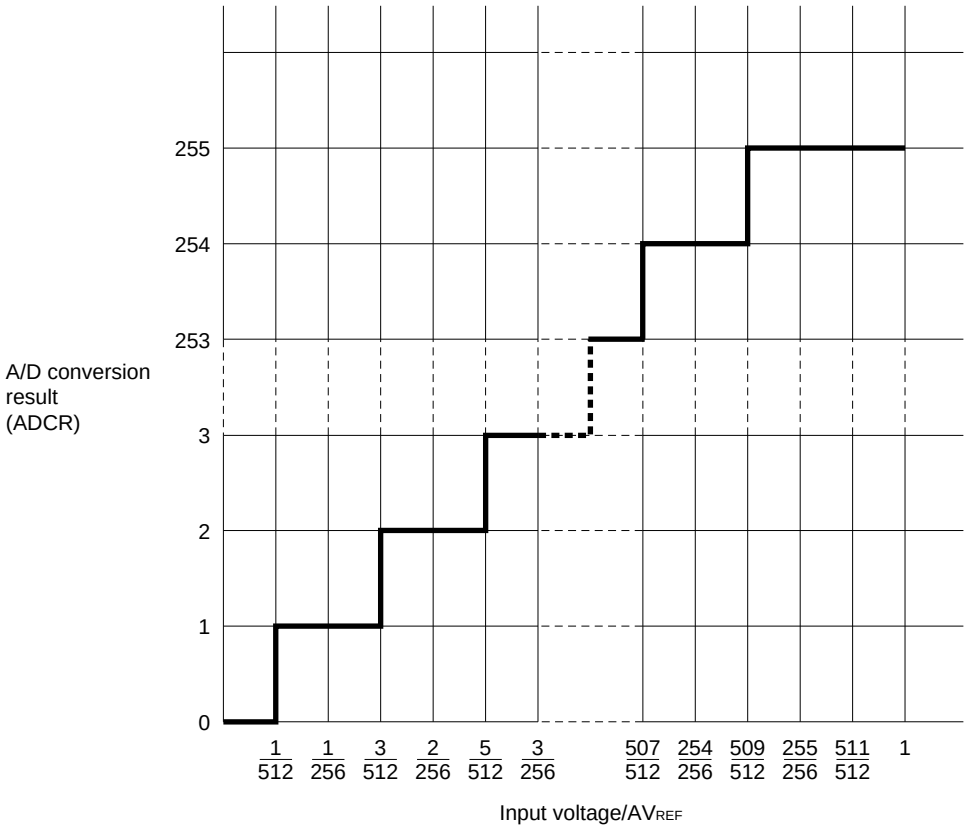
or,

$$(ADCR - 0.5) \times \frac{AV_{REF}}{256} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{REF}}{256}$$

Remark INT( ) : function returning integer of value in ( )
V_{IN} : analog input voltage
AV_{REF} : AV_{REF} pin voltage
ADCR : value of A/D conversion result register (ADCR)

Figure 12-5 shows the relations between the analog input voltage and A/D conversion result.

Figure 12-5. Relations between Analog Input Voltage and A/D Conversion Result



12.4.3 Operation mode of A/D converter

Select one analog input channel from ANI0 to ANI7 for A/D conversion by using the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and start the A/D conversion.

The A/D conversion can be started in the following two ways:

- Hardware start : Conversion is started by trigger input (INTP3).
- Software start : Conversion is started by setting ADM.

The result of the A/D conversion is stored in the A/D conversion result register (ADCR), and at the same time, an interrupt request signal (INTAD) is generated.

(1) A/D conversion operation by hardware start

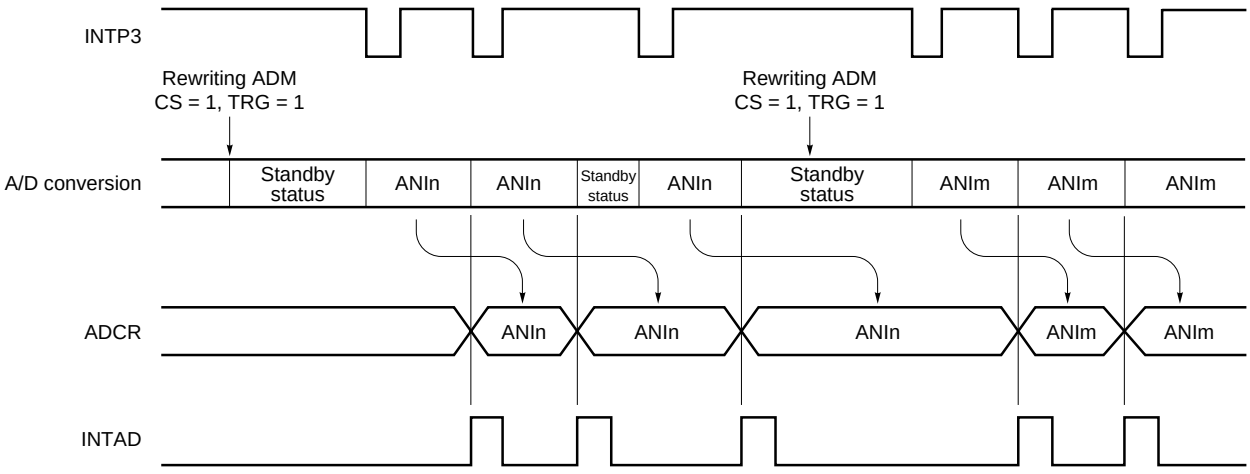
The A/D conversion stands by when both bits 6 (TRG) and 7 (CS) of A/D converter mode register (ADM) are set to 1. When an external trigger signal (INTP3) is input, the voltage applied to the analog input pin specified by bits 1 to 3 (ADM1 to ADM3) of ADM is converted into a digital value.

When the A/D conversion has been completed, the result of the conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. Once the A/D conversion has been started and when one A/D conversion has been completed, the next A/D conversion is not started unless a new external trigger signal is input.

If data whose CS is 1 is written again to ADM during A/D conversion, the AD conversion under execution is stopped, and stands by until a new external trigger signal is input. When the external trigger signal is input, A/D conversion is performed again from the start.

When 0 is written to the CS bit of ADM during A/D conversion, the conversion is immediately stopped.

Figure 12-6. A/D Conversion by Hardware Start



Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

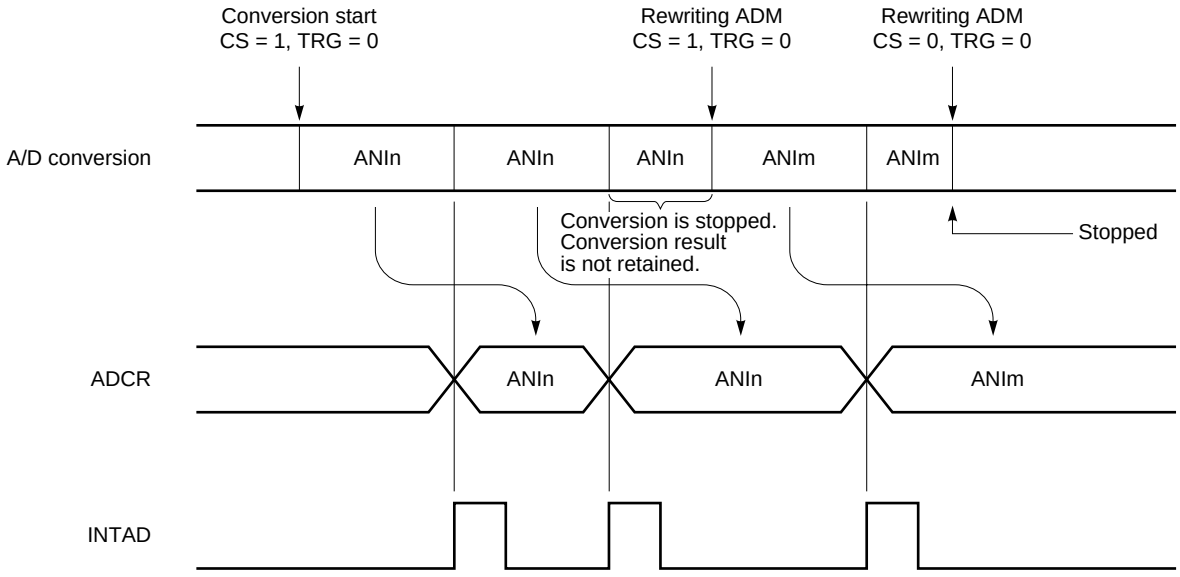
(2) A/D conversion by software start

By setting bit 6 (TRG) of the A/D converter mode register (ADM) to 0 and setting bit 7 (CS) to 1, the voltage applied to the analog input pin specified by bits 1 to 3 (ADM1 to ADM3) of ADM is converted into digital values. When the A/D conversion has been completed, the result of the conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. When the A/D conversion has been started once, and one A/D conversion has been completed, the next A/D conversion is immediately started. In this way, A/D conversion is repeatedly executed until new data is written to ADM.

If data whose CS is 1 is written again to ADM during A/D conversion, the conversion under execution is stopped, and the A/D conversion of the newly written data is started.

If data whose CS is 0 is written to ADM during A/D conversion, the conversion is immediately stopped.

Figure 12-7. A/D Conversion by Software Start



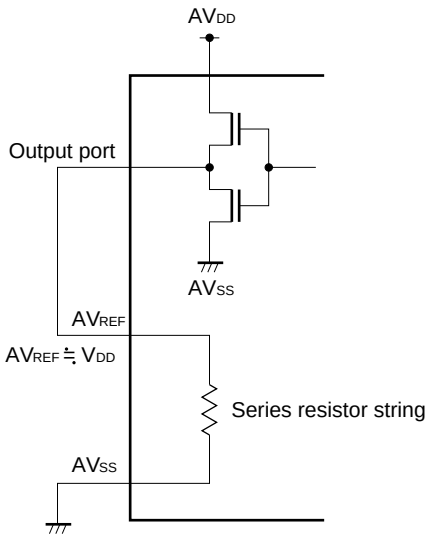
Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

12.5 Notes on A/D Converter

(1) Current consumption in standby mode

The A/D converter operates on the main system clock. Therefore, its operation is stopped in the STOP mode, or in the HALT mode on the subsystem clock. Even at this time, a current flows into the AV_{REF} pin, and it is necessary to cut this current in order to reduce the current consumption of the entire system. In the case shown in Figure 12-8 below, the current consumption can be reduced if a low level is output to the output ports in the standby mode. However, the actual AV_{REF} voltage is not accurate, and therefore, the conversion value itself is not accurate but can be used for relative comparison only.

Figure 12-8. Example of Reducing Current Consumption in Standby Mode



μPD78014H Subseries

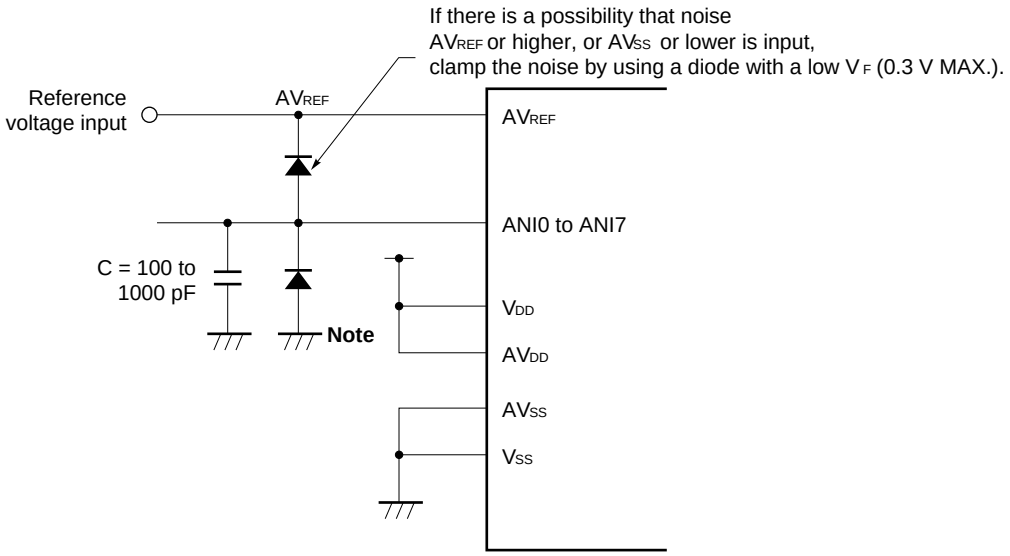
(2) ANI0 to ANI7 input range

Observe the rated range of the ANI0 to ANI7 input voltage. If a voltage of AV_{REF} or higher, or 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) Countermeasures against noise

To keep the resolution of 8 bits, noise superimposed on the AV_{REF} and ANI0 to ANI7 pins must be suppressed as much as possible. The higher the output impedance of the analog input source, the greater the effect. To suppress noise, connecting an external capacitor as shown in Figure 12-9 is recommended.

Figure 12-9. Processing Analog Input Pin



Note In order to lower the level of EMI noise, supply different supply voltages to V_{DD} and AV_{DD}, and connect V_{SS} and AV_{SS} to different ground lines.

(4) ANI0/P10 to ANI7/P17

The analog input pins (ANI0 to ANI7) are also used as I/O port pins (PORT 1).

To use these pins as the analog input pins, specify the input mode.

When A/D conversion is performed with any of ANI0 to ANI7 selected, do not execute the input instruction to PORT 1 while conversion is in progress; otherwise, the conversion resolution may be degraded.

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 adjacent pins to the pin under A/D conversion.

(5) Input impedance to AV_{REF} pin

A series resistor string of about 10 kΩ is connected between the AV_{REF} and AV_{SS} pins.

If the output impedance of the reference voltage source is high, therefore, an error of the reference voltage increases by connecting the impedance in parallel with the series resistor string between the AV_{REF} and AV_{SS} pins.

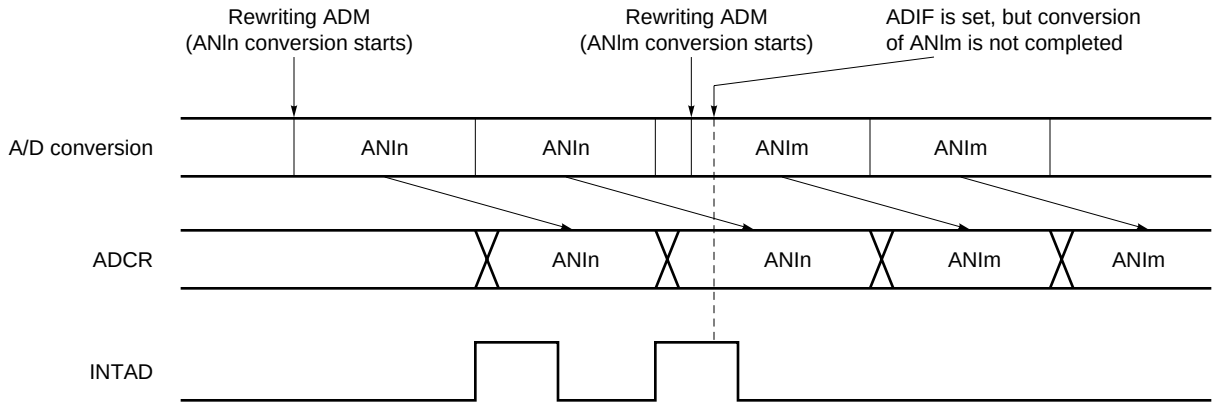
(6) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even when the contents of the A/D converter mode register (ADM) are changed.

When the analog input pin is changed during A/D conversion, therefore, the chances are that the A/D conversion result of the old analog input. At this time, interrupt request flags are set immediately before the contents of ADM are rewritten. Consequently, ADIF may be set even if A/D conversion for the newly specified analog input pin has not yet been completed when ADIF is read immediately after ADM has been rewritten (refer to **Figure 12-10**).

To resume A/D conversion that has been once stopped, clear ADIF before resuming the conversion.

Figure 12-10. A/D Conversion End Interrupt Request Generation Timing



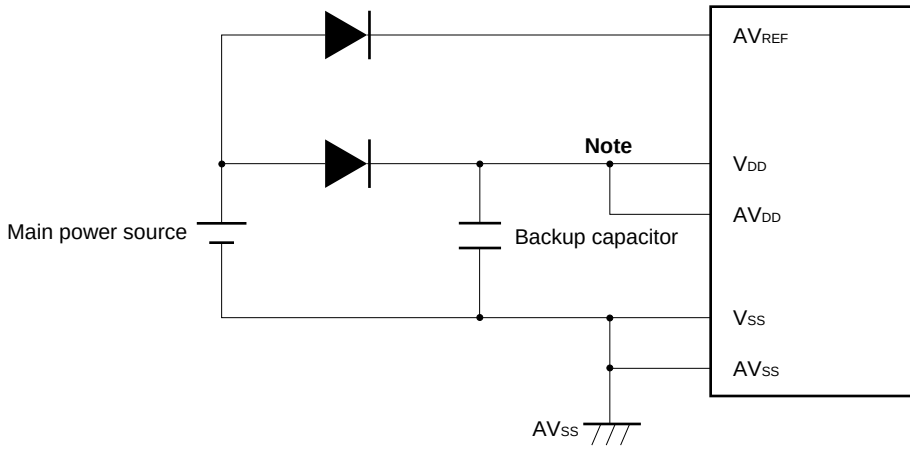
Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

(7) AV_{DD} pin

The AV_{DD} pin is the power supply pin to the analog circuit and supplies power to the input circuit of ANI0/P10 to ANI7/P17.

Therefore, even in the application which can be switched over to backup power source, be sure to apply the same voltage as V_{DD} as shown in Figure 12-11.

Figure 12-11. Processing of AV_{DD} Pin



Note In order to lower the level of EMI noise, supply different supply voltages to V_{DD} and AV_{DD}, and connect V_{SS} and AV_{SS} to different ground lines.

(8) Port operations during A/D conversion

For the port pins having alternate functions (refer to **2.1 (1) Port pins**), the following operations are prohibited. If these prohibitions are not observed, the total error-free operation during A/D conversion cannot be guaranteed.

<1> Changing the value of the output latch of the given port when it is used as a port

<2> Changing the output level of the pin used as an output even if it is not used as a port

CHAPTER 13 SERIAL INTERFACE CHANNEL 0

The μPD78014H Subseries is provided with two channels of clocked serial interfaces.

The differences between channels 0 and 1 are as indicated in the table below (for the details of serial interface channel 1, refer to **CHAPTER 14 SERIAL INTERFACE CHANNEL 1**).

Table 13-1. Differences between Channels 0 and 1

Serial Transfer Mode		Channel 0	Channel 1
3-wire serial I/O	Clock selection	$f_X/2^{2}$[Note], $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, $f_X/2^8$, $f_X/2^9$, external clock, TO2 output	$f_X/2^{2}$[Note], $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, $f_X/2^8$, $f_X/2^9$, external clock, TO2 output
	Transfer method	MSB first/LSB first selectable	MSB first/LSB first selectable Automatic transmission/reception function
	Transfer end flag	Serial interface channel 0 transfer end interrupt request flag (CSIIF0)	Serial interface channel 1 transfer end interrupt request flag (CSIIF1 and TRF) Not available
SBI (serial bus interface)		Available	
2-wire serial I/O			

Note Can be set only when the main system clock oscillates at 4.19 MHz or less.

13.1 Functions of Serial Interface Channel 0

Serial interface channel 0 has the following four modes:

Table 13-2. Differences in Modes of Serial Interface Channel 0

Operation Mode	Pins Used	Features	Applications
Operation stop mode	–	• Mode used when no serial transfer is performed • Power dissipation can be reduced.	–
3-wire serial I/O mode	$\overline{SCK0}$ (serial clock), SO0 (serial output), SI0 (serial input)	• Short data transfer processing time because independent input and output lines are used, allowing simultaneous transmission and reception. • MSB/LSB selectable for first bit of 8-bit data by serial transfer.	Useful for connecting peripheral I/Os and display controllers with conventional clocked serial interface such as 75X/XL Series, 78K Series, and 17K Series
SBI mode	$\overline{SCK0}$ (serial clock), SB0 or SB1 (serial data bus)	• Because serial bus consists of two signal lines, number of ports can be reduced and wiring distance on PWB can be shortened even when plural microcontrollers are connected. • High-speed serial interface conforming to NEC's standard bus format. • Serial bus has address, command, data information. • Wake-up function for hand-shake, output function of acknowledge and busy signals available. • Any data transfer format can be supported by the program	
2-wire serial I/O mode	$\overline{SCK0}$ (serial clock), SB0 or SB1 (serial data bus)	and lines for hand-shake conventionally necessary for connecting multiple devices can be eliminated.	

Caution During the operation of serial interface channel 0, do not change the operation mode being used (3-wire serial I/O, 2-wire serial I/O, or SBI).
When changing the operation mode, stop the serial operation beforehand.

13.2 Configuration of Serial Interface Channel 0

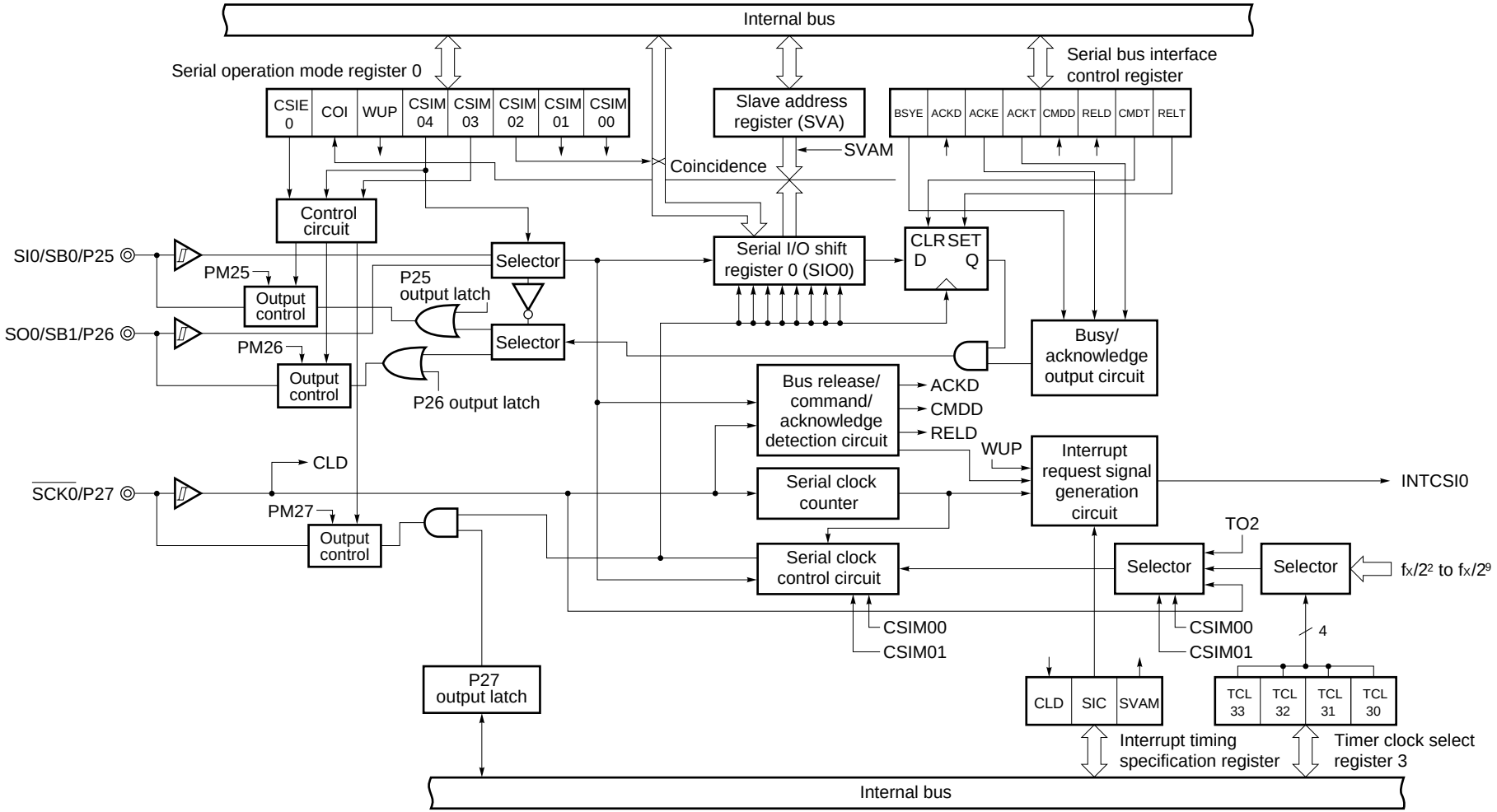
Serial interface channel 0 consists of the following hardware:

Table 13-3. Configuration of Serial Interface Channel 0

Item	Configuration
Register	Serial I/O shift register 0 (SIO0) Slave address register (SVA)
Control register	Timer clock select register 3 (TCL3) Serial operation mode register 0 (CSIM0) Serial bus interface control register (SBIC) Interrupt timing specification register (SINT) Port mode register 2 (PM2)[Note] Port 2 (P2)

Note Refer to **Figure 4-6 Block Diagram of P20, P21, P23 to P26** and **Figure 4-7 Block Diagram of P22, P27**.

Figure 13-1. Block Diagram of Serial Interface Channel 0



Remark The output control selects CMOS output or N-ch open drain output.

(1) Serial I/O shift register 0 (SIO0)

This 8-bit register converts parallel data into serial data, and transmits/receives serial data (shift operation) in synchronization with the serial clock.

SIO0 is set by an 8-bit memory manipulation instruction.

When the bit 7 (CSIE0) of the serial operation mode register 0 (CSIM0) is 1, the serial operation is started when data is written to SIO0.

The data written to SIO0 is output to the serial output line (SO0) or serial data bus (SB0/SB1) for transmission.

When data is received, it is read from the serial input line (SI0) or SB0/SB1 to SIO0.

In the SBI mode and 2-wire serial I/O mode bus configuration, the input and output pins are shared. Therefore, the device that is to receive data must write FFH to SIO0 in advance (except, however, when an address is received by setting 1 to bit 5 (WUP) of CSIM0).

In the SBI mode, the busy status can be released by writing data to SIO0. In this case, bit 7 (BSYE) of the serial bus interface control register (SBIC) is not cleared to 0.

The contents of SIO0 become undefined when the $\overline{\text{RESET}}$ signal is input.

(2) Slave address register (SVA)

This 8-bit register sets the value of a slave address when the microcontroller is connected to the serial bus as a slave device. This register is not used in the 3-wire serial I/O mode.

SVA is set by an 8-bit memory manipulation instruction.

The master outputs a slave address to the slaves connected to it, to select a specific slave. The slave address output by the master and the value of the SVA are compared by an address comparator. If the two addresses coincide, the slave is selected. At this time, bit 6 (COI) of the serial operation mode register 0 (CSIM0) is set to 1.

By setting bit 4 (SVAM) of the interrupt timing specification register (SINT), it is possible to compare addresses with 7-bit data which is created by masking LSB of 8-bit data.

If no coincidence is detected when the address is received, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. In the SBI mode, the wake-up function can be used by setting bit 5 (WUP) of CSIM0 to 1. In this case, an interrupt request signal (INTCI0) is generated only when the slave address output by the master coincides with the value of SVA. This is interrupt signal indicates that the master requests communication. If bit 5 (SIC) of the interrupt timing specification register (SINT) is set to 1, the wake-up function cannot be used even if WUP is set to 1 (the interrupt request signal is generated on detection of bus release). Clear SIC to 0 when using the wake-up function.

When the microcontroller transmits data as the master or a slave in the SBI mode or 2-wire serial I/O mode, errors can be detected by using the SVA value.

The contents of SVA become undefined when the $\overline{\text{RESET}}$ signal is input.

(3) SO0 latch

This latch retains the levels of SI0/SB0/P25 and SO0/SB1/P26 pins. It can also be directly controlled by software. In the SBI mode, this latch is set when the eighth serial clock has been input.

(4) Serial clock counter

This counter counts the serial clocks output or input during transmission/reception operation, and checks whether 8-bit data has been transmitted/received.

(5) Serial clock control circuit

This circuit controls supply of the serial clock to the serial I/O shift register 0 (SIO0). When the internal system clock is used, it also controls the clock output to the $\overline{\text{SCK0}}$/P27 pin.

(6) Interrupt request signal generation circuit

This circuit controls generation of an interrupt request signal. It generates an interrupt request signal in the following cases:

- **In 3-wire serial I/O mode and 2-wire serial I/O mode**

  Generates the interrupt request signal each time eight serial clocks have been counted.

- **In SBI mode**

  When WUP[Note] is 0 ..... Generates the interrupt request signal each time eight serial clocks have been counted.

  When WUP[Note] is 1 ..... Generates the interrupt request signal when the values of the serial I/O shift register 0 (SIO0) and slave address register (SVA) coincide after an address has been received.

Note WUP : wake-up function specification bit. Bit 5 of serial operation mode register 0 (CSIM0). Clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0 when using the wake-up function (WUP = 1).

(7) Busy/acknowledge output circuit and bus release/command/acknowledge detection circuit

These circuits output and detect various control signals in the SBI mode.

They do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

13.3 Registers Controlling Serial Interface Channel 0

The following four types of registers control serial interface channel 0:

- Timer clock select register 3 (TCL3)
- Serial operation mode register 0 (CSIM0)
- Serial bus interface control register (SBIC)
- Interrupt timing specification register (SINT)

(1) Timer clock select register 3 (TCL3) (refer to Figure 13-2)

This register sets the serial clock of serial interface channel 0.

TCL3 is set by an 8-bit memory manipulation instruction.

This register is set to 88H when the $\overline{\text{RESET}}$ signal is input.

Remark TCL3 also has a function to set the serial clock of serial interface channel 1 in addition to the function to set the serial clock of serial interface channel 0.

(2) Serial operation mode register 0 (CSIM0) (refer to Figure 13-3)

This register sets the serial clock and operation mode of serial interface channel 0, enables/disables the operation of the interface, sets the wake-up function, and indicates the coincidence signal of the address comparator.

CSIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Caution During the operation of serial interface channel 0, do not change the operation mode being used (3-wire serial I/O, 2-wire serial I/O, or SBI).
When changing the operation mode, stop the serial operation beforehand.

Figure 13-2. Format of Timer Clock Select Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Selects serial clock of serial interface channel 0
0	1	1	0	$f_X/2^2$ Note
0	1	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	0	$f_X/2^4$ (625 kHz)
1	0	0	1	$f_X/2^5$ (313 kHz)
1	0	1	0	$f_X/2^6$ (156 kHz)
1	0	1	1	$f_X/2^7$ (78.1 kHz)
1	1	0	0	$f_X/2^8$ (39.1 kHz)
1	1	0	1	$f_X/2^9$ (19.5 kHz)
Others				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Selects serial clock of serial interface channel 1
0	1	1	0	$f_X/2^2$ Note
0	1	1	1	$f_X/2^3$ (1.25 MHz)
1	0	0	0	$f_X/2^4$ (625 kHz)
1	0	0	1	$f_X/2^5$ (313 kHz)
1	0	1	0	$f_X/2^6$ (156 kHz)
1	0	1	1	$f_X/2^7$ (78.1 kHz)
1	1	0	0	$f_X/2^8$ (39.1kHz)
1	1	0	1	$f_X/2^9$ (19.5 kHz)
Others				Setting prohibited

Note Can be set only when the main system clock oscillates at 4.19 MHz or less.

Caution To write data other than that already written to TCL3, stop the serial transfer once.

Remarks 1. f_X : main system clock oscillation frequency

2. ( ) : at f_X = 10.0-MHz operation

Figure 13-3. Format of Serial Operation Mode Register 0 (1/2)

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Selects clock of serial interface channel 0
0	×	Clock externally input to $\overline{SCK0}$ pin
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation mode	First bit	SI0/SB0/P25 pin function	SO0/SB1/P26 pin function	$\overline{SCK0}$/P27 pin function
0	×	0	1	×	0	0	0	1	3-wire serial I/O mode	MSB	SI0 Note 2 (input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS I/O)
		1								LSB			
1	0	0	Note 3 ×	Note 3 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open drain I/O)	$\overline{SCK0}$ (CMOS I/O)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0 (N-ch open drain I/O)	P26 (CMOS I/O)	
1	1	0	Note 3 ×	Note 3 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open drain I/O)	$\overline{SCK0}$ (N-ch open drain I/O)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0 (N-ch open drain I/O)	P26 (CMOS I/O)	

R/W

WUP	Controls wake-up function Note 4
0	Generates interrupt request signal in all modes each time serial transfer is executed
1	Generates interrupt request signal when address received after bus has been released in SBI mode (when CMDD = RELD = 1) coincides with data of slave address register (SVA)

Notes 1. Bit 6 (COI) is a read-only bit.

2. This pin can be used as P25 (CMOS input) when used only for transmission.

3. This pin can be freely used for port function.

4. Clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0 when using the wake-up function (WUP = 1).

Remark × : don't care
PM×× : port mode register
P×× : output latch of a port

Figure 13-3. Format of Serial Operation Mode Register 0 (2/2)

R	COI	Slave address comparison result flag Note 1
	0	Data of slave address register (SVA) does not coincide with data of serial I/O shift register 0 (SIO0)
	1	Data of slave address register (SVA) coincides with data of serial I/O shift register 0 (SIO0)

R/W	CSIE0	Controls operation of serial interface channel 0 Note 2
	0	Stops operation
	1	Enables operation

Notes 1. COI is set to 0 when CSIE0 = 0.

★ **2.** In the SBI mode, serial interface channel 0 operation stops (CSIE ← 0) when WUP is cleared to. If WUP is not cleared to 0, P25 is fixed high level and it cannot be used as a normal port.

(3) Serial bus interface control register (SBIC)

This register sets the operation of the serial bus interface and indicates the status.

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 13-4. Format of Serial Bus Interface Control Register (1/2)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W Note

R/W

RELT	Used to output bus release signal. SO0 latch is set to 1 when RELT = 1. After setting SO0 latch, RELT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

R/W

CMDT	Used to output command signal. SO0 latch is cleared to 0 when CMDT = 1. After clearing SO0 latch, CMDT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

R

RELD	Bus release detection	
	Clear condition (RELD = 0)	Set condition (RELD = 1)
	• When transfer start instruction is executed • When values of SIO0 and SVA do not coincide when address is received • When $\overline{\text{CSIE0}}$ = 0 • When $\overline{\text{RESET}}$ is input	• When bus release signal (REL) is detected

R

CMDD	Command detection	
	Clearing conditions (CMDD = 0)	Setting condition (CMDD = 1)
	• When transfer start instruction is executed • When bus release signal (REL) is detected • When $\overline{\text{CSIE0}}$ = 0 • When $\overline{\text{RESET}}$ is input	• When command signal (CMD) is detected

Note Bits 2, 3, and 6 (RELD, CMDD, and ACKD) are read-only bits.

Remark CSIE0: bit 7 of serial operation mode register 0 (CSIM0)

Figure 13-4. Format of Serial Bus Interface Control Register (2/2)

R/W		
R/W	ACKT	Outputs acknowledge signal in synchronization with falling edge of clock of $\overline{SCK0}$ immediately after instruction that sets this bit to 1 has been executed. After acknowledge signal has been output, this bit is automatically cleared to 0. This bit is also cleared to 0 when transfer of serial interface is started or when CSIE0 = 0.

R/W	ACKE	Acknowledge signal output control	
R/W	0	Disables automatic output of acknowledge signal (output by ACKT is enabled)	
R/W	1	Before completion of transfer	Outputs acknowledge signal in synchronization with falling edge of 9th clock of $\overline{SCK0}$ (automatically outputs when ACKE = 1).
R/W	1	After completion of transfer	Outputs acknowledge signal in synchronization with falling edge of clock of $\overline{SCK0}$ immediately after instruction that sets this bit to 1 has been executed (automatically output when ACKE = 1). However, this bit is not automatically cleared to 0 after acknowledge signal has been output.

R	ACKD	Acknowledge detection	
	Clearing conditions (ACKD = 0)		Setting condition (ACKD = 1)
	• At falling edge of clock of $\overline{SCK0}$ immediately after busy mode is released after transfer start instruction has been executed • When CSIE0 = 0 • When $\overline{RESET}$ is input		• When acknowledge signal ($\overline{ACK}$) is detected at rising edge of clock of $\overline{SCK0}$ after completion of transfer

R/W	Note BSYE	Synchronous busy signal output control
R/W	0	Disables output of busy signal in synchronization with falling edge of clock of $\overline{SCK0}$ immediately after instruction that clears this bit to 0 has been executed.
R/W	1	Outputs busy signal from falling edge of clock of $\overline{SCK0}$ following acknowledge signal.

★ **Note** The busy mode can be released when transfer by the serial interface has been started. However, the BSYE flag is not cleared to 0.

Remarks **1.** Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when they are read after data has been set.

2. CSIE0: bit 7 of serial operation mode register 0 (CSIM0)

(4) Interrupt timing specification register (SINT)

This register sets the bus release interrupt and address mask function, and indicates the status of the level of the $\overline{\text{SCK0}}$/P27 pin.

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 13-5. Format of Interrupt Timing Specification Register

Symbol	7	<6>	<5>	<4>	3	2	1	0	Address	After reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W Note 1

R/W

SVAM	Bits of SVA used as slave address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	Selects interrupt source of INTCSI0 Note 2
0	Sets CSIIF0 at end of transfer by serial interface channel 0
1	Sets CSIIF0 at end of transfer by serial interface channel 0 or on detection of bus release signal

R

CLD	Level of $\overline{\text{SCK0}}$/P27 pin Note 3
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is a read-only bit.

2. Set SIC to 0 when using the wake-up function in SBI mode.

3. CLD is 0 when CSIE0 = 0.

Caution Be sure to set bits 0 through 3 to 0.

Remark SVA : slave address register

CSIIF0 : interrupt request flag for INTCSI0

CSIE0 : bit 7 of serial operation mode register 0 (CSIM0)

13.4 Operation of Serial Interface Channel 0

Serial interface channel 0 operates in the following four operation modes:

- Operation stop mode
- 3-wire serial I/O mode
- SBI mode
- 2-wire serial I/O mode

13.4.1 Operation stop mode

Serial transfer is not executed in this mode. Consequently, the power dissipation can be reduced.

The serial I/O shift register 0 (SIO0) can be used as an ordinary 8-bit register because it does not perform the shift operation.

In the operation stop mode, the P25/SIO/SB0, P26/SO0/SB1, and P27/$\overline{SCK0}$ pins can be used as ordinary I/O port pins.

(1) Register setting

The operation stop mode is set by using the serial operation mode register 0 (CSIM0).

CSIM0 is set by using a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{RESET}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

R/W	CSIE0	Serial interface channel 0 operation control
	0	Stops operation
	1	Enables operation

13.4.2 Operation in 3-wire serial I/O mode

This mode is useful for connecting peripheral I/Os and display controllers that have the conventional clocked serial interface of the 75X/XL Series, 78K Series, and 17K Series.

In this mode, communication is established by using three signal lines: serial clock ($\overline{SCK0}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set by using the serial operation mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operation mode register 0 (CSIM0)

CSIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{RESET}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W[Note 1]

R/W

CSIM01	CSIM00	Selects clock of serial interface channel 0
0	×	Clock externally input to $\overline{\text{SCK0}}$ pin
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation mode	First bit	SI0/SB0/P25 pin function	SO0/SB1/P26 pin function	$\overline{\text{SCK0}}$/P27 pin function
0	×	0	1	×	0	0	0	1	3-wire serial I/O mode	MSB	SI0[Note 2] (input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS I/O)
		1								LSB			
1	0	SBI mode (Refer to **13.4.3 Operation in SBI mode**.)											
1	1	2-wire serial I/O mode (Refer to **13.4.4 Operation in 2-wire serial I/O mode**.)											

R/W

WUP	Controls wake-up function[Note 3]
0	Generates interrupt request signal in all modes each time serial transfer is executed
1	Generates interrupt request signal when address received after bus has been released in SBI mode (when CMDD = RELD = 1) coincides with data of slave address register (SVA)

R/W

CSIE0	Controls operation of serial interface channel 0
0	Stops operation
1	Enables operation

Notes **1.** Bit 6 (COI) is a read-only bit.

2. This pin can be used as P25 (CMOS input) when used only for transfer.

3. Be sure to set WUP to 0 in the 3-wire serial I/O mode.

Remark × : don't care

PM×× : port mode register

P×× : output latch of a port

(b) Serial bus interface control register (SBIC)

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W

RELT	SO0 latch is set to 1 when RELT = 1. After setting SO0 latch, RELT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

R/W

CMDT	SO0 latch is cleared to 0 when CMDT = 1. After clearing SO0 latch, CMDT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

CSIE0: bit 7 of serial operation mode register 0 (CSIM0)

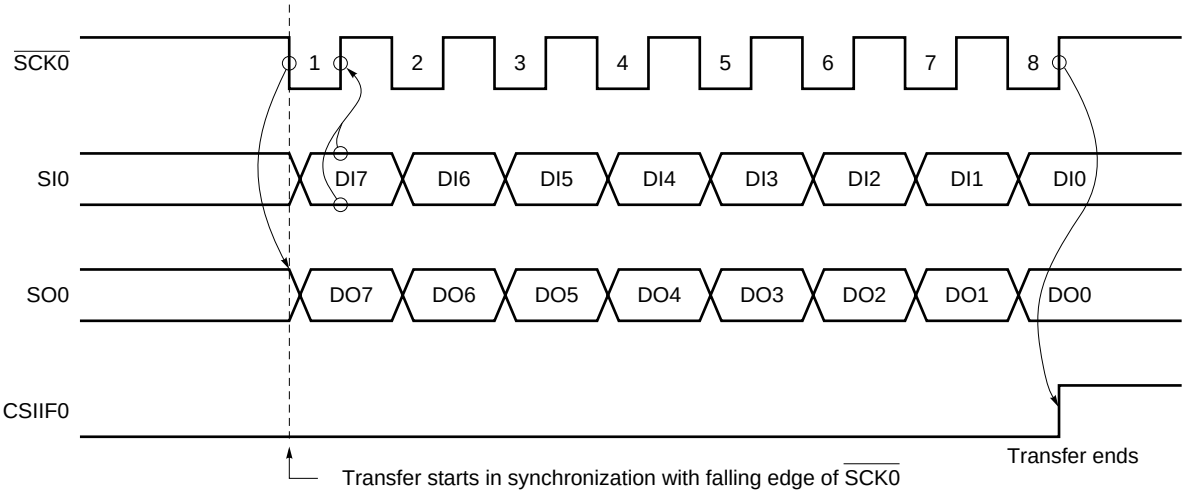
(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted/received in 8-bit units. Data is transmitted/received on a 1-bit-by-1-bit basis in synchronization with the serial clock.

The shift operation of the serial I/O shift register 0 (SIO0) is performed in synchronization with the falling edge of the serial clock ($\overline{SCK0}$). The transmitted data is retained by the SO0 latch and output from the SO0 pin. The receive data input to the SI0 pin is latched to SIO0 at the rising edge of $\overline{SCK0}$.

When the 8-bit data has been completely transferred, the operation of SIO0 is automatically stopped, and an interrupt request flag (CSIIF0) is set.

Figure 13-6. Timing of 3-wire Serial I/O Mode



The SO0 pin serves as a CMOS output pin and outputs the status of the SO latch. The output status of the SO0 pin can be manipulated by setting the bits 0 (RELT) and 1 (CMDT) of the serial bus interface control register (SBIC).

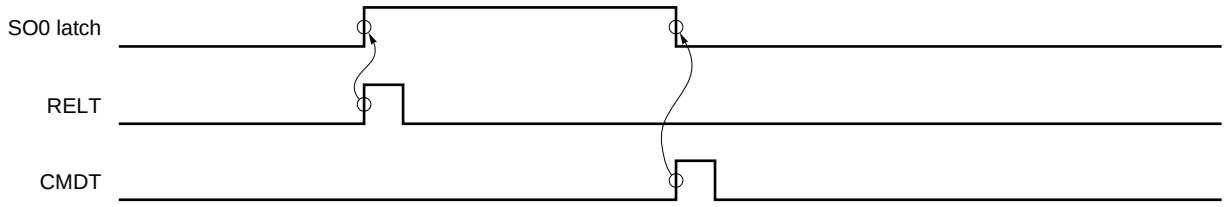
However, do not perform this manipulation during serial transfer.

The output level of the $\overline{SCK0}$ pin is controlled by manipulating the P27 output latch in the output mode (mode of the internal system clock) (refer to **13.4.5 Manipulating $\overline{SCK0}$/P27 pin output**).

(3) Signals

Figure 13-7 shows the operations of RELT and CMDT.

Figure 13-7. Operations of RELT and CMDT



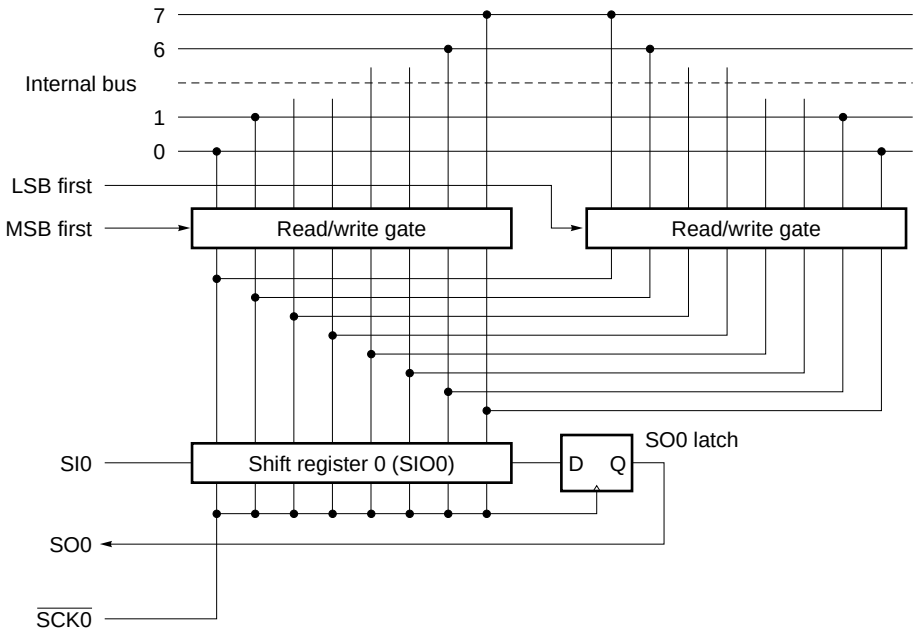
(4) MSB/LSB first selection

In the 3-wire serial I/O mode, whether data is transferred with the MSB or LSB first can be selected.

Figure 13-8 shows the configuration of the serial I/O shift register 0 (SIO0) and internal bus. As shown in the figure, data can be read/written with the MSB/LSB inverted.

Whether the MSB or LSB is transferred first can be specified by using the bit 2 (CSIM02) of the serial operation mode register 0 (CSIM0).

Figure 13-8. Transfer Bit Sequence Select Circuit



The first bit is selected by changing the bit order in which data is written to SIO0. The shift sequence of SIO0 is always the same.

Therefore, specifying whether the MSB or LSB is the first bit should be performed before writing data to the shift register.

(5) Transfer start

Serial transfer is started by setting the transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied:

- Operation control bit of serial interface channel 0 (CSIE0) = 1
- When internal serial clock is stopped or $\overline{SCK0}$ is high after 8-bit serial transfer

Caution Even if CSIE0 is set to "1" after data has been written to SIO0, transfer is not started.

Serial transfer is automatically stopped at the end of 8-bit transfer, and an interrupt request flag (CSIIF0) is set.

13.4.3 Operation in SBI mode

SBI (serial bus interface) is a high-speed serial interface mode conforming to NEC's serial bus format.

SBI is a clocked serial I/O method in a format with a function for bus configuration added, so that a single master can communicate with two or more devices with a high-speed serial bus consisting of two signal lines. Therefore, the number of ports and wirings on a printed wiring board can be reduced when the serial bus consists of plural microcontrollers and peripheral ICs.

The master can output "addresses" that select the target device(s) for serial communication, "commands" that directs the target device(s), and actual "data" to the slaves via serial data bus. A slave can identify the received data as an "address", "command", or "data" by hardware. This function can simplify the portion in application program that controls channel 0.

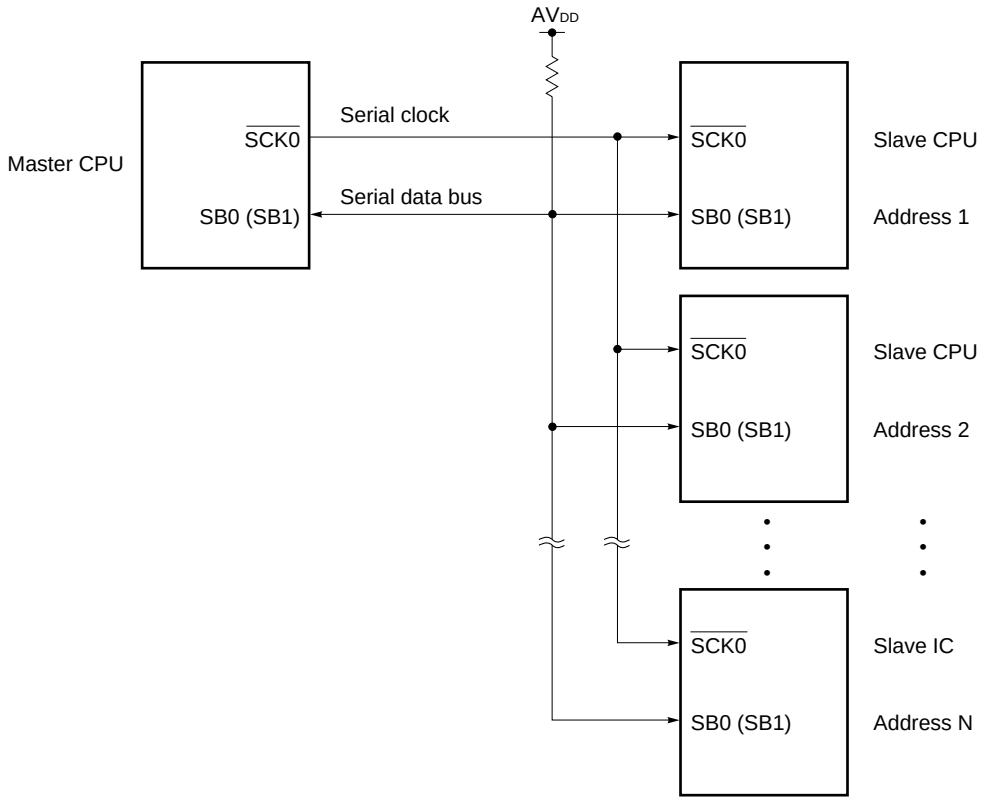
The SBI function is provided to some devices such as the 75X/XL Series and 78K Series.

Figure 13-9 shows an example of configuration of the serial bus when a CPU or peripheral IC with a serial interface conforming to SBI is used.

Because the serial data bus pin SB0 (SB1) in SBI is an open-drain output pin, the serial data bus line is wired-ORed. A pull-up resistor is necessary for the serial data bus line.

When using the SBI mode, refer to **(11) Notes on SBI mode (d)**.

Figure 13-9. Example of Serial Bus Configuration by SBI



Caution When the master is exchanged with a slave, a pull-up resistor is necessary for the serial clock line ($\overline{\text{SCK0}}$) because switching over between the input and output mode of the serial clock line ($\overline{\text{SCK0}}$) is performed asynchronously between the master and the slave.

(1) Function of SBI

With the existing serial I/O method, many ports and wirings are necessary to identify a chip select signal, command, data, and busy status when a serial bus consists of plural devices because only a data transfer function is provided. To perform this control by software, the work load of the software increases.

SBI can constitute a serial bus by using two signal lines: serial clock $\overline{SCK0}$ and serial data bus SB0 (SB1). Therefore, the number of ports of the microcontroller and the wiring length on the printed wiring board can be reduced effectively.

SBI has the following functions:

(a) Address/command/data identification function

Serial data is identified as an address, a command, or data.

(b) Chip select status by address

The master selects a slave chip by transferring an address to the slave.

(c) Wake-up function

The slave can easily judge that it has received an address (chip select judgement), by using the wake-up function (which can be set or released by software).

When the wake-up function is set, an interrupt request signal (INTCSI0) is generated when the slave receives an address that matches the address of the slave.

Therefore, even when the master communicates with two or more devices, the CPU of the slaves other than that selected can operate regardless of serial communication.

(d) Acknowledge signal ($\overline{ACK}$) control function

An acknowledge signal is controlled to check reception of serial data.

(e) Busy signal ($\overline{BUSY}$) control function

A busy signal that indicates the busy status of the slave is controlled.

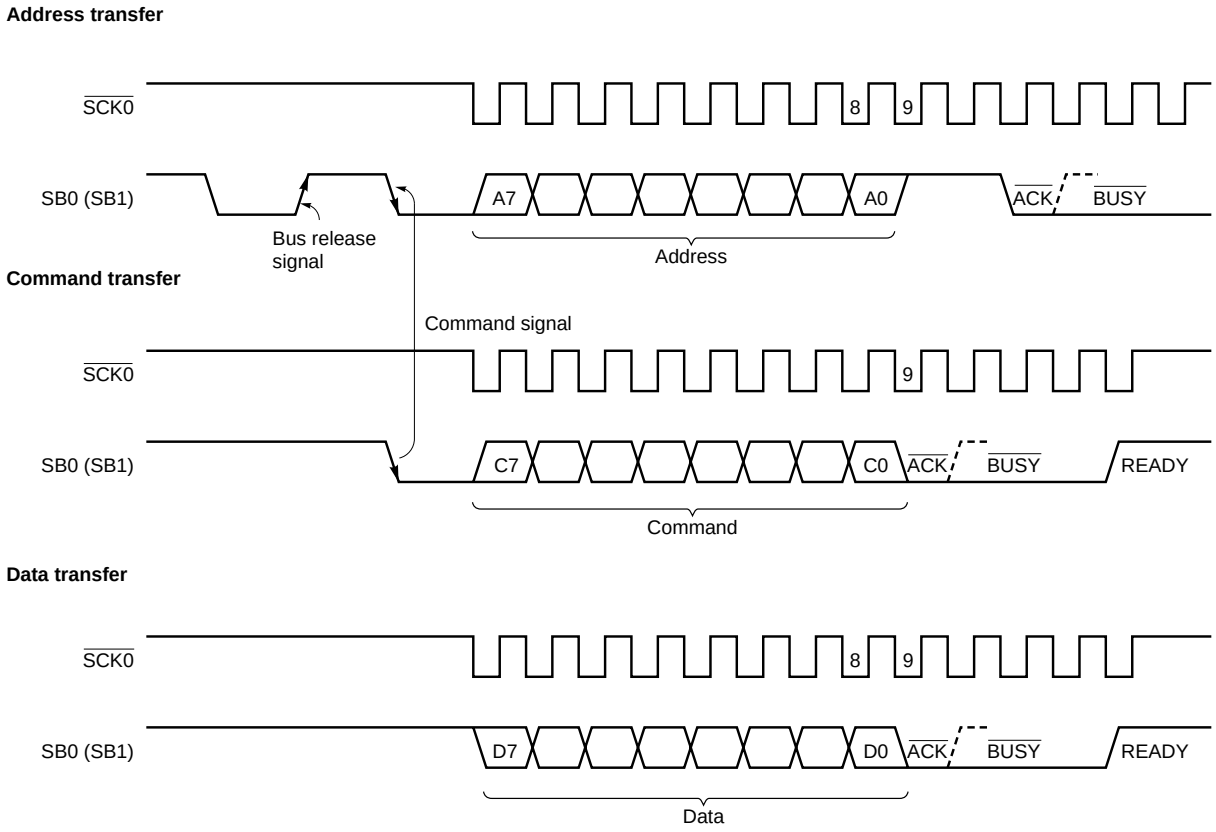
(2) Definition of SBI

This section describes the serial data format of SBI, and the meaning of used data.

The serial data transferred by SBI are classified into “addresses”, “commands”, and “data”.

Figure 13-10 shows the transfer timing of the address, command, and data.

Figure 13-10. SBI Transfer Timing



Remark The broken line indicates the READY status.

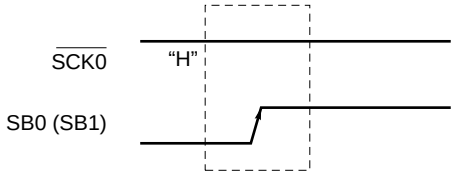
The bus release signal and command signal are output by the master. $\overline{\text{BUSY}}$ is output by the slave. $\overline{\text{ACK}}$ can be output by both the master and slave (usually, this signal is output by the 8-bit data reception side). The master continues outputting the serial clock from the start of 8-bit data transfer, until $\overline{\text{BUSY}}$ is released.

(a) Bus release signal (REL)

The bus release signal is the positive transition signal of the SB0 (SB1) line, i.e., transition from the low to high level, when the $\overline{SCK0}$ line is high (when the serial clock is not output).

This signal is output by the master.

Figure 13-11. Bus Release Signal



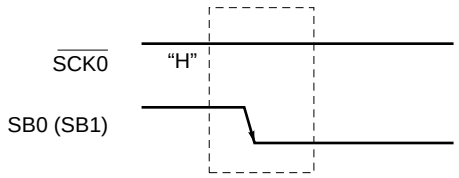
The bus release signal indicates that the master is to transmit an address to the slave. The slave is provided with hardware that detects the bus release signal.

★ **Caution A transition of the SB0 (SB1) line from low to high is recognized as a bus release signal while the $\overline{SCK0}$ line is high. Therefore shifting of the change timing of the bus due to the influence of the board capacitance, etc., may cause incorrect identification as a bus release signal, regardless of whether data is being transmitted. For this reason, special care must be taken regarding wiring.**

(b) Command signal (CMD)

The command signal is the negative transition signal of the SB0 (SB1) line, i.e., transition from the high to low level, when the $\overline{SCK0}$ line is high (when the serial clock is not output). This signal is output by the master.

Figure 13-12. Command Signal



A command signal indicates that a command is going to be sent from the master device to a slave device. (However, the command signal following a bus release signal indicates that an address is going to be sent next.)

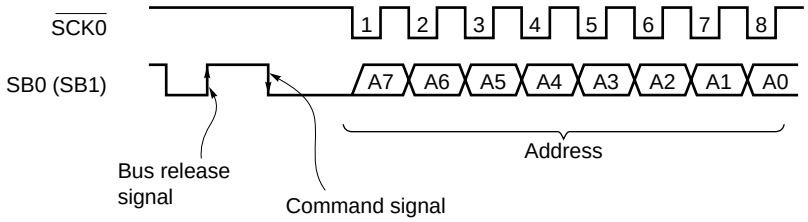
The slave is provided with hardware that detects the command signal.

★ **Caution A transition of the SB0 (SB1) line from high to low is recognized as a bus release signal while the $\overline{SCK0}$ line is high. Therefore shifting of the change timing of the bus due to the influence of the board capacitance, etc., may cause incorrect identification as a command signal, regardless of whether data is being transmitted. For this reason, special care must be taken regarding wiring.**

(c) Address

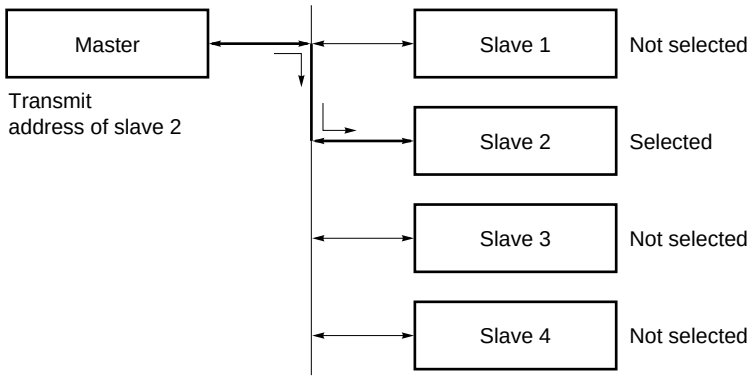
An address is an 8-bit data which the master outputs to the slaves connected to the bus lines in order to select a specific slave.

Figure 13-13. Address



The 8-bit data that follows the bus release signal and command signal is defined as an address. The slave detects this condition by hardware and checks by hardware whether the 8-bit data matches the identification number of the slave itself (slave address). If the 8-bit data matches the slave address of a slave, that slave is selected. After that, the slave communicates with the master, until it is later directed to be disconnected from the master.

Figure 13-14. Selecting Slave by Address



(d) Command and data

The master transmits commands or transmits/receives data to the slave it has selected by transmitting an address.

Figure 13-15. Command

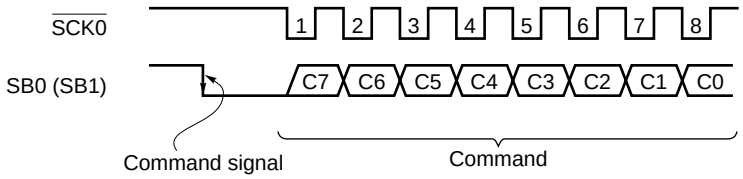
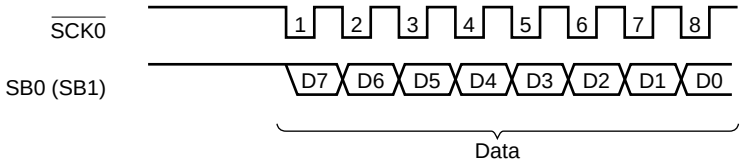


Figure 13-16. Data



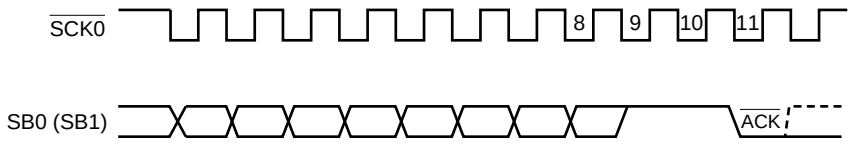
The 8-bit data following the command signal is defined as a command. The 8-bit data that does not follow the command signal is defined as data. The method of using the command and data can be arbitrarily determined by the communication specifications.

(e) Acknowledge signal ($\overline{ACK}$)

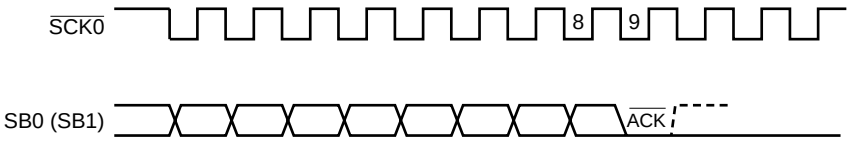
The acknowledge signal is used for confirmation of reception of serial data between the transmission and reception sides.

Figure 13-17. Acknowledge Signal

[When output in synchronization with $\overline{SCK0}$ of 11th clock]



[When output in synchronization with $\overline{SCK0}$ of 9th clock]



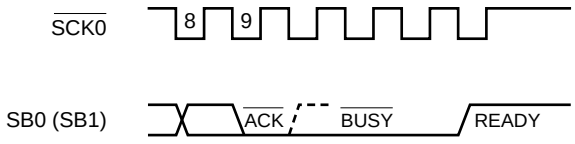
Remark The broken line indicates the READY status.

The acknowledge signal is a one-shot pulse synchronized with the falling edge of $\overline{SCK0}$ after 8-bit data has been transferred. Its position is arbitrary, and may be synchronized with $\overline{SCK0}$ of clock n.
After the transmission side has transferred 8-bit data, it checks whether the reception side has returned an acknowledge signal. If no acknowledge signal is returned within a specific time after data transmission, it is judged that the data was not received correctly.

(f) Busy signal ($\overline{\text{BUSY}}$), ready signal (READY)

The busy signal informs the master that the slave is getting ready for transmitting/receiving data.
The ready signal informs the master that the slave is ready to transmit/receive data.

Figure 13-18. Busy Signal and Ready Signal



Remark The broken line indicates the READY status.

With SBI, the slave informs the master of the busy status by making the SB0 (SB1) line low.
The busy signal is output following the acknowledge signal output by the master or the slave. The busy signal is set or released in synchronization with the falling edge of $\overline{\text{SCK0}}$. The master automatically ends outputting serial clock $\overline{\text{SCK0}}$ when the busy signal is released.
The master can start the next transfer when the busy signal has been released and the ready signal is issued.

Caution In the SBI mode, the $\overline{\text{BUSY}}$ signal keeps being output after a release of $\overline{\text{BUSY}}$ is directed until the next falling edge of the serial clock ($\overline{\text{SCK0}}$). If the WUP bit is set to 1 inadvertently during that period, $\overline{\text{BUSY}}$ cannot be released. Therefore, be sure to release $\overline{\text{BUSY}}$ and set SB0 (SB1) pin to high before setting WUP bit to 1.

(3) Register setting

The SBI mode is set by the serial operation mode register 0 (CSIM0), serial bus interface control register (SBIC), and interrupt timing specification register (SINT).

(a) Serial operation mode register 0 (CSIM0)

CSIM0 is set by a 1-bit or 8-bit memory manipulation instruction.
This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W Note 1

R/W

CSIM01	CSIM00	Selects clock of serial interface channel 0
0	×	Clock externally input to $\overline{SCK0}$ pin
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation mode	First bit	SI0/SB0/P25 pin function	SO0/SB1/P26 pin function	$\overline{SCK0}$/P27 pin function
0	×	3-wire serial I/O mode (Refer to **13.4.2 Operation in 3-wire serial I/O mode**.)											
1	0	0	Note 2 ×	Note 2 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open drain I/O)	$\overline{SCK0}$ (CMOS I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open drain I/O)	P26 (CMOS I/O)	
1	1	2-wire serial I/O mode (Refer to **13.4.4 Operation in 2-wire serial I/O mode**.)											

R/W

WUP	Controls wake-up function Note 3
0	Generates interrupt request signal in all modes each time serial transfer is executed
1	Generates interrupt request signal when address received after bus has been released in SBI mode (when CMDD = RELD = 1) coincides with data of slave address register (SVA)

R

COI	Slave address comparison result flag Note 4
0	Data of slave address register (SVA) does not coincide with data of serial I/O shift register 0 (SIO0)
1	Data of slave address register (SVA) coincides with data of serial I/O shift register 0 (SIO0)

R/W

CSIE0	Controls operation of serial interface channel 0 Note 5
0	Stops operation
1	Enables operation

Notes **1.** Bit 6 (COI) is a read-only bit.

2. This pin can be freely used for port function.

3. Clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0 when using the wake-up function (WUP = 1).

4. When CSIE0 = 0, COI is 0.

★ **5.** In the SBI mode, serial interface channel 0 operaion stops (CSIE ← 0) when WUP is cleared to 0. If WUP is not cleared to 0, P25 is fixed to high level and it cannot be used as a normal port.

Remark × : don't care

PM×× : port mode register

P×× : output latch of a port

(b) Serial bus interface control register (SBIC)

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W[Note]

R/W

RELT	Used to output bus release signal. SO0 latch is set to 1 when RELT = 1. After setting SO0 latch, RELT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

R/W

CMDT	Used to output command signal. SO0 latch is cleared to 0 when CMDT = 1. After clearing SO0 latch, CMDT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

R

RELD	Bus release detection	
Clearing conditions (RELD = 0)		Setting condition (RELD = 1)
• When transfer start instruction is executed • When values of SIO0 and SVA do not coincide when address is received • When CSIE0 = 0 • When $\overline{\text{RESET}}$ is input		• When bus release signal (REL) is detected

R

CMDD	Command detection	
Clearing conditions (CMDD = 0)		Setting condition (CMDD = 1)
• When transter start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ is input		• When command signal (CMD) is detected

R/W

ACKT	Outputs an acknowledge signal in synchronization with the falling edge of the $\overline{\text{SCK0}}$ clock immediately after the instruction that sets this bit to 1 has been executed, and then is automatically cleared to 0. Used as ACKE = 0. This bit is also cleared to 0 when transfer of serial interface is started or when CSIE0 = 0.

Note Bits 2, 3, and 6 (RELD, CMDD, and ACKD) are read-only bits.

Remarks **1.** Bits 0, 1, and 4 (RELT, CMDT, ACKT) are 0 when they are read after data has been set.

2. CSIE0: bit 7 of serial operation mode register 0 (CSIM0)

R/W

ACKE		
Controls output of acknowledge signal		
0	Disables automatic output of acknowledge signal (output by ACKT is enabled)	
1	Before completion of transfer	Outputs acknowledge signal in synchronization with falling edge of 9th clock of $\overline{SCK0}$ (automatically output when ACKE = 1).
	After completion of transfer	Outputs acknowledge signal in synchronization with falling edge of clock of $\overline{SCK0}$ immediately after instruction that sets this bit to 1 has been executed (automatically output when ACKE = 1). However, this bit is not automatically cleared to 0 after acknowledge signal has been output.

R

ACKD	Detects acknowledge
Clearing conditions (ACKD = 0)	Setting condition (ACKD = 1)
• At falling edge of clock of $\overline{SCK0}$ immediately after busy mode is released after transfer start instruction has been executed • When CSIE0 = 0 • When $\overline{RESET}$ signal is input	• When acknowledge signal ($\overline{ACK}$) is detected at rising edge of clock of $\overline{SCK0}$ after completion of transfer

R/W

Note BSYE	Controls synchronous busy signal output
0	Disables output of busy signal in synchronization with falling edge of clock of $\overline{SCK0}$ immediately after instruction that clears this bit to 0 has been executed.
1	Outputs busy signal from falling edge of clock of $\overline{SCK0}$ following acknowledge signal.

★ **Note** The busy mode can be released when transfer by the serial interface has been started. However, the BSYE flag is not cleared to 0.

Remark CSIE0: bit 7 of serial operation mode register 0 (CSIM0)

(c) Interrupt timing specification register (SINT)

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	7	<6>	<5>	<4>	3	2	1	0	Address	After reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W[Note 1]

R/W

SVAM	Bits of SVA used as slave address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	Selects interrupt source of INTCSI0[Note 2]
0	Sets CSIIF0 at end of transfer by serial interface channel 0
1	Sets CSIIF0 at end of transfer by serial interface channel 0 or on detection of bus release

R

CLD	Level of $\overline{\text{SCK0}}$/P27 pins[Note 3]
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is a read-only bit.

2. Set SIC to 0 when using the wake-up function in the SBI mode.

3. CLD is 0 when CSIE0 = 0.

Caution Be sure to set bits 0 through 3 to 0.

Remark SVA : slave address register

CSIIF0: interrupt request flag for INTCSI0

CSIE0 : bit 7 of serial operation mode register 0 (CSIM0)

(4) Signals

Figures 13-19 through 13-24 show the signals of serial bus interface control register (SBIC) and the operations of the flags of SBIC. Table 13-4 lists the signals of SBI.

Figure 13-19. Operations of RELT, CMDT, RELD, and CMDD (Master)

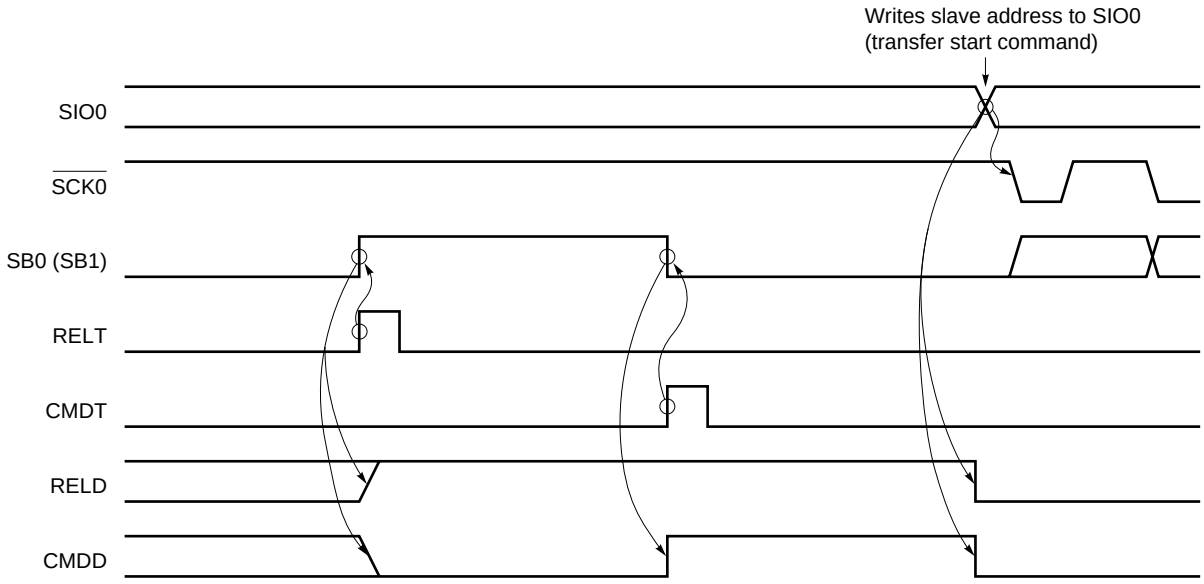


Figure 13-20. Operations of RELD and CMDD (Slave)

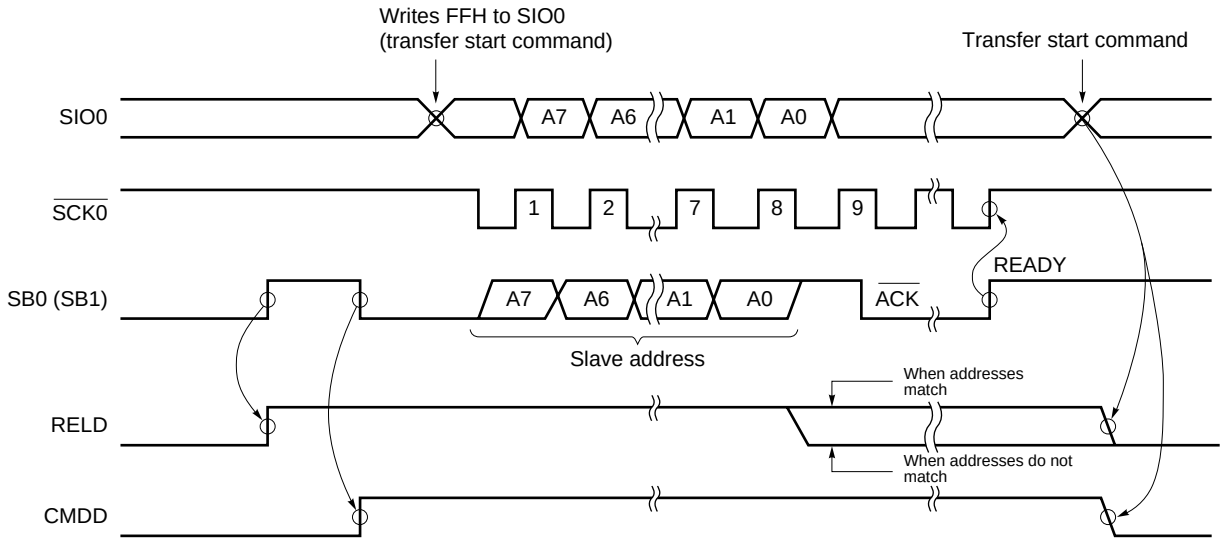
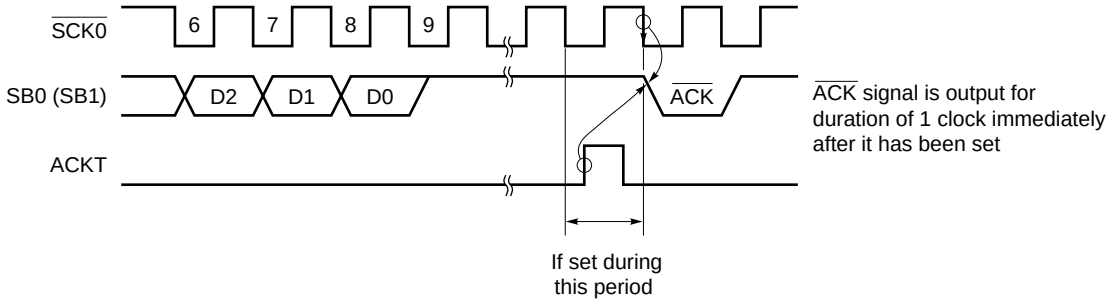


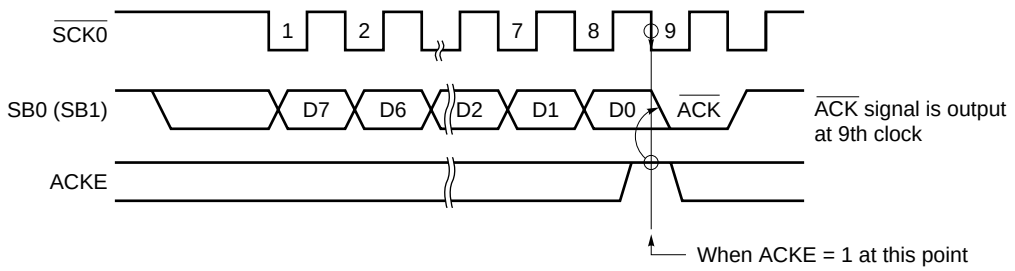
Figure 13-21. Operation of ACKT



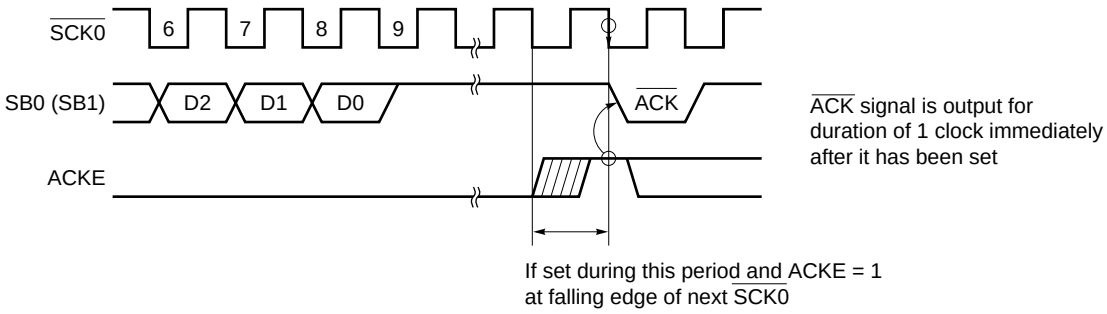
Caution Do not set ACKT before end of transfer.

Figure 13-22. Operation of ACKE

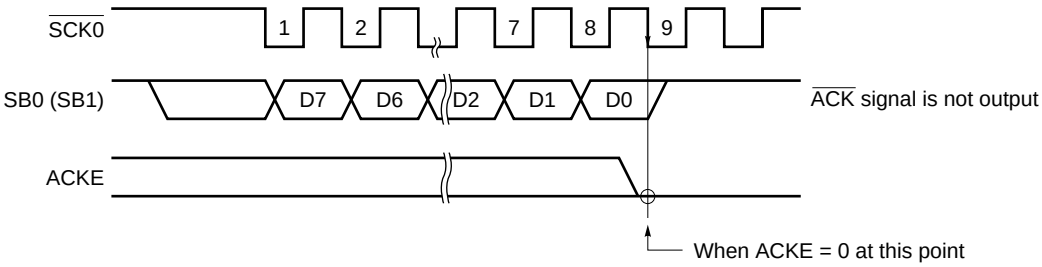
(a) When ACKE = 1 at end of transfer



(b) When set after transfer



(c) When ACKE = 0 at end of transfer



(d) If period of ACKE = 1 is short

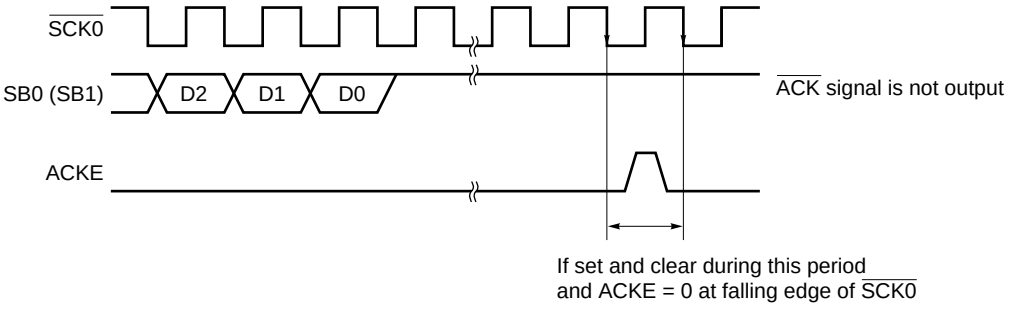
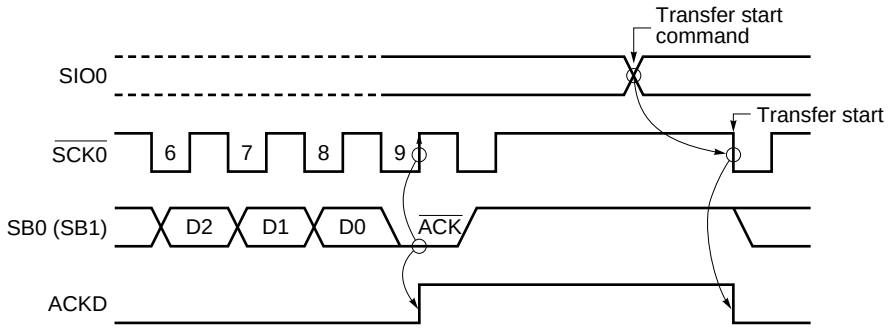
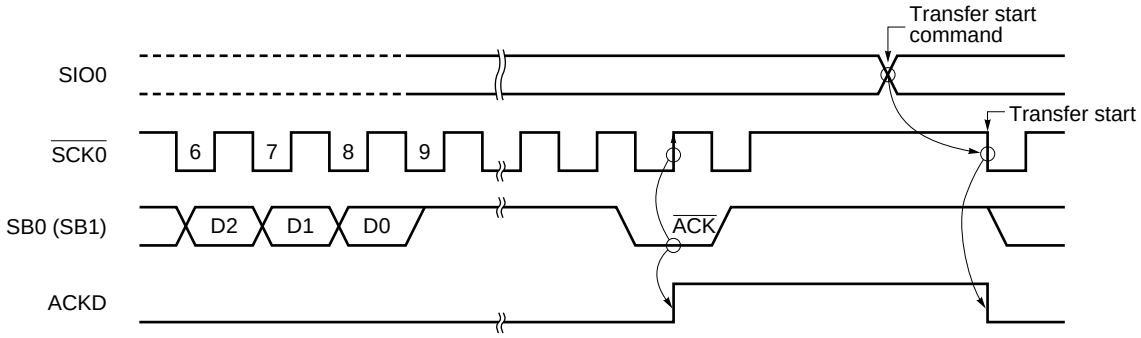


Figure 13-23. Operation of ACKD

(a) If $\overline{ACK}$ signal is output during 9th clock period of $\overline{SCK0}$



(b) If $\overline{ACK}$ signal is output after 9th clock of $\overline{SCK0}$



(c) Clearing timing when transfer start command is issued during BUSY

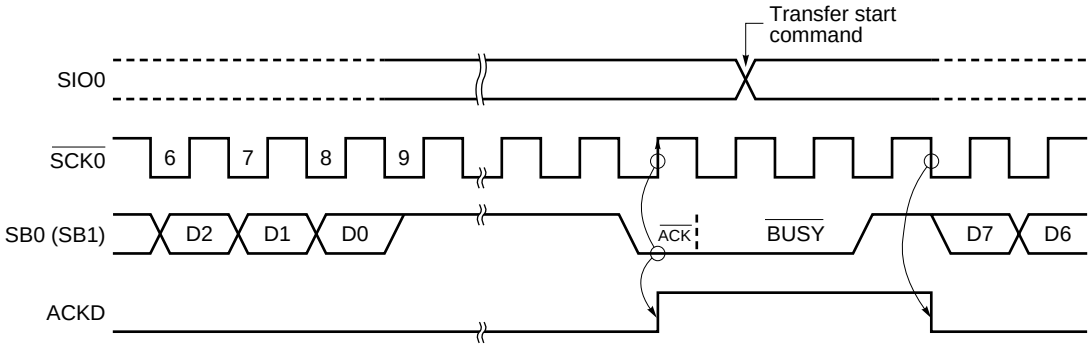


Figure 13-24. Operation of BSYE

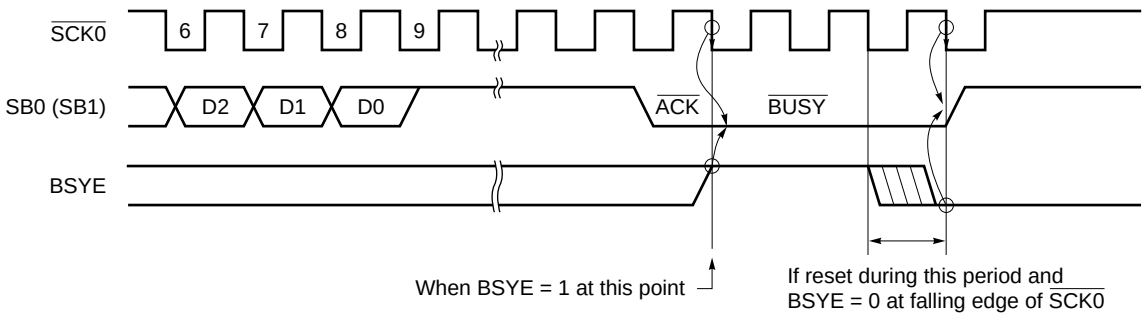
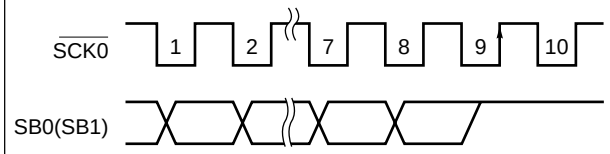
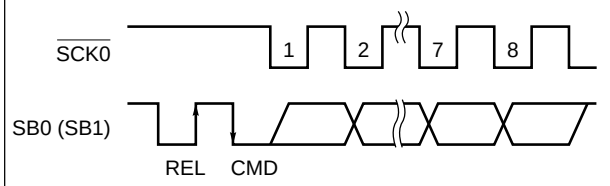
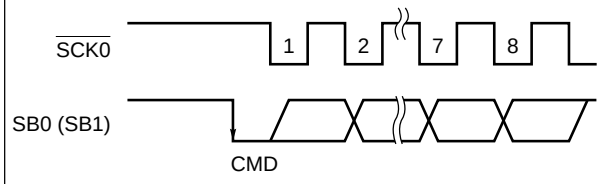
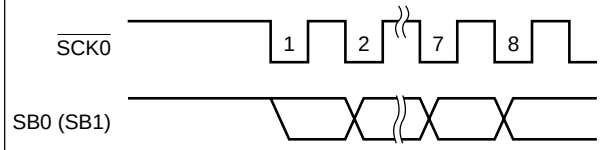


Table 13-4. Signals in SBI Mode (1/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Influence on Flag	Meaning of Signal
Bus release signal (REL)	Master	At rising edge of SB0 (SB1) when $\overline{SCK0}$ = 1	$\overline{SCK0}$ "H" SB0 (SB1)	• Setting of RELT	• RELD is set. • CMDD is cleared.	Subsequently outputs CMD signal and indicates that transmit data is address.
Command signal (CMD)	Master	At falling edge of SB0 (SB1) when $\overline{SCK0}$ = 1	$\overline{SCK0}$ "H" SB0 (SB1)	• Setting of CMDT	• CMDD is set.	i) Data transmitted after REL signal is output is address. ii) Data transferred without REL signal output is command.
Acknowledge signal ($\overline{ACK}$)	Master/ slave	Low-level signal output to SB0 (SB1) for duration of 1 clock of $\overline{SCK0}$ after serial reception has been completed.	[Synchronous busy output]	<1> ACKE = 1 <2> Setting of ACKT	• ACKD is set.	Reception completed.
Busy signal ($\overline{BUSY}$)	Slave	[Synchronous busy signal] Low-level signal output to SB0 (SB1) following acknowledge signal	$\overline{SCK0}$ 9 $\overline{ACK}$ $\overline{BUSY}$ SB0 (SB1) D0 READY	• BSYE = 1	—	Serial reception disabled because processing is in progress.
Ready signal (READY)	Slave	High-level signal output to SB0 (SB1) before start and after completion of serial transfer	$\overline{ACK}$ $\overline{BUSY}$ SB0 (SB1) D0 READY	<1> BSYE = 0 <2> Instruction execution that writes data to SIO0 (transfer start command)	—	Serial reception enabled status

Table 13-4. Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Influence on Flag	Meaning of Signal
Serial clock ($\overline{SCK0}$)	Master	Synchronous clock for output of address/command/data, $\overline{ACK}$ signal, and synchronous $\overline{BUSY}$ signal. Address/command/data is transferred when first eight of this signal are output.		Execution of instruction that writes data to SIO0 when CSIE0 = 1 (serial transfer start command)Note 2	CSIIF0 is set (rising edge of 9th clock of $\overline{SCK0}$)Note 1.	Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data transferred in synchronization with $\overline{SCK0}$ after REL and CMD signals have been output.				Address value of slave device on serial bus
Command (C7 to C0)	Master	8-bit data transferred in synchronization with $\overline{SCK0}$ after only CMD signal is output (REL signal is not output).				Command or message to slave device
Data (D7 to D0)	Master/ slave	8-bit data transferred in synchronization with $\overline{SCK0}$ when both REL and CMD signals are not output.				Numeric value processed by slave or master device

Notes 1. When WUP = 0, CSIIF0 is always set at the rising edge of the 9th clock of $\overline{SCK0}$.
When WUP = 1, an address is received. Only when this address matches the value of the slave address register (SVA), CSIIF0 is set (if the address does not match, RELD is cleared).

2. In the $\overline{BUSY}$ status, transfer is not started until the READY status is set.

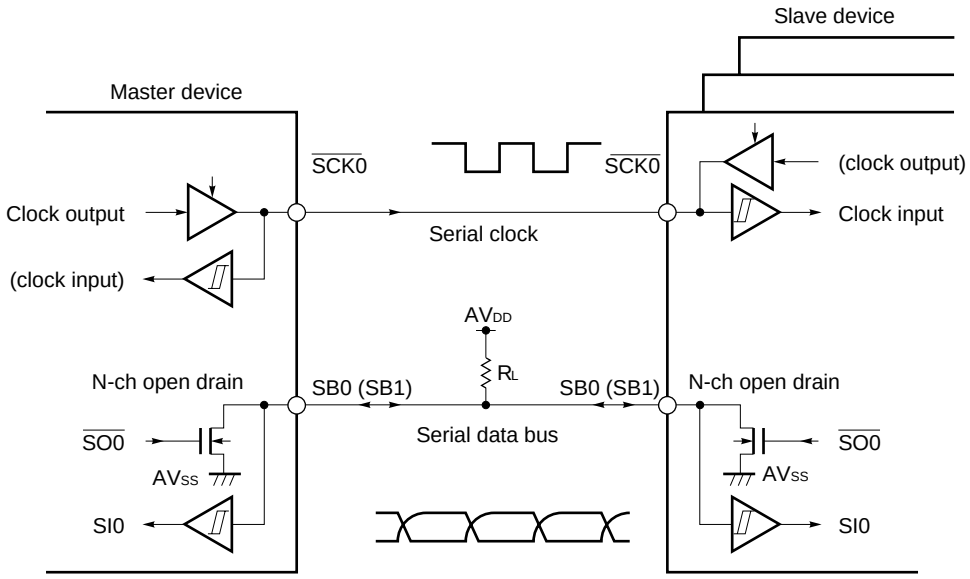
(5) Pin configuration

The configuration of the serial clock pin $\overline{SCK0}$ and serial data bus pin SB0 (SB1) is as follows:

(a) $\overline{SCK0}$ ..... Pin that inputs/outputs serial clock
  <1> Master .... CMOS, push-pull output
  <2> Slave ..... Schmitt input
(b) SB0 (SB1) ..... Serial data input/output dual pin.
  N-ch open drain output and Schmitt input for both master and slave

Because the serial data bus line is of N-ch open drain output, an external pull-up resistor is necessary.

Figure 13-25. Pin Configuration



Caution Because it is necessary to set the N-ch open drain gate to high-impedance state when data is received, write FFH to serial I/O shift register 0 (SIO0) in advance. Setting it to high impedance is allowed at any time even during transfer. However, when the wake-up function specification bit (WUP) = 1, the N-ch open-drain is always set to high-impedance state; therefore, it is not necessary to write FFH to SIO0 before reception.

(6) Method of detecting address matching

In the SBIC mode, a specific slave device can be selected when the master transmits a slave address. Whether the slave address output by the master coincides with the value of the slave address register (SVA) of a slave is automatically detected by hardware. When the wake-up function specification (WUP: bit 5 of serial operation mode register (CSIM0)) is 1 and only if the slave address transmitted by the master coincides with the address set to the SVA, CSIIF0 is set.

If bit 5 (SIC) of the interrupt timing specification register is set to 1, the wake-up function does not operate even if WUP is set to 1 (an interrupt request signal is generated on detection of bus release). Clear SIC to 0 to use the wake-up function.

Cautions 1. Whether a slave is selected or not is detected by matching of a slave address that has been received after the bus release signal has been issued (RELD = 1).
To detect matching of addresses, an address match interrupt (INTCSI0) that is generated when WUP = 1 is usually used. Therefore, check whether a slave device is selected or not by reception of a slave address when WUP = 1.
2. To detect whether a slave is selected or not when WUP = 0 without using the interrupt, do so by transmitting/receiving a command set by program in advance, instead of using the address matching detection method.

(7) Error detection

In the SBI mode, the status of the serial bus SB0 (SB1) is also loaded to the serial I/O shift register 0 (SIO0) of the device that is transmitting data; therefore, a transmit error can be detected by the following method:

(a) By comparing data of SIO0 before start and after completion of transmission

In this case, it is judged that an error has occurred if two data are different.

(b) By using slave address register (SVA)

The transmission data is set to SIO0 and SVA and transmission is executed. After completion of transmission, the COI bit (match signal from address comparator) of the serial operation mode register 0 (CSIM0) is tested. If this bit is “1”, it is judged that transmission has been completed normally. If it is “0”, it is judged that an error has occurred.

(8) Communication operation

In the SBI mode, the master usually selects one slave device for communication from two or more devices by outputting an “address” to the serial bus.

After the target device for communication has been determined, commands and data are transmitted/received between the master device and slave device, realizing serial communication.

Figures 13-26 through 13-29 show the timing chart of data communication.

The serial I/O shift register 0 (SIO0) performs shift operation in synchronization with the falling edge of the serial clock ($\overline{\text{SCK0}}$). The transmit data is latched to the SO0 latch, and is output from the SB0/P25 or SB1/P26 pin, starting from the MSB. The receive data input to the SB0 (or SB1) pin at the rising edge of $\overline{\text{SCK0}}$ is latched to the SIO0.

Figure 13-26. Address Transmission Operation from Master Device to Slave Device (WUP = 1)

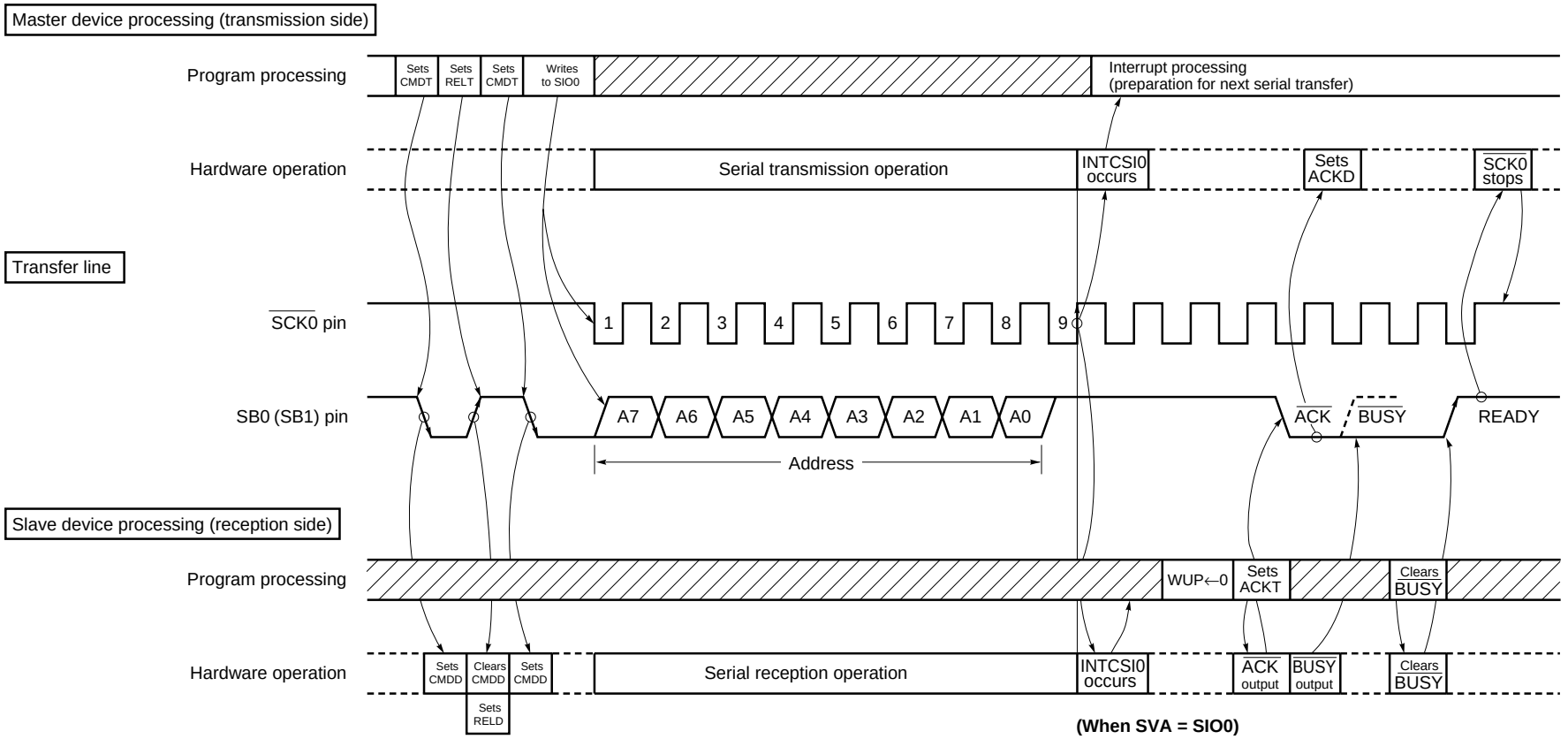


Figure 13-27. Command Transmission Operation from Master Device to Slave Device

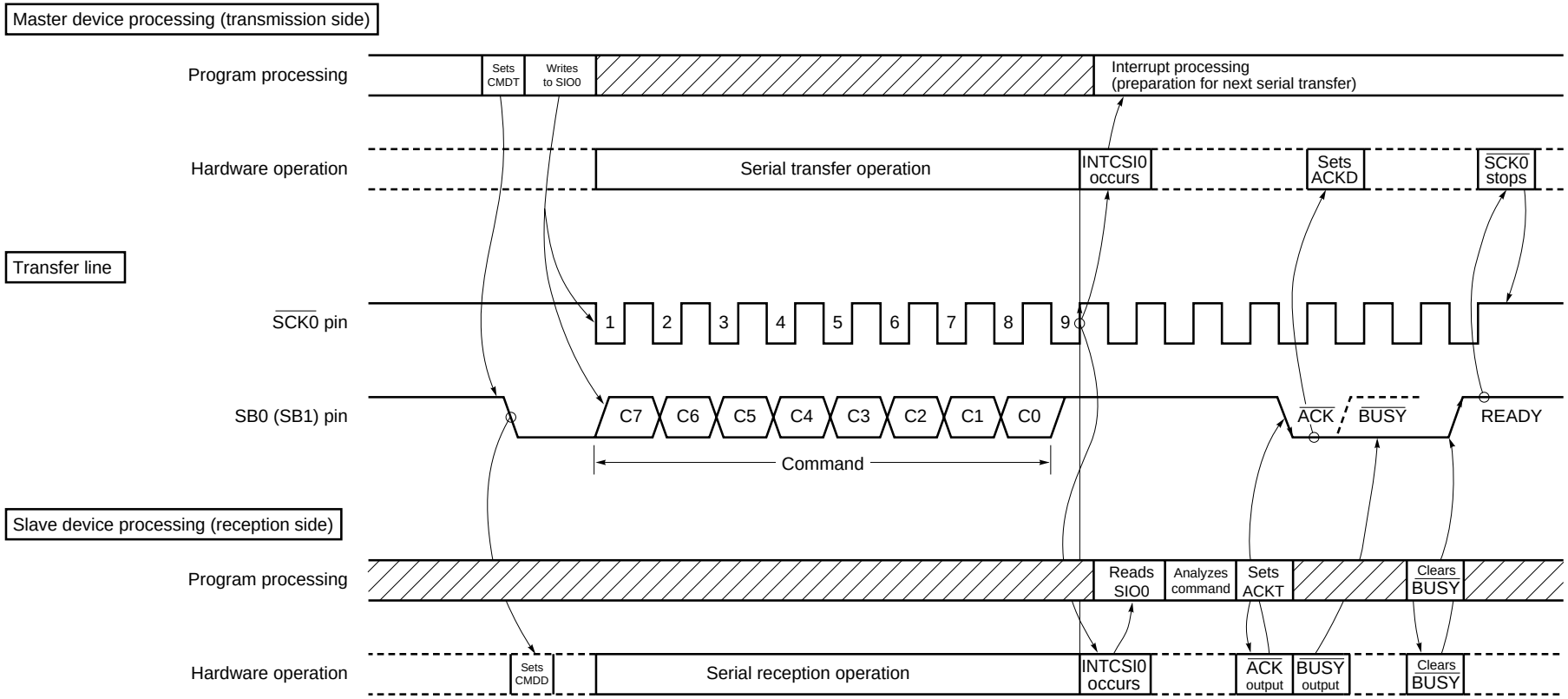


Figure 13-28. Data Transmission Operation from Master Device to Slave Device

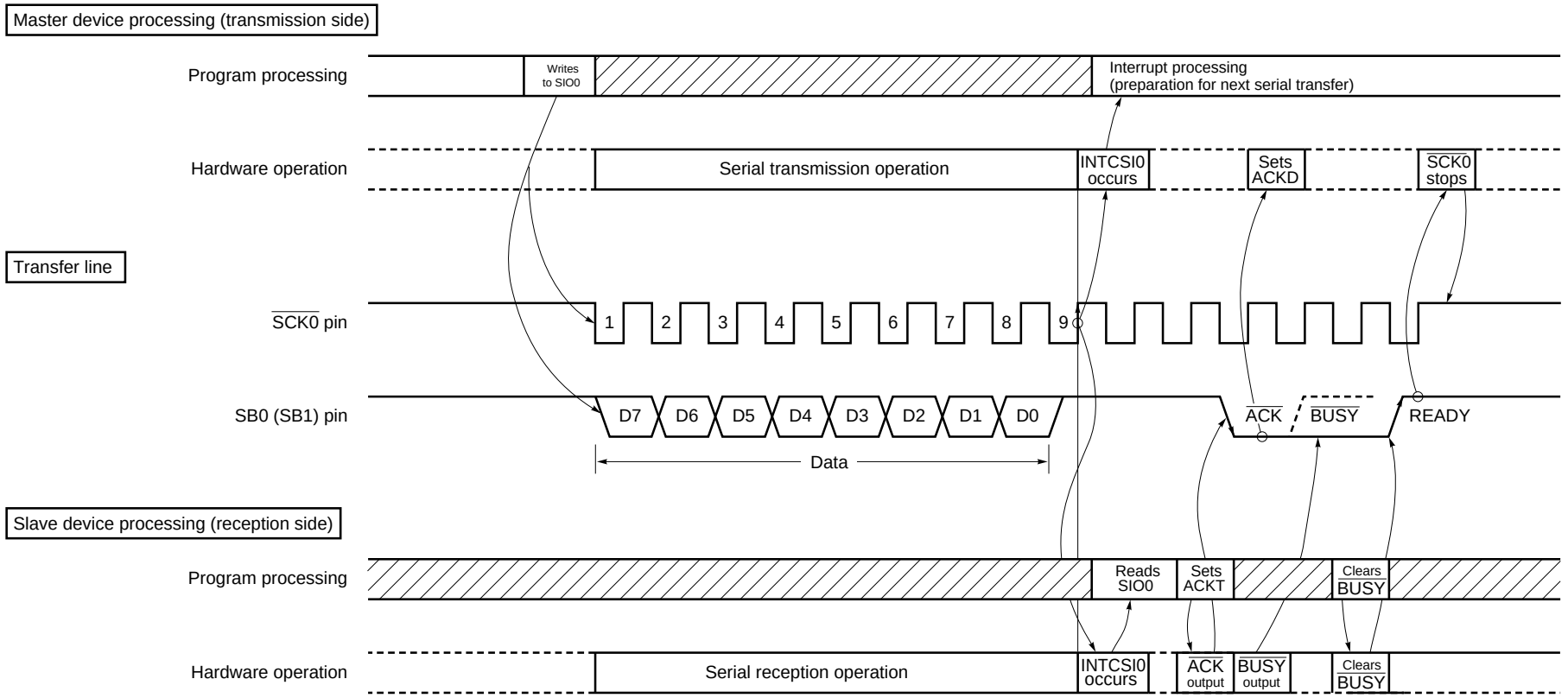
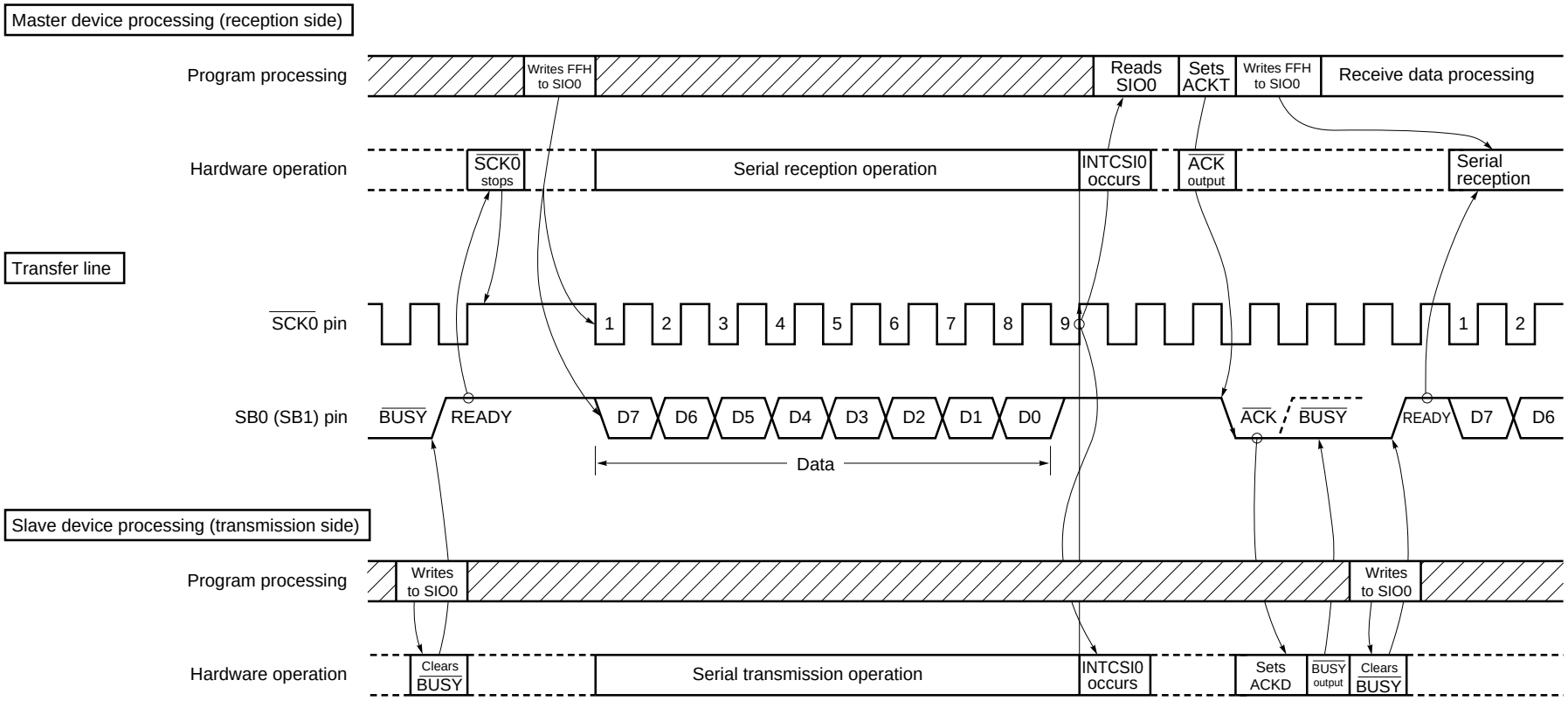


Figure 13-29. Data Transmission Operation from Slave Device to Master Device



(9) Transfer start

Serial transfer is started by setting the transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied:

- Operation control bit of serial interface channel 0 (CSIE0) = 1
- Internal serial clock is stopped or $\overline{SCK0}$ is high after 8-bit serial transfer

Cautions 1. Transfer is not started even when CSIE0 is set to "1" after data has been written to SIO0.

2. Because the N-ch open-drain must be set to high-impedance state during data reception, write FFH to SIO0 in advance.
However, when the wake-up function specification bit (WUP) = 1, the N-ch open-drain is always set to high-impedance state, and FFH needs not to be written to SIO0 before reception.

3. If data is written to SIO0 when the slave is busy, that data is not lost.
When SB0 (or SB1) input goes high (ready) after the busy status has been released, transfer is started.

When 8-bit transfer has been completed, serial transfer is automatically stopped, and an interrupt request flag (CSIIF0) is set.

Be sure to perform the following setting to the pin (SB0 or SB1) that is used to input/output data before serial transfer of 1 byte after the $\overline{RESET}$ signal has been input:

<1> Set 1 to the output latches of P25 and P26.
<2> Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.
<3> Set 0 to the output latches of P25 and P26 to which 1 has been set before.

(10) Checking busy status of a slave

Check whether a slave is in the busy status from the device in the master mode, in the following procedure:

<1> Detect generation of the acknowledge signal ($\overline{ACK}$) or interrupt request signal.
<2> Set the port mode register PM25 (or PM26) of the SB0/P25 (or SB1/P26) pin in the input mode.
<3> Read the status of the pin (if the pin is high, it is in the ready status).

After detecting the ready status, set 0 to the port mode register, to restore the output mode.

(11) Notes on SBI mode

(a) Whether a slave is selected or not is detected by matching of a slave address that has been received after the bus release signal has been issued (RELD = 1).
To detect matching of addresses, an address match interrupt request (INTCSI0) that is generated when WUP = 1 is usually used. Therefore, detect whether a slave is selected or not by reception of a slave address when WUP = 1.

(b) To detect whether a slave is selected or not when WUP = 0 without using the interrupt, do so by transmitting/receiving a command set by program in advance, instead of using the address matching method.

(c) In the SBI mode, output of the $\overline{\text{BUSY}}$ signal continues until the next serial clock ($\overline{\text{SCK0}}$) falling edge after a $\overline{\text{BUSY}}$ releasing command has been issued. If WUP is set to 1 by mistake during this period, the $\overline{\text{BUSY}}$ signal will not be released. For this reason, be sure to release the $\overline{\text{BUSY}}$ status, and make sure that the SB0 (SB1) pin has gone high before setting WUP = 1.

(d) Be sure to perform the following setting to the pin that is used to input/output data before serial transfer of 1 byte after the $\overline{\text{RESET}}$ signal has been input:

<1> Set 1 to the output latches of P25 and P26.
<2> Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.
<3> Set 0 to the output latches of P25 and P26 to which 1 has been set before.

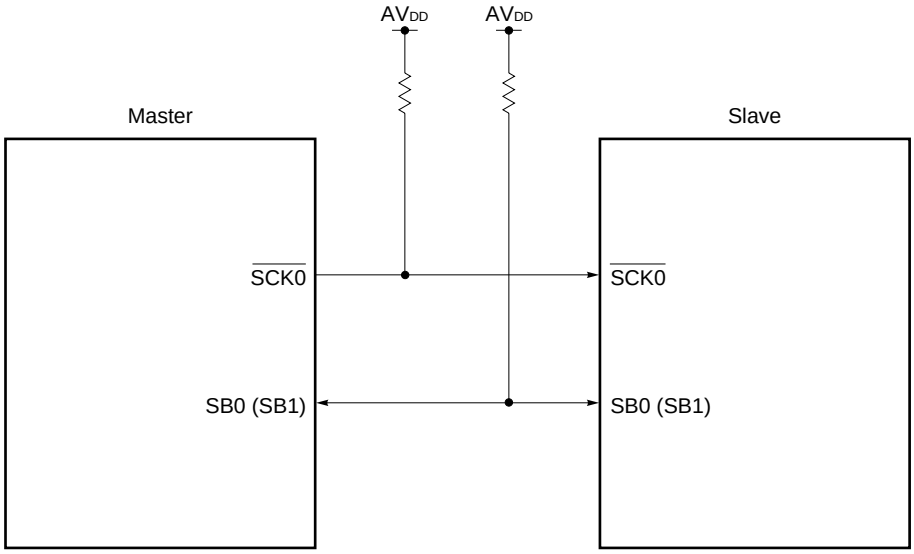
★ (e) A transition of the SB0 (SB1) line from low to high or high to low is recognized as a bus release signal or command signal while the $\overline{\text{SCK0}}$ line is high. Therefore shifting of the change timing of the bus due to the influence of the board capacitance, etc., may cause incorrect identification as a bus release signal (command signal), regardless of whether data is being transmitted. For this reason, special care must be taken regarding wiring.

13.4.4 Operation in 2-wire serial I/O mode

The 2-wire serial I/O mode can be used with any communication format by program.

Basically, two lines, serial clock ($\overline{SCK0}$) and serial data I/O (SB0 or SB1), are used to establish communication in this mode.

Figure 13-30. Example of Serial Bus Configuration by 2-wire Serial I/O



(1) Register setting

The 2-wire serial I/O mode is set by using the serial operation mode register 0 (CSIM0), serial bus interface control register (SBIC), and interrupt timing specification register (SINT).

(a) Serial operation mode register 0 (CSIM0)

CSIM0 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{RESET}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W[Note 1]

R/W

CSIM01	CSIM00	Selects clock of serial interface channel 0
0	×	Clock externally input to $\overline{\text{SCK0}}$ pin
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operation mode	First bit	SIO/SB0/P25 pin function	SO0/SB1/P26 pin function	$\overline{\text{SCK0}}$/P27 pin function
0	×	3-wire serial I/O mode (Refer to **13.4.2 Operation in 3-wire serial I/O mode**.)											
1	0	SBI mode (Refer to **13.4.3 Operation in SBI mode**.)											
1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open drain I/O)	$\overline{\text{SCK0}}$ (N-ch open drain I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open drain I/O)	P26 (CMOS I/O)	

R/W

WUP	Controls wake-up function[Note 3]
0	Generates interrupt request signal in all modes each time serial transfer is executed
1	Generates interrupt request signal when address received after bus has been released in SBI mode (when CMDD = RELD = 1) coincides with data of slave address register (SVA)

R

COI	Slave address comparison result flag[Note 4]
0	Data of slave address register (SVA) does not coincide with data of serial I/O shift register 0 (SIO0)
1	Data of slave address register (SVA)coincides with data of serial I/O shift register 0 (SIO0)

R/W

CSIE0	Controls operation of serial interface channel 0
0	Stops operation
1	Enables operation

Notes 1. Bit 6 (COI) is a read-only bit.

2. This pin can be used freely as a port pin.

3. Be sure to set WUP to 0 in the 2-wire serial I/O mode.

4. COI is 0 when CSIE0 = 0.

Remark × : don't care

PM×× : port mode register

P×× : output latch of a port

(b) Serial bus interface control register (SBIC)

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W		
	RELT	SO0 latch is set to 1 when RELT = 1. After setting SO0 latch, RELT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

R/W		
	CMDT	SO0 latch is cleared to 0 when CMDT = 1. After clearing SO0 latch, CMDT is automatically cleared to 0. This bit is also cleared to 0 when CSIE0 = 0.

CSIE0: bit 7 of serial operation mode register 0 (CSIM0)

(c) Interrupt timing specification register (SINT)

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	7	<6>	<5>	<4>	3	2	1	0	Address	After reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W[Note 1]

R/W

SIC	Selects interrput source of INTCSI0
0	Sets CSIIF0 at end of transfer by serial interface channel 0
1	Sets CSIIF0 at end of transfer by serial interface channel 0 or on detection of bus release

R

CLD	Level of $\overline{\text{SCK0}}$/P27 pin[Note 2]
0	Low level
1	High level

Notes 1. Bit 6 (CLD) is a read-only bit.

2. CLD is 0 when CSIE0 = 0.

Caution Be sure to set bits 0 through 3 to 0.

Remark CSIIF0 : interrupt request flag for INTCSI0

CSIE0 : bit 7 of serial operation mode register 0 (CSIM0)

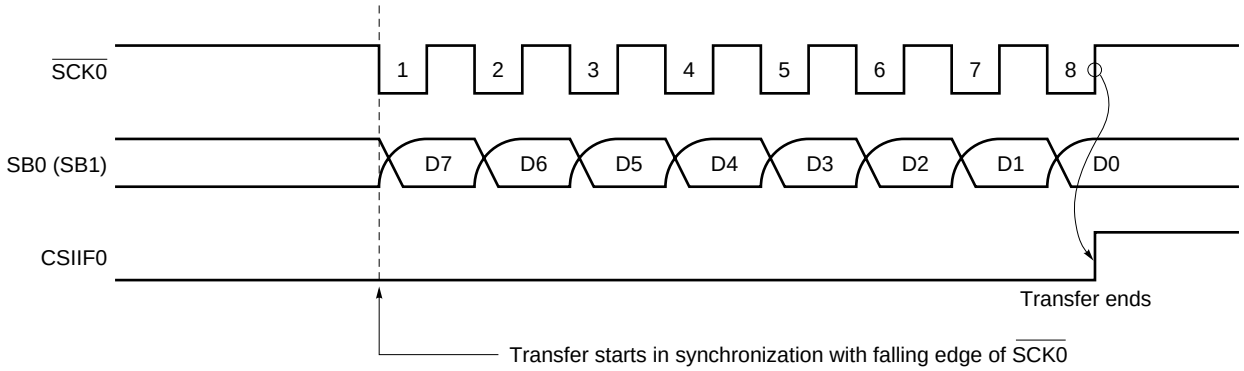
(2) Communication operation

In the 2-wire serial I/O mode, data is transmitted/received in 8-bit units. Data is transmitted/received on a 1-bit-by-1-bit basis in synchronization with the serial clock.

The shift operation of the serial I/O shift register 0(SIO0) is performed in synchronization with the falling edge of the serial clock ($\overline{SCK0}$). The transmitted data is retained by the SO0 latch and output from the SB0/P25 (or SB1/P26) pin, starting from the MSB. The received data input from the SB0 (or SB1) pin is latched to SIO0 at the rising edge of $\overline{SCK0}$.

When the 8-bit data has been completely transferred, the operation of SIO0 is automatically stopped, and an interrupt request flag (CSIIF0) is set.

Figure 13-31. Timing of 2-wire Serial I/O Mode



The pin of the SB0 (or SB1) pin specified as the serial data bus must be externally pulled up because this pin is an N-ch open drain I/O pin. When data is received, write FFH to SIO0 in advance because the N-ch transistor must be set to high impedance.

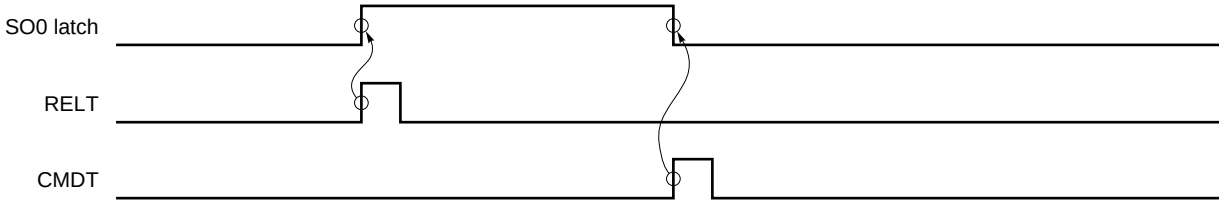
Because the SB0 (or SB1) pin outputs the status of the SO0 latch, the output status of the SB0 (or SB1) pin can be manipulated by setting the bit 0 (RELT) and bit 1 (CMDT) of serial bus interface control register (SBIC). However, do not manipulate the output status of the pin during serial transfer.

The output level of the $\overline{SCK0}$ pin is controlled by manipulating the P27 output latch in the output mode (mode of the internal system clock) (refer to **13.4.5 Manipulating $\overline{SCK0}$/P27 pin output**).

(3) Signals

Figure 13-32 shows the operations of RELT and CMDT.

Figure 13-32. Operations of RELT and CMDT



(4) Transfer start

Serial transfer is started by setting the transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied:

- Operation control bit of serial interface channel 0 (CSIE0) = 1
- When internal serial clock is stopped or $\overline{SCK0}$ is high after 8-bit serial transfer

Cautions 1. Even if CSIE0 is set to "1" after data has been written to SIO0, transfer is not started.

2. Write FFH to SIO0 in advance because the N-ch open drain must be set to high impedance during data reception.

Serial transfer is automatically stopped at the end of 8-bit transfer, and an interrupt request flag (CSIIF0) is set.

(5) Error detection

In the 2-wire serial I/O mode, the status of the serial bus SB0 (SB1) under transmission is also loaded to serial I/O shift register 0 (SIO0) of the device that is transmitting data; therefore, a transfer error can be detected by the following method:

(a) By comparing data of SIO0 before start of and after completion of transmission

In this case, it is judged that an transmission error has occurred if two data are different.

(b) By using slave address register (SVA)

The transmitted data is set to SIO0 and SVA and transmission is executed. After completion of transmission, the COI bit (match signal from address comparator) of the serial operation mode register 0 (CSIM0) is tested. If this bit is "1", it is judged that transmission has been completed normally. If it is "0", it is judged that a transmission error has occurred.

13.4.5 Manipulating $\overline{\text{SCK0}}$/P27 pin output

Because the $\overline{\text{SCK0}}$/P27 pin is provided with an output latch, it can also perform static output through software manipulation, in addition to output through the ordinary serial clock.

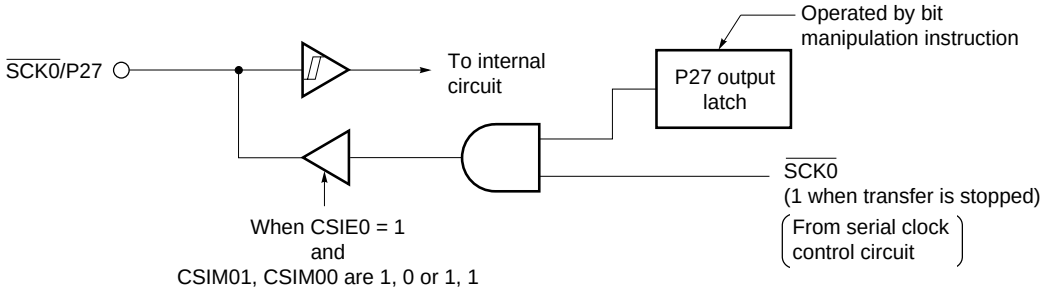
By manipulating the P27 output latch, the value of $\overline{\text{SCK0}}$ can be arbitrarily set by software (the SI0/SB0 and SO0/SB1 pins are controlled by bit 0 (RELT), bit 1 (CMDT) of serial bus interface control register (SBIC)).

The $\overline{\text{SCK0}}$/P27 pin output is manipulated as follows:

<1> Set the serial operation mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin: output mode, serial operation: enabled). $\overline{\text{SCK0}}$ = 1 while serial transfer is stopped.

<2> Manipulate the P27 output latch by using a bit manipulation instruction.

Figure 13-33. Configuration of $\overline{\text{SCK0}}$/P27 Pin



[MEMO]

CHAPTER 14 SERIAL INTERFACE CHANNEL 1

14.1 Function of Serial Interface Channel 1

Serial interface channel 1 has the following three modes:

Table 14-1. Modes of Serial Interface Channel 1

Operation Mode	Pins Used	Features	Applications
Operation stop mode	–	• Mode used when serial transfer is not executed • Can reduce power dissipation	–
3-wire serial I/O mode	$\overline{SCK1}$ (serial clock) SO1 (serial output) SI1 (serial input)	• Independent input and output lines. Short data transfer processing time because transmission and reception can be executed simultaneously • First bit of 8-bit data to be serial transferred can be specified to be MSB or LSB	Useful for connecting peripheral I/Os and display controllers with conventional clocked serial interface such as 75X/XL Series, 78K Series, and 17K Series
3-wire serial I/O mode with automatic transmission/ reception function (switchable between MSB first and LSB first)	$\overline{SCK1}$ (serial clock) SO1 (serial output) SI1 (serial input)	• This mode has all the 3-wire serial I/O mode functions with the automatic transmit/receive function added. • Can transmit/receive up to 32 bytes of data. Therefore, data can be transmitted/received by hardware to/from display controller/driver device for OSD (on-screen display) that operates independently of CPU. As a result, workload of software can be reduced.	

14.2 Configuration of Serial Interface Channel 1

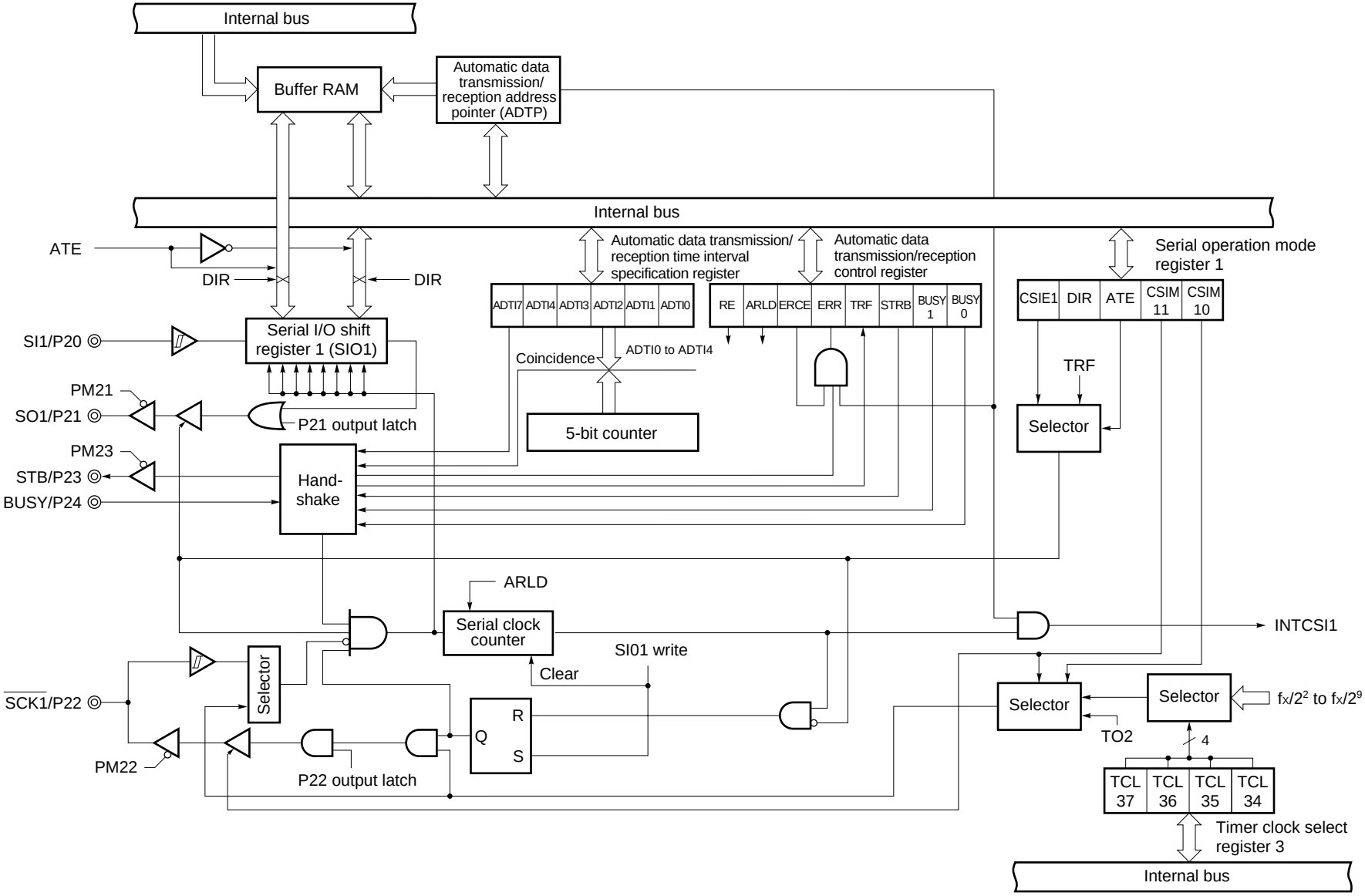
Serial interface channel 1 consists of the following hardware:

Table 14-2. Configuration of Serial Interface Channel 1

Item	Configuration
Register	Serial I/O shift register 1 (SIO1) Automatic data transmission/reception address pointer (ADTP)
Control register	Timer clock select register 3 (TCL3) Serial operation mode register 1 (CSIM1) Automatic data transmission/reception control register (ADTC) Automatic data transmission/reception time interval specification register (ADTI) Port mode register 2 (PM2)[Note] Port 2 (P2)

Note Refer to **Figure 4-6 Block Diagram of P20, P21, P23 to P26** and **Figure 4-7 Block Diagram of P22, P27**.

Figure 14-1. Block Diagram of Serial Interface Channel 1



(1) Serial I/O shift register 1 (SIO1)

This 8-bit register converts parallel data into serial data, and transmits/receives serial data (shift operation) in synchronization with the serial clock.

SIO1 is set by an 8-bit memory manipulation instruction.

When bit 7 (CSIE1) of the serial operation mode register 1 (CSIM1) is 1, the shift operation is started when data is written to SIO1.

When data is transmitted the data written to SIO1 is output to the serial output line (SO1). When data is received, it is read from the serial input line (SI1) to SIO1.

The contents of SIO1 become undefined when the $\overline{\text{RESET}}$ signal is input.

Caution Do not write data to SIO1 when the automatic transmission/reception operation is performed.

(2) Automatic data transmission/reception address pointer (ADTP)

This register stores a value of (number of transmission data bytes - 1) when the automatic transmission/reception function is performed. Its contents are automatically decremented when data transmission/reception is executed.

ADTP is set by an 8-bit memory manipulation instruction. At this time, set the higher 3 bits to 0.

The contents of this register are set to 00H when the $\overline{\text{RESET}}$ signal is input.

Caution Do not write data to ADTP when the automatic transmission/reception operation is performed.

(3) Serial clock counter

This counter counts the serial clocks output or input during transmission/reception operation, and checks whether 8-bit serial data has been transmission/received.

14.3 Registers Controlling Serial Interface Channel 1

The following four types of registers control serial interface channel 1:

- Timer clock select register 3 (TCL3)
- Serial operation mode register 1 (CSIM1)
- Automatic data transmission/reception control register (ADTC)
- Automatic data transmission/reception time interval specification register (ADTI)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 1.

TCL3 is set by an 8-bit memory manipulation instruction.

This register is set to 88H when the $\overline{\text{RESET}}$ signal is input.

Remark TCL3 also has a function to set the serial clock of serial interface channel 0 in addition to the function to set the serial clock of serial interface channel 1.

Figure 14-2. Format of Timer Clock Select Register 3

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Selects serial clock of serial interface channel 0
0	1	1	0	$f_X/2^{2}$ Note
0	1	1	1	$f_X/2^3$ (1.25MHz)
1	0	0	0	$f_X/2^4$ (625kHz)
1	0	0	1	$f_X/2^5$ (313kHz)
1	0	1	0	$f_X/2^6$ (156kHz)
1	0	1	1	$f_X/2^7$ (78.1kHz)
1	1	0	0	$f_X/2^8$ (39.1kHz)
1	1	0	1	$f_X/2^9$ (19.5kHz)
Others				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Selects serial clock of serial interface channel 1
0	1	1	0	$f_X/2^{2}$ Note
0	1	1	1	$f_X/2^3$ (1.25MHz)
1	0	0	0	$f_X/2^4$ (625kHz)
1	0	0	1	$f_X/2^5$ (313kHz)
1	0	1	0	$f_X/2^6$ (156kHz)
1	0	1	1	$f_X/2^7$ (78.1kHz)
1	1	0	0	$f_X/2^8$ (39.1kHz)
1	1	0	1	$f_X/2^9$ (19.5kHz)
Others				Setting prohibited

Note Can be set only when the main system clock oscillates at 4.19 MHz or less.

Caution To write data other than that already written to TCL3, stop the serial transfer once.

Remarks 1. f_X : main system clock oscillation frequency

2. ( ) : at f_X = 10.0 MHz

(2) Serial operation mode register 1 (CSIM1)

This register sets the serial clock and operation mode, and enables/disables the operation and automatic transmission/reception operation of serial interface channel 1.

CSIM1 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 14-3. Format of Serial Operation Mode Register 1

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Selects clock of serial interface channel 1
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin Note 1
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Selects operation mode of serial interface channel 1
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmission/reception function

DIR	First bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock count operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{\text{SCK1}}$/P22 pin function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation disabled	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enabled	Count operation	SI1 Note 3 (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. When external clock input is selected by setting CSIM11 to 0, set bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmission/reception control register (ADTC) to 0, 0.

2. These pins can be used freely as port pins.

3. When data is only transmitted, this pin can be used as P20 (CMOS I/O) (set bit 7 (RE) of ADTC to 0).

Remark × : don't care

PM×× : port mode register

P×× : output latch of a port

(3) Automatic data transmission/reception control register (ADTC)

This register enables/disables automatic transmission reception, operation mode, strobe output and busy input and indicates execution of automatic transmission/reception.

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

The contents of this register are reset to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 14-4. Format of Automatic Data Transmission/Reception Control Register

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	After reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Controls busy input
0	×	Does not use busy input
1	0	Enables busy input (high active)
1	1	Enables busy input (low active)

R/W

STRB	Controls strobe output
0	Disables strobe output
1	Enables strobe output

R

TRF	Status of automatic transmission/reception function
0	Detects end of automatic transmission/reception (set to 0 when automatic transmission/reception is aborted, or when ARLD = 0)
1	Automatic transfer/reception is in progress (set to 1 when this bit is written to SIO1)

R

ERR	Error detection of automatic transmission/reception function
0	No error during automatic transmission /reception (set to 0 when this bit is written to SIO1)
1	Error during automatic transmission/reception

R/W

ERCE	Controls error check of automatic transmission/ reception function
0	Disables error check during automatic transmission/ reception
1	Enables error check during automatic transmission/ reception (only when BUSY1 = 1)

R/W

ARLD	Selects operation mode of automatic transmission/ reception function
0	Single mode
1	Repeated mode

R/W

RE	Controls reception of automatic transmission/ reception function
0	Disables reception
1	Enables reception

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. The completion of automatic transmission/reception should be judged with TRF instead of the interrupt request flag (CSIIF1).

Caution Set STRB and BUSY1 of ADTC to 0, 0 when external clock input is selected by setting bit 1 (CSIM11) of the serial operation mode register 1 (CSIM1) to 0.

Remark ×: don't care

(4) Automatic data transmission/reception interval time specification register (ADTI)

This register sets the interval time at which data is transferred by the automatic transmission/reception function.

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

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 14-5. Format of Automatic Data Transmission/Reception Interval Time Specification Register (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Controls interval time of data transfer
0	Interval time not controlled by ADTI[Note 1]
1	Interval time controlled by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time of data transfer (f_X = 10.0-MHz operation)	
					Minimum value[Note 2]	Maximum value[Note 2]
0	0	0	0	0	18.4 μs + 0.5/f_{SCK}	20.0 μs + 1.5/f_{SCK}
0	0	0	0	1	31.2 μs + 0.5/f_{SCK}	32.8 μs + 1.5/f_{SCK}
0	0	0	1	0	44.0 μs + 0.5/f_{SCK}	45.6 μs + 1.5/f_{SCK}
0	0	0	1	1	56.8 μs + 0.5/f_{SCK}	58.4 μs + 1.5/f_{SCK}
0	0	1	0	0	69.6 μs + 0.5/f_{SCK}	71.2 μs + 1.5/f_{SCK}
0	0	1	0	1	82.4 μs + 0.5/f_{SCK}	84.0 μs + 1.5/f_{SCK}
0	0	1	1	0	95.2 μs + 0.5/f_{SCK}	96.8 μs + 1.5/f_{SCK}
0	0	1	1	1	108.0 μs + 0.5/f_{SCK}	109.6 μs + 1.5/f_{SCK}
0	1	0	0	0	120.8 μs + 0.5/f_{SCK}	122.4 μs + 1.5/f_{SCK}
0	1	0	0	1	133.6 μs + 0.5/f_{SCK}	135.2 μs + 1.5/f_{SCK}
0	1	0	1	0	146.4 μs + 0.5/f_{SCK}	148.0 μs + 1.5/f_{SCK}
0	1	0	1	1	159.2 μs + 0.5/f_{SCK}	160.8 μs + 1.5/f_{SCK}
0	1	1	0	0	172.0 μs + 0.5/f_{SCK}	173.6 μs + 1.5/f_{SCK}
0	1	1	0	1	184.8 μs + 0.5/f_{SCK}	186.4 μs + 1.5/f_{SCK}
0	1	1	1	0	197.6 μs + 0.5/f_{SCK}	199.2 μs + 1.5/f_{SCK}
0	1	1	1	1	210.4 μs + 0.5/f_{SCK}	212.0 μs + 1.5/f_{SCK}

Notes 1. The interval time is dependent on the CPU processin only.

2. The interval time for data transfer is variable. The minimum and maximum values of the interval time for transferring each data can be calculated by the following expressions (n: value placed in ADTI0 through ADTI4). If the minimum value calculated by the following expression is less than $2/f_{SCK}$, however, the minimum interval time is assumed to be $2/f_{SCK}$.

$$\text{Minimum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write data to ADTI while the automatic transmission/reception function is in use.

2. Be sure to set bits 5 and 6 to 0.

3. When the interval time for data transfer of automatic transfer/reception is controlled by using ADTI, busy control is disabled (refer to 14.4.3 (4) (a) Busy control option).

Remark f_X : main system clock oscillation frequency

f_{SCK} : serial clock frequency

Figure 14-5. Format of Automatic Data Transmission/Reception Interval Time Specification Register (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time for data transfer (f_X = 10.0-MHz operation)	
					Minimum value[Note]	Maximum value[Note]
1	0	0	0	0	223.2 μs + 0.5/f_{SCK}	224.8 μs + 1.5/f_{SCK}
1	0	0	0	1	236.0 μs + 0.5/f_{SCK}	237.6 μs + 1.5/f_{SCK}
1	0	0	1	0	248.8 μs + 0.5/f_{SCK}	250.4 μs + 1.5/f_{SCK}
1	0	0	1	1	261.6 μs + 0.5/f_{SCK}	263.2 μs + 1.5/f_{SCK}
1	0	1	0	0	274.4 μs + 0.5/f_{SCK}	276.0 μs + 1.5/f_{SCK}
1	0	1	0	1	287.2 μs + 0.5/f_{SCK}	288.8 μs + 1.5/f_{SCK}
1	0	1	1	0	300.0 μs + 0.5/f_{SCK}	301.6 μs + 1.5/f_{SCK}
1	0	1	1	1	312.8 μs + 0.5/f_{SCK}	314.4 μs + 1.5/f_{SCK}
1	1	0	0	0	325.6 μs + 0.5/f_{SCK}	327.2 μs + 1.5/f_{SCK}
1	1	0	0	1	338.4 μs + 0.5/f_{SCK}	340.0 μs + 1.5/f_{SCK}
1	1	0	1	0	351.2 μs + 0.5/f_{SCK}	352.8 μs + 1.5/f_{SCK}
1	1	0	1	1	364.0 μs + 0.5/f_{SCK}	365.6 μs + 1.5/f_{SCK}
1	1	1	0	0	376.8 μs + 0.5/f_{SCK}	378.4 μs + 1.5/f_{SCK}
1	1	1	0	1	389.6 μs + 0.5/f_{SCK}	391.2 μs + 1.5/f_{SCK}
1	1	1	1	0	402.4 μs + 0.5/f_{SCK}	404.0 μs + 1.5/f_{SCK}
1	1	1	1	1	415.2 μs + 0.5/f_{SCK}	416.8 μs + 1.5/f_{SCK}

Note The interval time for data transfer is variable. The minimum and maximum values of the interval time for transferring each data can be calculated by the following expressions (n: value placed in ADTI0 through ADTI4). If the minimum value calculated by the following expression is less than $2/f_{SCK}$, however, the minimum interval time is assumed to be $2/f_{SCK}$.

$$\text{Minimum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write data to ADTI while the automatic transmission/reception function is in use.

2. Be sure to set bits 5 and 6 to 0.

3. When the interval time for data transfer of automatic transfer/reception is controlled by using ADTI, busy control is disabled (refer to 14.4.3 (4) (a) Busy control option).

Remark f_X : main system clock oscillation frequency

f_{SCK} : serial clock frequency

14.4 Operation of Serial Interface Channel 1

Serial interface channel 1 operates in the following three operation modes:

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmission/reception function

14.4.1 Operation stop mode

Serial transfer is not executed in this mode. Consequently, the power dissipation can be reduced. The serial I/O shift register 1 (SIO1) can be used as an ordinary 8-bit register because it does not perform the sift operation.

In the operation stop mode, the P20/SI1, P21/SO1, P22/$\overline{\text{SCK1}}$, P23/STB, and P24/BUSY pins can be used as ordinary I/O port pins.

(1) Register setting

The operation stop mode is set by using the serial operation mode register 1 (CSIM1).

CSIM1 is set by using a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock count operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{\text{SCK1}}$/P22 pin function
0	×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Operation disabled	Cleared	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 2 1	Note 2 ×	0	0	1	×	Operation enabled	Count operation	SI1 Note 2 (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. These pins can be used freely as port pins.

2. P20 (CMOS I/O) is used when only transmission is executed. Clear bit 7 (RE) of the automatic data transmission/reception control register (ADCT) to 0.

Remark × : don't care

PM×× : port mode register

P×× : output latch of a port

14.4.2 Operation in 3-wire serial I/O mode

This mode is useful for connecting peripheral I/Os and display controllers that have the conventional clocked serial interface of the 75X/XL Series, 78K Series, and 17K Series.

In this mode, communication is established by using three signal lines: serial clock ($\overline{SCK1}$), serial output (SO1), and serial input (SI1).

(1) Register setting

The 3-wire serial I/O mode is set by using the serial operation mode register 1 (CSIM1).

CSIM1 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{RESET}$ signal is input.

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Selects clock of serial interface channel 1
0	×	Clock externally input to $\overline{SCK1}$ pin[Note 1]
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Selects operation mode of serial interface channel 1
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmission/reception function

DIR	First bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock count operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{SCK1}$/P22 pin function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation disabled	Cleared	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enabled	Count operation	SI1[Note 3] (input)	SO1 (CMOS output)	$\overline{SCK1}$ (input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes 1. When external clock input is selected by setting CSIM11 to 0, set bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmission/reception control register (ADTC) to 0, 0.

2. These pins can be used freely as port pins.

3. When data is only transmitted, this pin can be used as P20 (CMOS I/O) (set bit 7 (RE) of ADTC to 0).

Remark × : don't care

PM×× : port mode register

P×× : output latch of a port

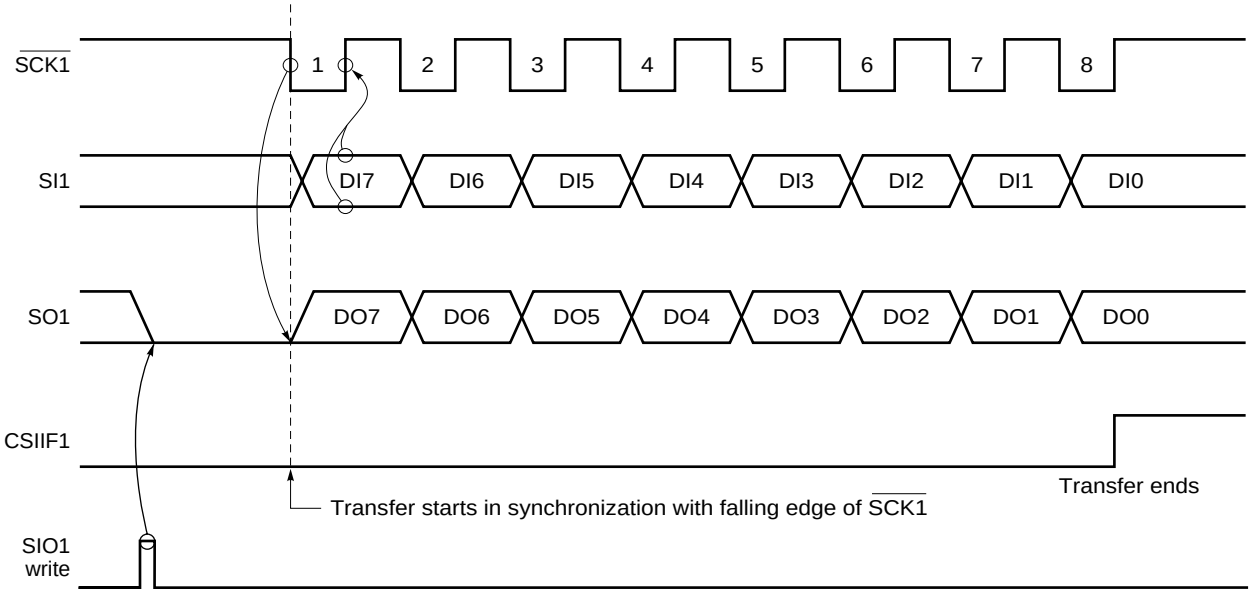
(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted/received in 8-bit units. Data is transmitted/received on a 1-bit-by-1-bit basis in synchronization with the serial clock.

The shift operation of the serial I/O shift register 1 (SIO1) is performed in synchronization with the falling edge of the serial clock ($\overline{SCK1}$). The transmitted data is retained by the SO1 latch and output from the SO1 pin. The receive data input to the SI1 pin is latched to SIO1 at the rising edge of $\overline{SCK1}$.

When the 8-bit data has been completely transferred, the operation of SIO1 is automatically stopped, and a interrupt request flag (CSIIF1) is set.

Figure 14-6. Timing of 3-wire Serial I/O Mode

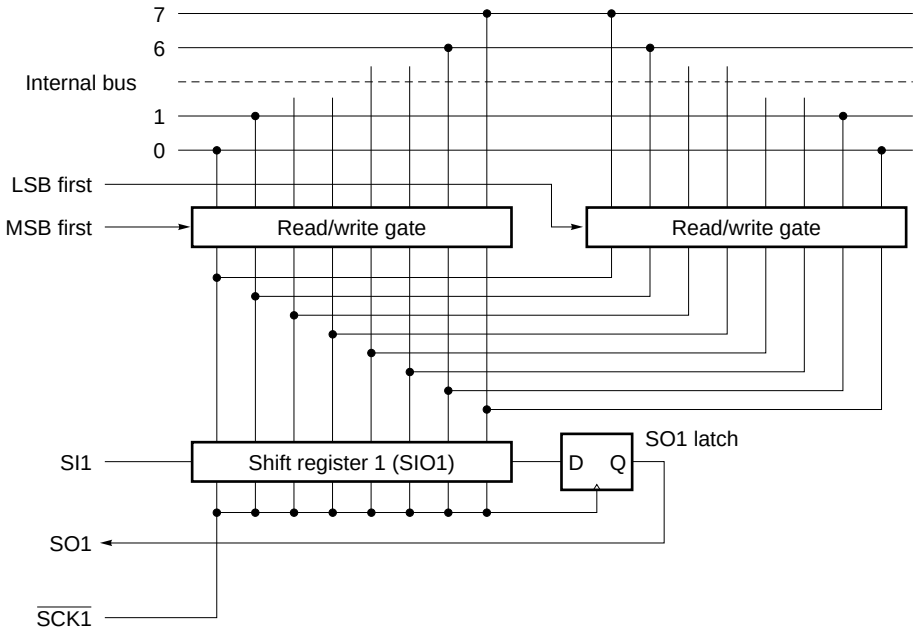


Caution The SIO1 pin goes low when SO1 is written.

(3) Switching transfer order (MSB first or LSB first)

The 3-wire serial I/O mode has a function to switch the transfer bit order, that is, MSB first or LSB first. Figure 14-7 shows the configuration of serial I/O shift register 1 (SIO1) and the internal bus. As shown in the diagram, the order of reading/writing the bus bits can be reversed with bit 6 (DIR) of serial operation mode register 1 (CSIM1).

Figure 14-7. Transfer Bit Order Switching Circuit



Note that, when the transfer data are written to shift register 1 (SIO1), the transfer bit order must have been changed; the shift direction of SIO1 does not change. Therefore, when the transfer order needs to be changed, first-bit specification must be made before the transfer data is written to SIO1.

(4) Starting transfer

Serial transfer is started by placing transfer data in serial I/O shift register 1 (SIO1) if the following two conditions are atisfied:

- Serial interface channel 1 operation control bit (CSIE1) = 1
- The internal serial clock is stopped or $\overline{\text{SCK1}}$ is high after 8-bit serial data has been transferred

Caution If CSIE1 is set to "1" after data has been written to SIO1, the transfer is not started.

Serial transfer is automatically stopped and an interrupt request flag (CSIIF1) is set after 8 bits of data have been transferred.

14.4.3 Operation in 3-wire serial I/O mode with automatic transmission/reception function

This 3-wire serial I/O mode is to transmit/receive data of up to 32 bytes without intervention by software. When transfer is started, data stored in RAM in advance can be transmitted by the set number of bytes, or data can be received by the set number of bytes and stored in RAM.

To transmit/receive data successively, handshake signals (STB and BUSY) are supported by hardware, so that OSD (On Screen Display) LSIs and peripheral LSIs such as LCD controllers/drivers can be easily connected.

(1) Register setting

The 3-wire serial I/O mode with automatic transmission/reception function is set by using the serial operation mode register 1 (CSIM1) and automatic data transmission/reception control register (ADTC) and automatic data transmission/reception time interval specification register (ADTI).

(a) Serial operation mode register 1 (CSIM1)

CSIM1 is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	6	<5>	4	3	2	1	0	Address	After reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Selects clock of serial interface channel 1
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin[Note 1]
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Selects operation mode of serial interface channel 1
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmission/reception function

DIR	First bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift register 1 operation	Serial clock counter operation control	SI1/P20 pin function	SO1/P21 pin function	$\overline{\text{SCK1}}$/P22 pin function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation disabled	Cleared	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enabled	Count operation	SI1[Note 3] (input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. When external clock input is selected by setting CSIM11 to 0, set bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmission/reception control register (ADTC) to 0, 0.

2. These pins can be used freely as port pins.

3. When data is only transferred, this pin can be used as P20 (CMOS I/O) (set bit 7 (RE) of ADTC to 0).

Remark × : don't care
PM×× : port mode register
P×× : output latch of a port

(b) Automatic data transmission/reception control register (ADTC)

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

The contents of this register are set to 00H when the $\overline{\text{RESET}}$ signal is input.

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	After reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Controls busy input
0	×	Does not use busy input
1	0	Enables busy input (high active)
1	1	Enables busy input (low active)

R/W

STRB	Controls strobe output
0	Disables strobe output
1	Enables strobe output

R

TRF	Status of automatic transmission/reception function Note 2
0	Detects end of automatic transmission/reception (set to 0 when automatic transmission/reception is aborted, or when ARLD = 0)
1	Automatic transmission/reception is in progress (set to 1 when this bit is written to SIO1)

R

ERR	Error detection of automatic transmission/reception function
0	No error during automatic transmission/reception (cleared to 0 when this bit is written to SIO1)
1	Error during automatic transmission/reception

R/W

ERCE	Controls error check of automatic transmission/ reception function
0	Disables error check during automatic transmission/ reception
1	Enables error check during automatic transmission/ reception (only when BUSY1 = 1)

R/W

ARLD	Selects operation mode of automatic transmission/ reception function
0	Single mode
1	Repeat mode

R/W

RE	Controls reception of automatic transmission/ reception function
0	Disables reception
1	Enables reception

Notes 1. Bits 3 and 4 (TRF and ERR) are read-only bits.

2. The completion of automatic transmission/reception should be judged with TRF, instead of the interrupt request flag (CSIIF1).

Caution Set STRB and BUSY1 of ADTC to 0, 0 when external clock input is selected by setting bit 1 (CSIM11) of the serial operation mode register 1 (CSIM1) to 0 (handshake control cannot be performed when an external clock is input).

Remark ×: don't care

(c) Automatic data transmission/reception interval time specification register (ADTI)

This register sets the interval time at which data is transferred by the automatic transmission/reception function.

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

This register is set to 00H when the $\overline{\text{RESET}}$ is input.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Controls interval time for data transfer
0	Interval time not controlled by ADTI[Note 1]
1	Interval time controlled by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time for data transfer (f_X = 10.0-MHz operation)	
					Minimum value[Note 2]	Maximum value[Note 2]
0	0	0	0	0	18.4 μs + 0.5/f_{SCK}	20.0 μs + 1.5/f_{SCK}
0	0	0	0	1	31.2 μs + 0.5/f_{SCK}	32.8 μs + 1.5/f_{SCK}
0	0	0	1	0	44.0 μs + 0.5/f_{SCK}	45.6 μs + 1.5/f_{SCK}
0	0	0	1	1	56.8 μs + 0.5/f_{SCK}	58.4 μs + 1.5/f_{SCK}
0	0	1	0	0	69.6 μs + 0.5/f_{SCK}	71.2 μs + 1.5/f_{SCK}
0	0	1	0	1	82.4 μs + 0.5/f_{SCK}	84.0 μs + 1.5/f_{SCK}
0	0	1	1	0	95.2 μs + 0.5/f_{SCK}	96.8 μs + 1.5/f_{SCK}
0	0	1	1	1	108.0 μs + 0.5/f_{SCK}	109.6 μs + 1.5/f_{SCK}
0	1	0	0	0	120.8 μs + 0.5/f_{SCK}	122.4 μs + 1.5/f_{SCK}
0	1	0	0	1	133.6 μs + 0.5/f_{SCK}	135.2 μs + 1.5/f_{SCK}
0	1	0	1	0	146.4 μs + 0.5/f_{SCK}	148.0 μs + 1.5/f_{SCK}
0	1	0	1	1	159.2 μs + 0.5/f_{SCK}	160.8 μs + 1.5/f_{SCK}
0	1	1	0	0	172.0 μs + 0.5/f_{SCK}	173.6 μs + 1.5/f_{SCK}
0	1	1	0	1	184.8 μs + 0.5/f_{SCK}	186.4 μs + 1.5/f_{SCK}
0	1	1	1	0	197.6 μs + 0.5/f_{SCK}	199.2 μs + 1.5/f_{SCK}
0	1	1	1	1	210.4 μs + 0.5/f_{SCK}	212.0 μs + 1.5/f_{SCK}

Notes 1. The interval time is dependent on the CPU processin only.

2. The interval time for data transfer is variable. The minimum and maximum values of the interval time for transferring each data can be calculated by the following expressions (n: value placed in ADTI0 through ADTI4). If the minimum value calculated by the following expression is less than $2/f_{SCK}$, however, the minimum interval time is assumed to be $2/f_{SCK}$.

$$\text{Minimum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write data to ADTI while the automatic transmission/reception function is in use.

2. Be sure to set bits 5 and 6 to 0.

3. When the interval time for data transfer of automatic transfer/reception is controlled by using ADTI, busy control is disabled (refer to 14.4.3 (4) (a) Busy control option).

Remark f_X : main system clock oscillation frequency

f_{SCK} : serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Specifies interval time for data transfer (f_X = 10.0-MHz operation)	
					Minimum value[Note]	Maximum value[Note]
1	0	0	0	0	223.2 μs + 0.5/f_{SCK}	224.8 μs + 1.5/f_{SCK}
1	0	0	0	1	236.0 μs + 0.5/f_{SCK}	237.6 μs + 1.5/f_{SCK}
1	0	0	1	0	248.8 μs + 0.5/f_{SCK}	250.4 μs + 1.5/f_{SCK}
1	0	0	1	1	261.6 μs + 0.5/f_{SCK}	263.2 μs + 1.5/f_{SCK}
1	0	1	0	0	274.4 μs + 0.5/f_{SCK}	276.0 μs + 1.5/f_{SCK}
1	0	1	0	1	287.2 μs + 0.5/f_{SCK}	288.8 μs + 1.5/f_{SCK}
1	0	1	1	0	300.0 μs + 0.5/f_{SCK}	301.6 μs + 1.5/f_{SCK}
1	0	1	1	1	312.8 μs + 0.5/f_{SCK}	314.4 μs + 1.5/f_{SCK}
1	1	0	0	0	325.6 μs + 0.5/f_{SCK}	327.2 μs + 1.5/f_{SCK}
1	1	0	0	1	338.4 μs + 0.5/f_{SCK}	340.0 μs + 1.5/f_{SCK}
1	1	0	1	0	351.2 μs + 0.5/f_{SCK}	352.8 μs + 1.5/f_{SCK}
1	1	0	1	1	364.0 μs + 0.5/f_{SCK}	365.6 μs + 1.5/f_{SCK}
1	1	1	0	0	376.8 μs + 0.5/f_{SCK}	378.4 μs + 1.5/f_{SCK}
1	1	1	0	1	389.6 μs + 0.5/f_{SCK}	391.2 μs + 1.5/f_{SCK}
1	1	1	1	0	402.4 μs + 0.5/f_{SCK}	404.0 μs + 1.5/f_{SCK}
1	1	1	1	1	415.2 μs + 0.5/f_{SCK}	416.8 μs + 1.5/f_{SCK}

Note The interval time for data transfer is variable. The minimum and maximum values of the interval time for transferring each data can be calculated by the following expressions (n: value placed in ADTI0 through ADTI4). If the minimum value calculated by the following expression is less than 2/f_{SCK}, however, the minimum interval time is assumed to be 2/f_{SCK}.

$$\text{Minimum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum value} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions
1. **Do not write data to ADTI while the automatic transmission/reception function is in use.**
2. **Be sure to set bits 5 and 6 to 0.**
3. **When the interval time for data transfer of automatic transfer/reception is controlled by using ADTI, busy control is disabled (refer to 14.4.3 (4) (a) Busy control option).**

Remark f_X : main system clock oscillation frequency
f_{SCK} : serial clock frequency

(2) Setting of automatic transmission/reception data

(a) Setting of transmitting data

<1> Write the transmit data from the lowest address FAC0H of the internal buffer RAM (up to FADFH). However, the data must be transmitted from the higher address to the lower address.
<2> Set the value of the number of transmit data bytes minus 1 to the automatic data transmission/reception address pointer (ADTP).

(b) Setting of automatic transmission/reception mode

<1> Set bit 7 (CSIE1) of the serial operation register 1 (CSIM1) to 1, and set bit 5 (ATE) to 1.
<2> Set bit 7 (RE) of the automatic transmission/reception control register (ADTC) to 1.
<3> Set a data transmission/reception transfer interval time to the automatic data transmission/reception time interval specification register (ADTI)
<4> Write any value to the serial I/O shift register 1 (SIO1) (transfer start trigger).

Caution Writing any value to SIO1 is to indicate the start of the automatic transmission/reception operation, and the written value has no meaning.

The following operation is automatically executed by setting (a) and (b) above.

- After the data in the internal buffer RAM specified by ADTP has been transferred to SIO1, transfer is executed (start of the automatic transmission/reception operation).
- Received data is written to an address of the buffer RAM specified by ADTP.
- The contents of ADTP are decremented, and the next data is transmitted/received. Data transmission/reception is performed until the output of the decrementer of ADTP reaches 00H, and the data at address FAC0H is output (end of the automatic transfer/reception operation).
- When the automatic transmission/reception is completed, TRF is cleared to 0.

(3) Communication operation

(a) Basic transmission/reception mode

This mode is to execute data transmission/reception in 8-bit units by the specified number of times, like in the 3-wire serial I/O mode.

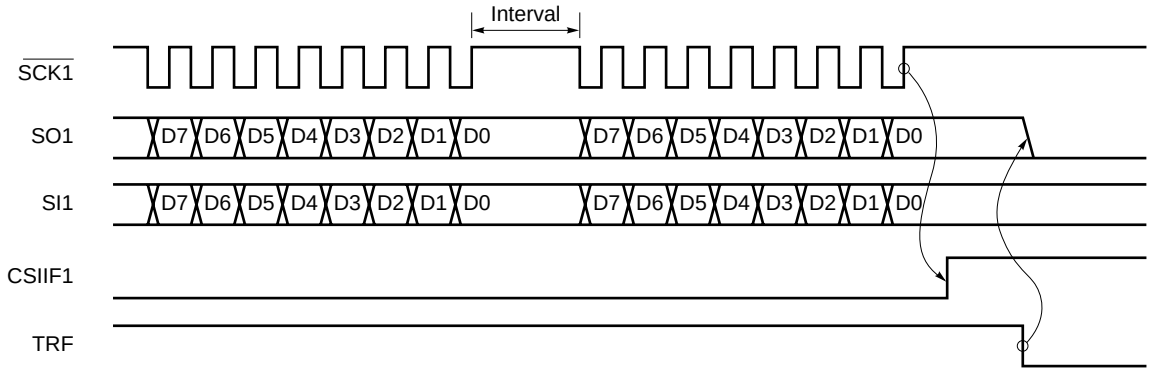
Serial transfer is started by writing any data to the serial I/O shift register 1 (SIO1) when the bit 7 (CSIE1) of the serial operation mode register 1 (CSIM1) is set to 1.

When the last byte has been completely transmitted, an interrupt request flag (CSIIF1) is set. However, use bit 3 (TRF) of the automatic data transmission/reception control register (ADTC), instead of CSIIF1, for judgement of automatic transmission/reception end.

When busy or strobe control is not performed, the P23/STB and P24/BUSY pins can be used as ordinary I/O port pins.

Figure 14-8 shows the operation timing of the basic transmission/reception mode, and Figure 14-9 shows an operation flowchart.

Figure 14-8. Operation Timing of Basic Transmission/Reception Mode



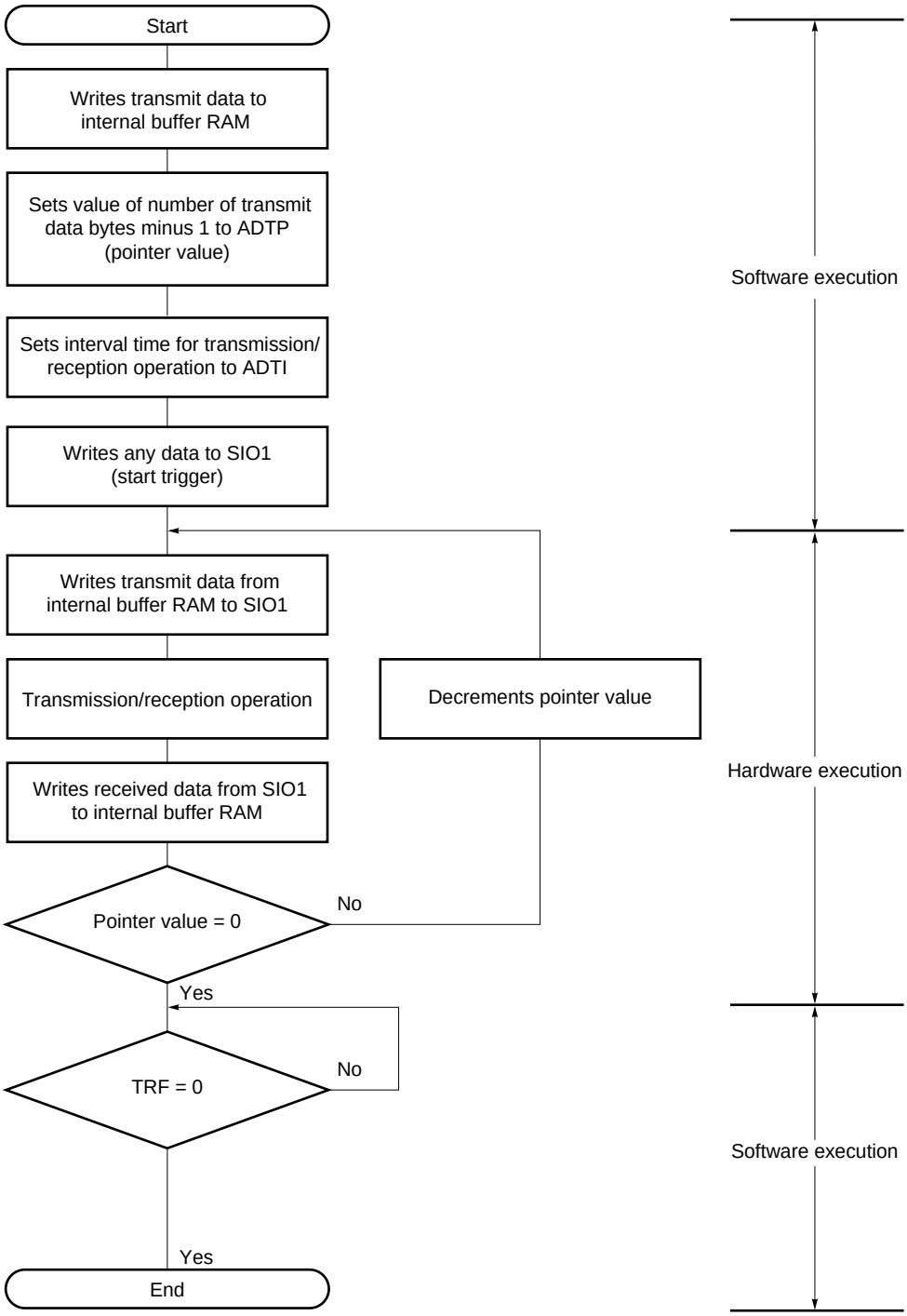
Cautions 1. In the basic transmission/reception mode, the internal buffer RAM is written/read after 1-byte data has been transmitted/received. Therefore, there is interval time until the next transmission/reception is executed. Because the internal buffer RAM is written/read simultaneously with the CPU processing, the maximum interval time depends on the CPU processing and a value of the automatic data transmission/reception time interval specification register (ADTI) (refer to (5) Interval time of automatic transmission/reception).

2. When TRF is cleared, the SO1 pin goes low.

Remark CSIIF1 : interrupt request flag

TRF : bit 3 of automatic data transmission/reception control register (ADTC)

Figure 14-9. Flowchart of Basic Transmission/Reception Mode



ADTP : automatic data transmission/reception address pointer
ADTI : automatic data transmission/reception time interval specification register
SIO1 : serial I/O shift register 1
TRF : bit 3 of automatic data transmission/reception control register (ADTC)

The internal buffer RAM operates as follows when 6 bytes are transmitted/received in the basic transmission/reception mode (ARLD = 0, RE = 1).

(i) Before transmission/reception (refer to Figure 14-10 (a))

Transmit data 1 (T1) is transferred from the buffer RAM to serial I/O shift register 1 (SIO1) after arbitrary data has been written to SIO1 (start trigger: this data is not transferred). When the first byte has been completely transmitted, receive data 1 (R1) is transferred from SIO1 to buffer RAM, and automatic data transmission/reception address pointer (ADTP) is decremented. Subsequently, transmit data 2 (T2) is transferred from buffer RAM to SIO1.

(ii) When 4th byte is transmitted/received (refer to Figure 14-10 (b))

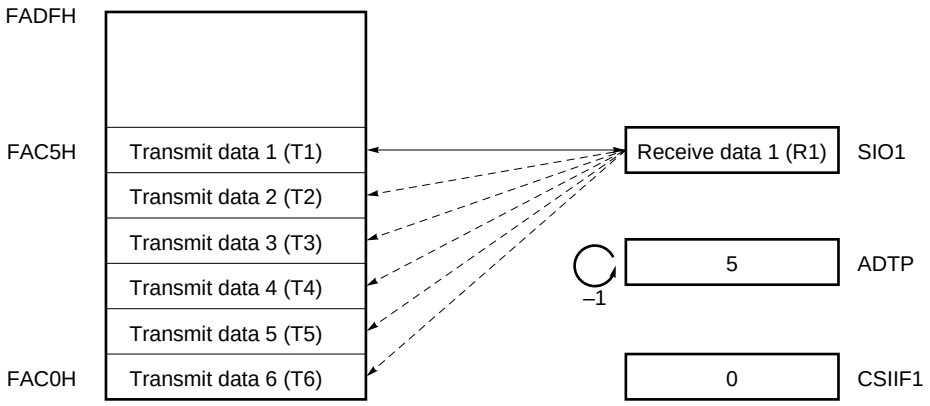
When the third byte has been transmitted/received completely, transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When the fourth byte has been transmitted, receive data 4 (R4) is transferred from SIO1 to buffer RAM, and ADTP is decremented.

(iii) End of transmission/reception (refer to Figure 14-10 (c))

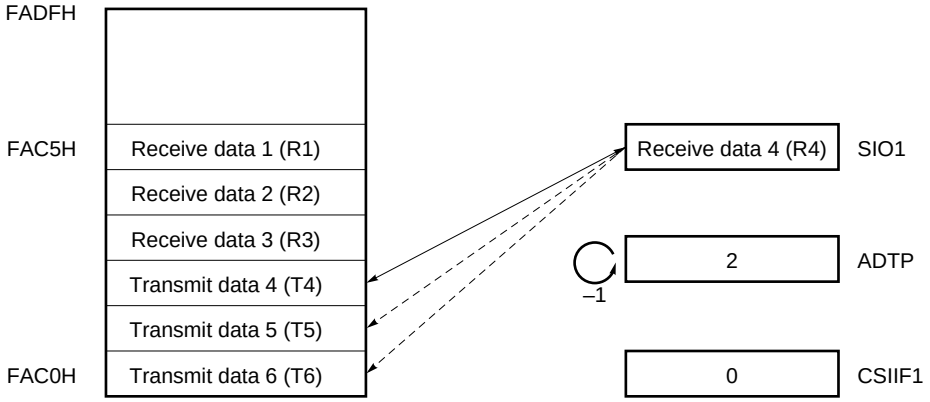
When the sixth byte has been transmitted, receive data 6 (R6) is transferred from SIO1 to the buffer RAM, and an interrupt request flag (CSIIF1) is set (INTCSI1 occurs).

Figure 14-10. Internal Buffer RAM Operation when 6 Bytes Are Transmitted/Received (in basic transmission/reception mode)

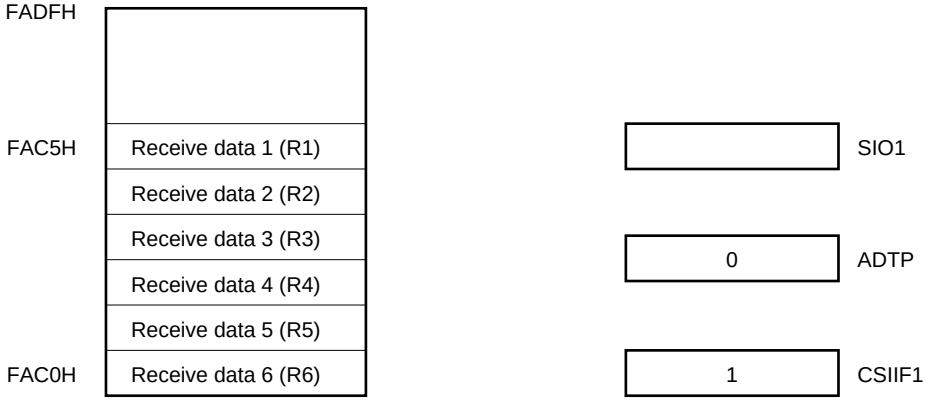
(a) Before transmission/reception



(b) When 4th byte has been transmitted/received



(c) At end of transmission/reception



(b) Basic transmission mode

This mode is to execute data transmission in 8-bit units by the specified number of times.

Serial transfer is started by writing any data to the serial I/O shift register 1 (SIO1) when the bit 7 (CSIE1) of the serial operation mode register 1 (CSIM1) is set to 1.

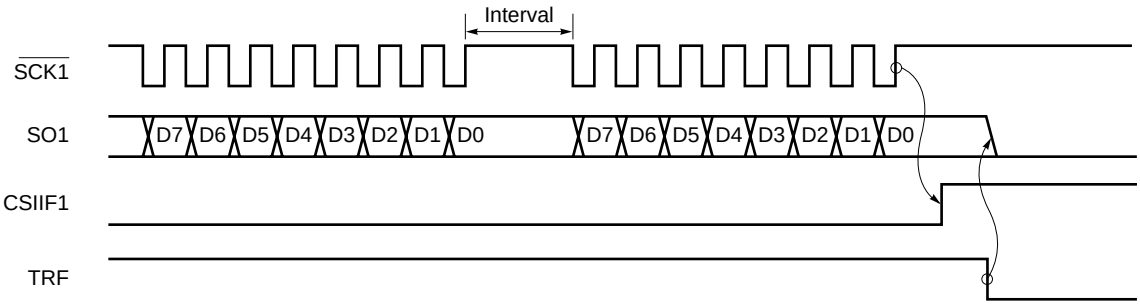
The interrupt request flag (CSIIF1) is set when the last byte has been completely transmitted. However, use bit 3 (TRF) of the automatic data transmission/reception control register (ADTC), instead of CSIIF1, for judgement of automatic transmission/reception end.

When reception operation, busy control, and strobe control are not performed, the P20/SI1, P23/STB, and P24/BUSY pins can be used as ordinary I/O ports.

Figure 14-11 shows the operation timing of the basic transmission mode, and Figure 14-12 shows an operation flowchart.

Figure 14-13 shows the operation of the internal buffer RAM when 6 bytes are transmitted.

Figure 14-11. Operation Timing of Basic Transmission Mode



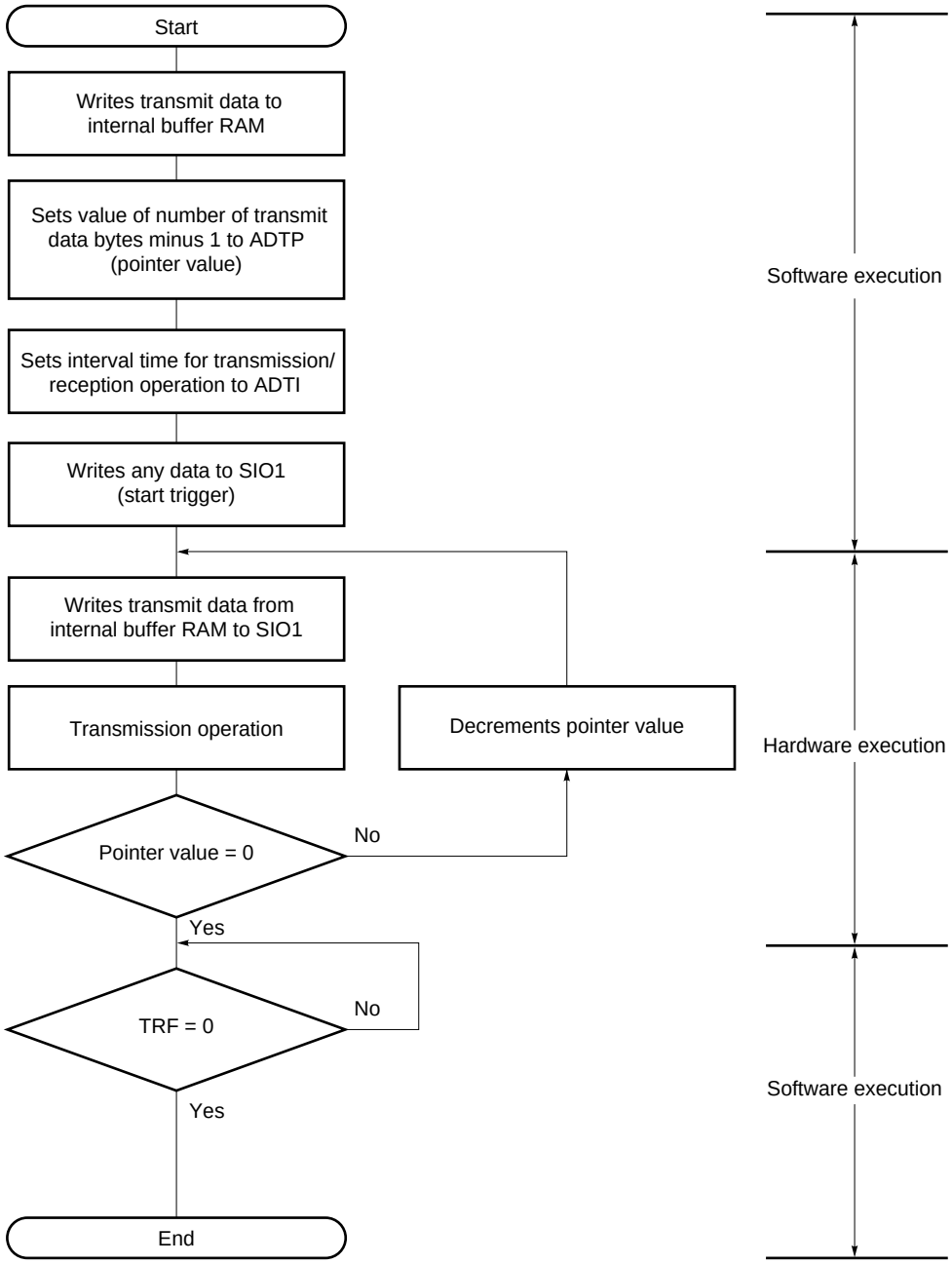
Cautions 1. In the basic transmission mode, the internal buffer RAM is read after 1-byte data has been transmitted. Therefore, there is an interval time until the next transmission is executed. Because the internal buffer RAM is read simultaneously with the CPU processing, the maximum interval time depends on the CPU processing and a value of the automatic data transmission/reception time interval specification register (ADTI) (refer to (5) Interval time of automatic transmission/reception).

2. When TRF is cleared, the SO1 pin goes low.

Remark CSIIF1 : interrupt request flag

TRF : bit 3 of automatic data transmission/reception control register (ADTC)

Figure 14-12. Flowchart of Basic Transmission Mode



ADTP : automatic data transmission/reception address pointer
ADTI : automatic data transmission/reception time interval specification register
SIO1 : serial I/O shift register 1
TRF : bit 3 of automatic data transmission/reception control register (ADTC)

The internal buffer RAM operates as follows when 6 bytes are transmitted in the basic transmission mode (ARLD = 0, RE = 0).

(i) Before transmission (refer to Figure 14-13 (a))

Transfer data 1 (T1) is transferred from the internal buffer RAM to serial I/O shift register 1 (SIO1) after arbitrary data has been written to SIO1 (start trigger: this data is not transferred). When the first byte has been completely transferred, automatic data transmission/reception address pointer (ADTP) is decremented. Subsequently, transfer data 2 (T2) is transferred from internal buffer RAM to SIO1.

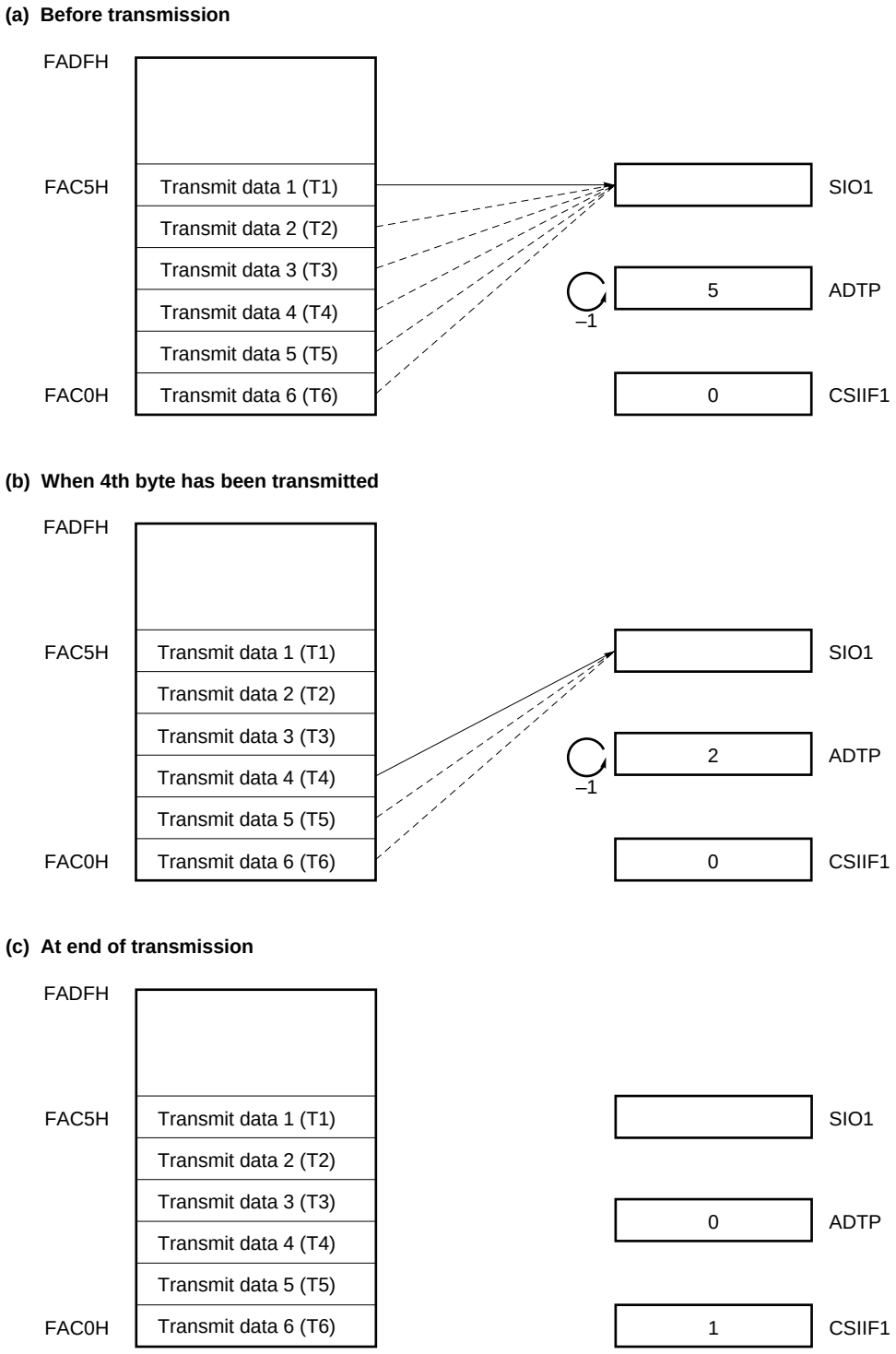
(ii) When 4th byte is transmitted (refer to Figure 14-13 (b))

When the third byte has been transmitted completely, transmit data 4 (T4) is transferred from the internal buffer RAM to SIO1. When the fourth byte has been transmitted, and ADTP is decremented.

(iii) End of transmission (refer to Figure 14-13 (c))

When the sixth byte has been transmitted, an interrupt request flag (CSIIF1) is set (INTCSI1 occurs).

Figure 14-13. Internal Buffer RAM Operation when 6 Bytes Are Transmitted (in basic transmission mode)



(c) Repetitive transmission mode

This mode is to repeatedly transmit the data stored in the internal buffer RAM.

The serial transfer is started by writing any data to the serial I/O shift register 1 (SIO1) when the bit 7 (CSIE1) of the serial operation mode register 1 (CSIM1) is set to 1.

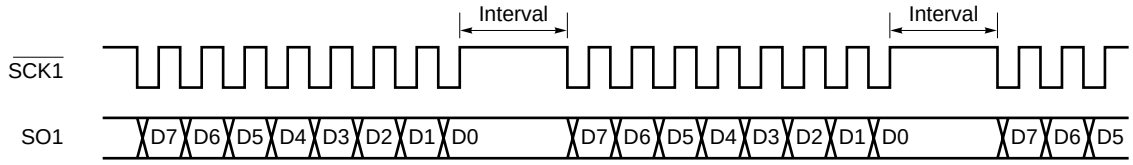
Unlike the basic transmission mode, the interrupt request flag (CSIIF1) is not set after the last byte (data at address FAC0H) has been transmitted, the value at which the transmission has been started is set again to the automatic data transmission/reception address pointer (ADTP), and the contents of the internal buffer RAM are transmitted again.

When reception operation, busy control, and strobe control are not performed, the P20/SI1, P23/STB, and P24/BUSY pins can be used as ordinary I/O ports.

Figure 14-14 shows the operation timing of the repetitive transmit mode, and Figure 14-15 shows an operation flowchart.

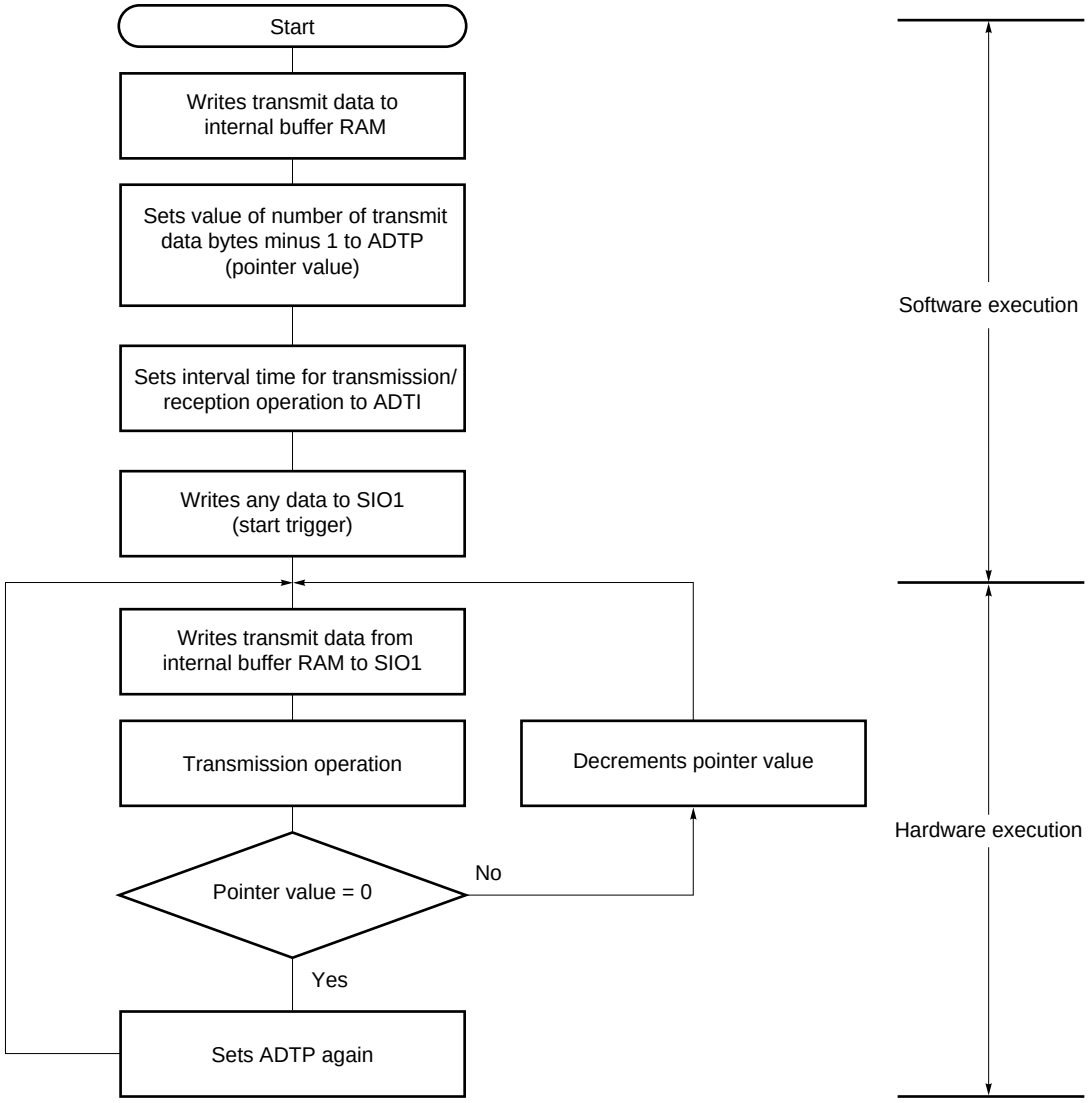
And, Figure 14-16 shows the operation of the internal buffer RAM when 6 bytes are transmitted in the repetitive transmission mode.

Figure 14-14. Operation Timing of Repetitive Transmission Mode



Caution In the repetitive transmission mode, the internal buffer RAM is read after 1-byte data has been transmitted. Therefore, there is interval time until the next transmission is executed. Because the internal buffer RAM is read simultaneously with the CPU processing, the maximum interval time depends on the CPU processing and the value of the automatic data transmission/reception time interval specification register (ADTI) (refer to (5) Interval time of automatic transmission/reception).

Figure 14-15. Flowchart of Repetitive Transmission Mode



ADTP : automatic data transmission/reception address pointer
ADTI : automatic data transmission/reception interval time specification register
SIO1 : serial I/O shift register 1

The internal buffer RAM operates as follows when 6 bytes are transmitted in the repetitive transmission mode (ARLD = 1, RE = 0).

(i) Before transmission (refer to Figure 14-16 (a))

Transmit data 1 (T1) is transferred from the internal buffer RAM to serial I/O shift register 1 (SIO1) after arbitrary data has been written to SIO1 (start trigger: this data is not transferred). When the first byte has been completely transmitted, automatic data transmission/reception address pointer (ADTP) is decremented. Subsequently, transmit data 2 (T2) is transferred from internal buffer RAM to SIO1.

(ii) When 6th byte has been transmitted (refer to Figure 14-16 (b))

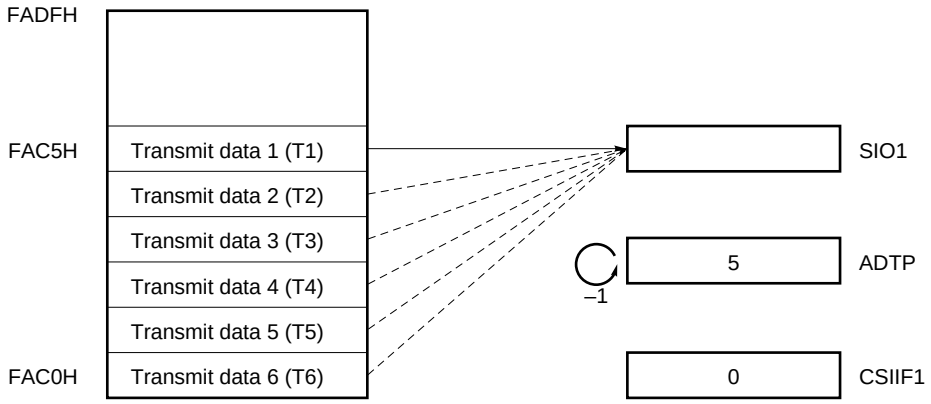
When the sixth byte has been transmitted, the interrupt request flag (CSIIF1) is not set.
ADTP will be reloaded with the initial pointer value.

(iii) When 7th byte is transmitted (refer to Figure 14-16 (c))

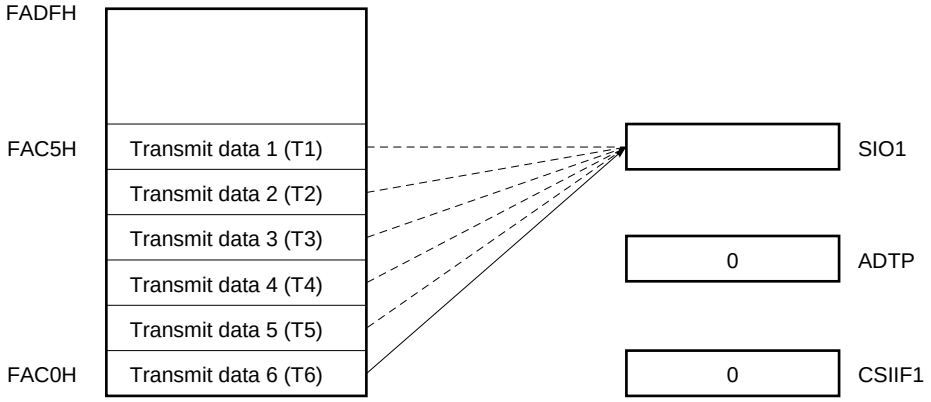
Transmit data 1 (T1) is transferred from the buffer RAM to SIO1 again. When the first byte has been completely transmitted, the ADTP is decremented. Subsequently, transmit data 2 (T2) is transferred from the internal buffer RAM to SIO1.

Figure 14-16. Internal Buffer RAM Operation when 6 Bytes Are Transmitted (in repetitive transmission mode)

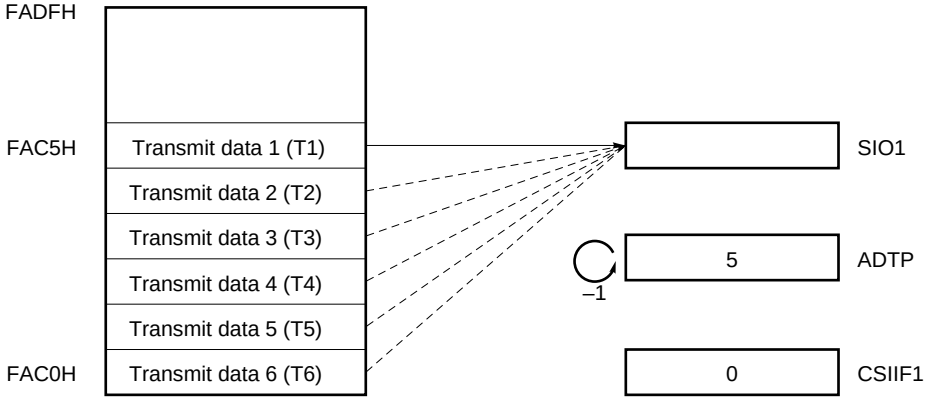
(a) Before transmission



(b) When 6th byte has been transmitted



(c) When 7th byte has been transmitted



(d) Stopping and resuming automatic transmission/reception

To temporarily stop the automatic transmission/reception operation under execution, reset the bit 7 (CSIE1) of the serial operation mode register 1 (CSIM1) to 0.

At this time, the operation is not stopped while 8-bit data is transferred. It is stopped after the 8-bit data currently transferred has been completely transferred.

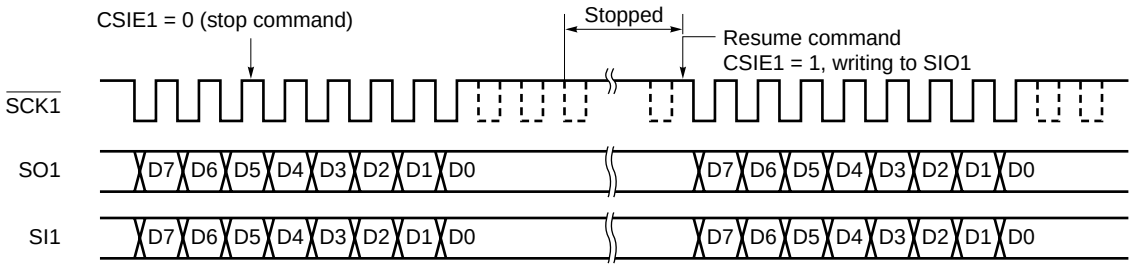
When the operation is stopped, the bit 3 (TRF) of the automatic data transmission/reception control register (ADTC) is cleared to 0, and all the port pins multiplexed with the serial interface pins (P20/SI1, P21/$\overline{SO1}$, P22/SCK1, P23/STB, and P24/BUSY) are set in the port mode after the eighth bit of the data has been transferred.

To resume the automatic transmission/reception, set CSIE1 to 1 and write any value to the serial I/O shift register 1 (SIO1). The remaining data can then be transferred.

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer is stopped and the HALT mode is set even if transfer of 8-bit data is in progress. When the HALT mode is released, the automatic transmission/reception operation is resumed from where it was stopped.

2. When automatic transmission/reception has been stopped, do not set the operation mode to 3-wire serial I/O mode while TRF = 1.

Figure 14-17. Stopping and Resuming Automatic Transmission/Reception



CSIE1: bit 7 of serial operation mode register 1 (CSIM1)

(4) Synchronization control

Busy control and shift control functions are used to synchronize transmission/reception between the master device and a slave device. Use of these functions enables the detection of bit shifts or other asynchronous conditions during transmission/reception.

(a) Busy control option

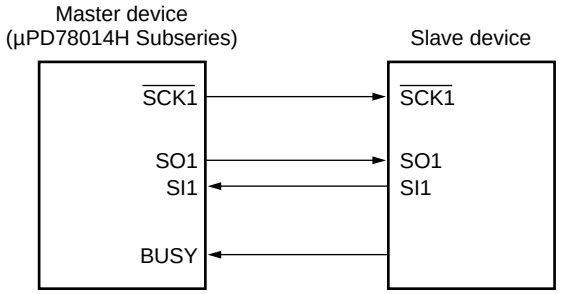
Busy control is a function which causes the transmission/reception of the master device to wait for the period when the busy signal is active. The busy signal is activated if a slave device sends a busy signal to the master device.

To use the busy control option, the following conditions are required.

- Bit 5 (ATE) of serial operation mode register 1 (CSIM1) is set to 1.
- Bit 1 (BUSY1) of the automatic data transmission/reception control register (ADTC) is set to 1.

The system configuration of master and slave devices with the busy control option used is shown in Figure 14-18.

Figure 14-18. System with Busy Control Option Enabled



If the slave device sends a busy signal, the master device receives the signal through its BUSY/P24 pin and samples the pin level at the falling edge of the serial clock. However, if the signal is sent during an 8-bit data transfer, wait cycles will not be inserted.

The condition in which a busy signal becomes valid is when the signal is activated at the rising edge of the serial clock two cycles after an 8-bit data transfer completion. If the condition is met, wait cycles will be inserted from that time until the busy signal becomes inactive.

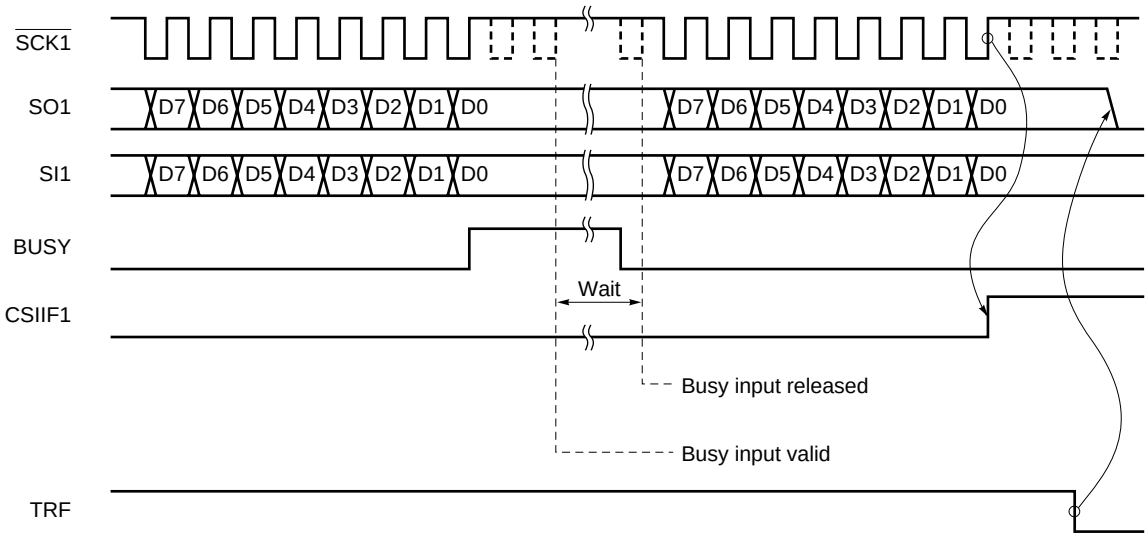
The active level of the busy signal can be changed as follows by setting bit 0 (BUSY0) of the ADTC register.

BUSY0 = 0 : Active high
BUSY0 = 1 : Active low

When using the busy control option, select the internal clock as the serial clock. With the external clock, control of busy signals is impossible. The operation timings when the busy control option is enabled is shown in Figure 14-19.

Caution When the busy control is working, do not use the interval time control by the automatic data transmission/reception interval specification register (ADTI); these two control functions are mutually exclusive. If use of both functions is attempted in the same period, busy control is automatically ignored.

Figure 14-19. Operation Timing when Busy Control Option Is Used (when BUSY0 = 0)



Caution When TRF is cleared, the SO1 pin goes low.

Remark CSIIF1: interrupt request flag
TRF : bit 3 of automatic data transmission/reception control register (ADTC)

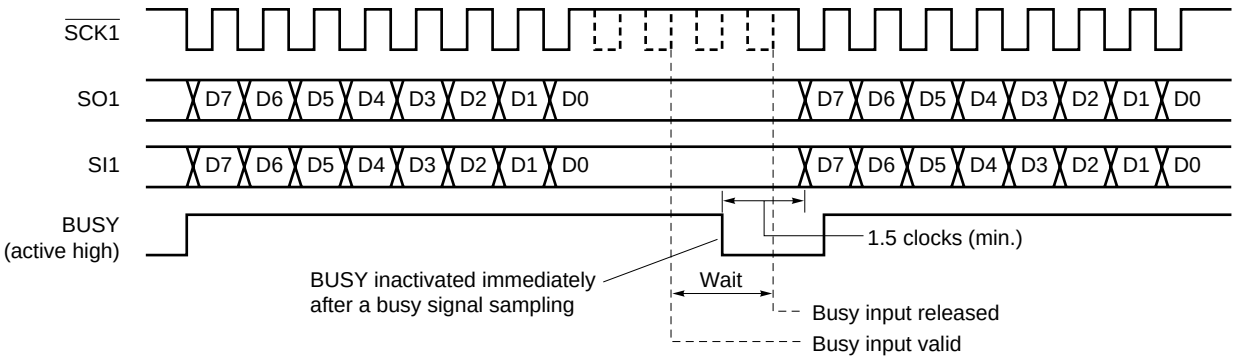
When the busy signal becomes inactive, that is, the master device detects the inactive level at the sampling pin, wait cycle insertion is terminated and transfer of the next 8-bit data starts.

Busy signals are sent asynchronously with the serial clock. Therefore, the busy signal which the slave inactivates may be recognized by the master device one clock cycle maximum after inactivation. In addition, it takes 0.5 clock cycles after the busy signal is recognized and before the next data transfer starts.

Therefore, in order to secure successful release of the wait state, the slave device should hold the inactive level of the busy signal at least 1.5 clock cycles.

The operation timings related to the busy signal and wait release are shown in Figure 14-20. This chart shows the case where the busy signal becomes active immediately after the start of transmission/reception.

Figure 14-20. Busy Signal and Wait Release Timing (BUSY0 = 0)



(b) Busy and strobe control option

Strobe control is a function that synchronizes the master device and slave devices during data transmission/reception operations. A strobe signal is output from the STB/P23 pin of the master device upon completion of an 8-bit transmission/reception. Receiving this signal, the slave device recognizes that the master device completed an 8-bit data transfer. Therefore, even if a bit shift occurs in a transfer due to noise on the serial clock line, the shift is rest by the strobe signal and does not affect succeeding data bytes; synchronized data transfer can be maintained in byte units.

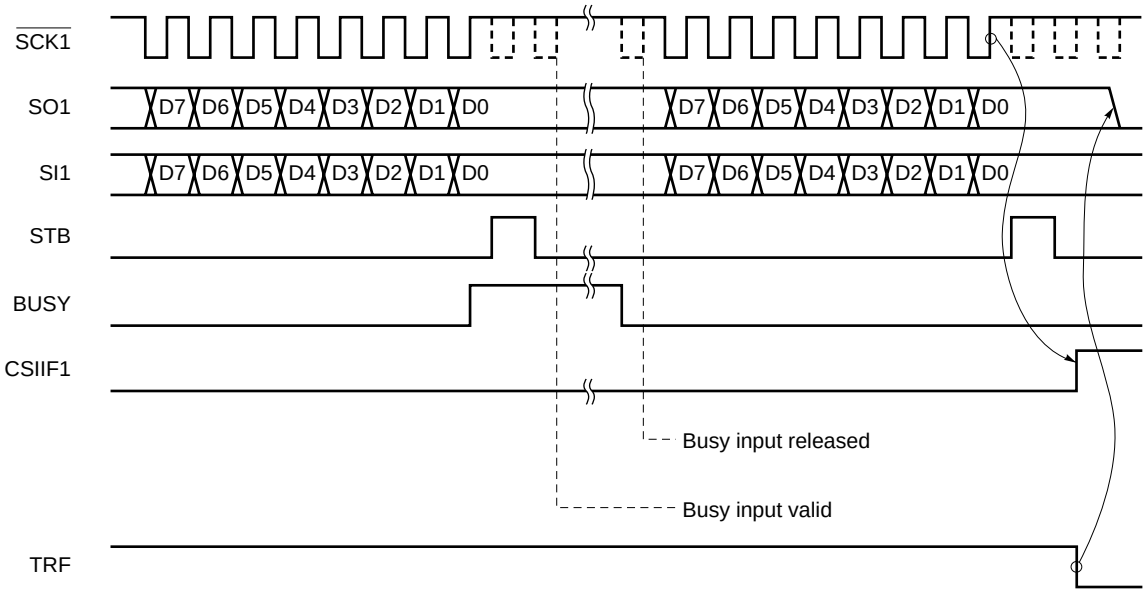
The following conditions must be met to use the strobe control option.

- Bit 5 (ATE) of serial operation mode register 1 (CSIM1) is set to 1.
- Bit 2 (STRB) of the automatic data transmission/reception control register (ADTC) is set to 1.

In most cases, busy control and strobe control options are used in combination as handshaking signals. If so, the actual operation is that, while sending a strobe signal from the STB/P23 pin, the master device samples the BUSY/P24 pin level, and suspends data transmission if the busy signal is active at that time. If the strobe control option is not used, the P23/STB pin can be used as a general-purpose input/output port pin.

Figure 14-21 shows the operation timings when both busy control and strobe control options are used. Note that, if the strobe control option is used, the interrupt request flag (CSIIF1), which is to be set at the completion of a transmission/reception, is set after a strobe signal is output.

Figure 14-21. Operation Timing when Busy and Strobe Control Option Is Used (when BUSY0 = 0)



Caution When TRF is cleared, the SO1 pin goes low.

Remark CSIIF1: interrupt request flag
TRF : bit 3 of automatic data transmission/reception control register (ADTC)

(c) Bit shift detection using busy signal

During an automatic transmission/reception operation, bit shifts on the serial clock from the master device to the slave device may occur due to electromagnetic noise. At this time, if the strobe control option is not enabled, the bit shift continues to be effective on the following bytes to be transferred. To prevent this, enable the busy control of the master device, which observes the busy signal during data transfer and detects bit shifts.

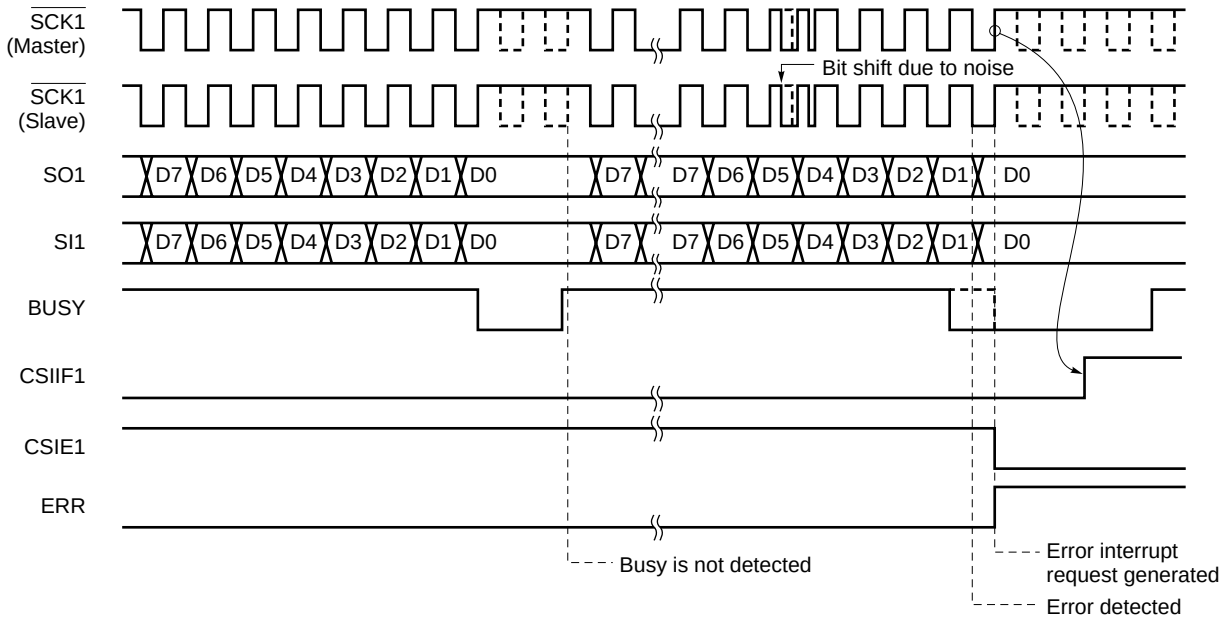
Detection of bit shifts based on the busy signal is performed as follows.

On the slave side, a busy signal is output at the rising edge of the 8th serial clock cycle in the data transfer. (At this time, if it is not preferable for wait cycles to be inserted by the generation of the busy signal, hold the busy signal inactive for two clock cycles after the generation of the busy signal.)

On the master side, the busy signal is sampled at each front-side falling edge of the serial clock. If there is no bit shift, the busy signal must be inactive in all eight sampling cycles. However, if a sampling result is active, the device determines that a bit shift has occurred, and thus starts the error processing, that is, sets bit 4 (ERR) of the automatic data transmission/reception control register (ADTC) to 1.

Figure 14-22 shows the operation timings of the bit shift detection making use of the busy signal.

Figure 14-22. Operation Timing of Bit Shift Detection with Busy Signal (when BUSY0 = 1)



CSIIF1 : interrupt request flag

CSIE1 : bit 7 of serial opration mode register 1 (CSIM1)

ERR : bit 4 of automatic data transmission/reception control register (ADTC)

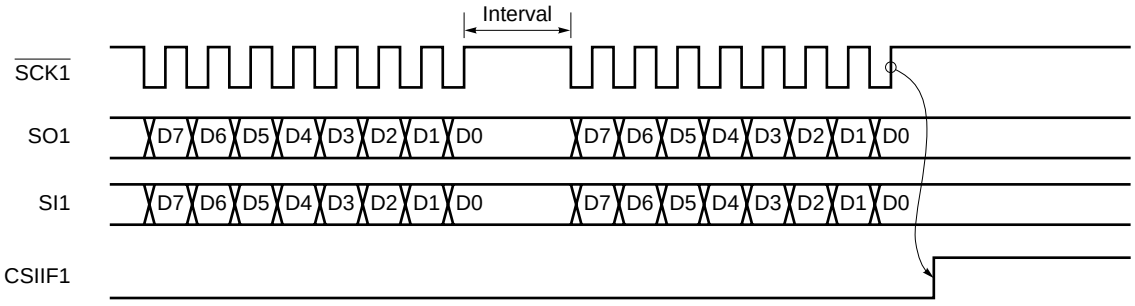
(5) Interval time of automatic transmission/reception

When using the automatic transmission/reception function and when 1-byte data has been transmitted/received, an interval time elapses until the next data is transmitted/received during which data is written to/read from the internal buffer RAM.

When the automatic transmission/reception function is executed with the internal clock, data is written to/read from the internal buffer RAM in parallel with the CPU processing. Therefore, the interval time depends on the CPU processing executed when the eighth serial clock rises and the set value of automatic data transmission/reception time interval specification register (ADTI). Whether the interval time is dependent on the value in ADTI can be selected by setting bit 7 (ADTI7) of ADTI. If ADTI7 is set to 0, the interval time is dependent on the CPU processing only. If ADTI7 is set to 1, the interval time is either that determined by the contents set to ADTI or the interval time dependent on the CPU processing, whichever is longer.

When using the automatic transfer/reception function with the external clock, an external clock that makes the interval time longer than that described in paragraph (b) below must be input.

Figure 14-23. Interval Time of Automatic Transmission/Reception



CSIIF1: interrupt request flag

(a) When using automatic transmission/reception function with internal clock

When bit 1 (CSIM11) of serial operation mode register 1 (CSIM1) is set to 1, the device operates with the internal clock.

When using the automatic transmission/reception function with the internal clock, the interval time varies as follows according to the CPU processing:

If automatic data transmission/reception interval time specification register (ADTI) is set to 0, the interval time is that determined by the CPU processing set. If bit 7 (ADT17) of ADTI is set to 1, the interval time is either that determined by the contents set to ADTI or the interval time dependent on the CPU processing, whichever is longer.

For the interval time set by ADTI, refer to **Figure 14-5 Format of Automatic Data Transmission/ Reception Interval Time Specification Register**.

Table 14-3. Interval Times by CPU Processing (with internal clock)

CPU Processing	Interval Time
When multiply instruction is used	MAX. ($2.5T_{SCK}$, $26T_{CPU}$)
When division instruction is used	MAX. ($2.5T_{SCK}$, $40T_{CPU}$)
External access 1 wait mode	MAX. ($2.5T_{SCK}$, $18T_{CPU}$)
Others	MAX. ($2.5T_{SCK}$, $14T_{CPU}$)

T_{SCK} : $1/f_{SCK}$

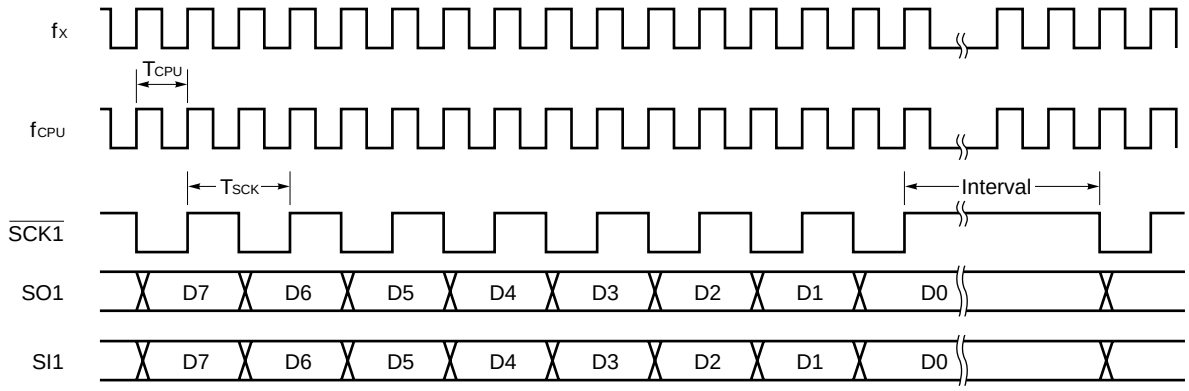
f_{SCK} : serial clock frequency

T_{CPU} : $1/f_{CPU}$

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of processor clock control register (PCC))

MAX. (a, b) : a or b whichever greater

Figure 14-24. Operation Timing When Automatic Transmission/Reception Function Is Executed with Internal Clock



f_X : main system clock oscillation frequency

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of processor clock control register (PCC))

T_{CPU} : $1/f_{CPU}$

T_{SCK} : $1/f_{SCK}$

f_{SCK} : serial clock frequency

(b) When using automatic transmission/reception function with external clock

When bit 1 (CSIM11) of serial operation mode register 1 (CSIM1) is set to 0, the device operates with the external clock.

To use the automatic transmission/reception function with the external clock, it is necessary to input an external clock that makes the interval time longer than the time shown below.

Table 14-4. Interval Times by CPU Processing (with external clock)

CPU Processing	Interval Time
When multiply instruction is used	$26T_{CPU}$ MIN.
When division instruction is used	$40T_{CPU}$ MIN.
External access 1 wait mode	$18T_{CPU}$ MIN.
Others	$14T_{CPU}$ MIN.

T_{CPU} : $1/f_{CPU}$

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of processor clock control register (PCC))

[MEMO]

CHAPTER 15 INTERRUPT FUNCTIONS AND TEST FUNCTIONS

15.1 Types of Interrupt Functions

The following three types of interrupt functions are available:

(1) Non-maskable interrupts

This interrupt is unconditionally accepted even in the interrupt disabled status. It is not subject to interrupt priority control and therefore takes precedence over all interrupt requests.
This interrupt generates a standby release signal.
As a non-maskable interrupt, one interrupt request from the watchdog timer is available.

(2) Maskable interrupts

These interrupts are subject to mask control, and can be divided into two groups according to the setting of the priority specification flag registers (PR0L, PR0H): one for higher priority and the other for lower priority. Higher-priority interrupts can nest lower-priority interrupts. The priority when two or more interrupt requests with the same priority occur at the same time is predetermined (refer to **Table 15-1**).
This interrupt generates a standby release signal.
As maskable interrupts, four external interrupt requests and 13 internal interrupt requests are available.

(3) Software interrupts

This is a vectored interrupt generated when the BRK instruction is executed and can be accepted even in the interrupt disabled status. This interrupt is not subject to interrupt priority control.

15.2 Interrupt Sources and Configuration

A total of 14 interrupt sources including non-maskable, maskable, and software interrupt sources are available (refer to **Table 15-1**).

Table 15-1. Interrupt Sources

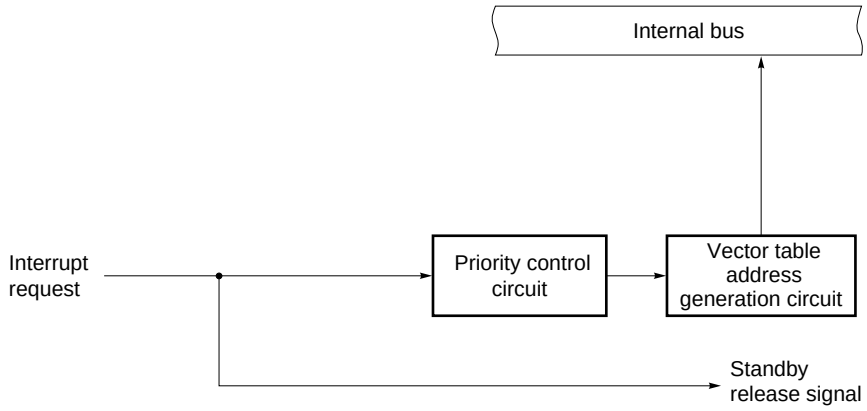
Interrupt Type	Default Priority [Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type [Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Overflow of watchdog timer (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Overflow of watchdog timer (when interval timer mode is selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	(D)
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTCSI0	End of transfer of serial interface channel 0	Internal	000EH	(B)
	6	INTCSI1	End of transfer of serial interface channel 1		0010H	
	7	INTTM3	Reference time interval signal from watch timer		0012H	
	8	INTTM0	Generation of coincidence signal from 16-bit timer/event counter		0014H	
	9	INTTM1	Generation of coincidence signal from 8-bit timer/event counter 1		0016H	
	10	INTTM2	Generation of coincidence signal from 8-bit timer/event counter 2		0018H	
	11	INTAD	End of conversion of A/D converter		001AH	
Software	—	BRK	Execution of BRK instruction	—	003EH	(E)

Notes 1. The default priority is used when two or more maskable interrupt requests occur at the same time. 0 is the highest and 11 is the lowest priority.

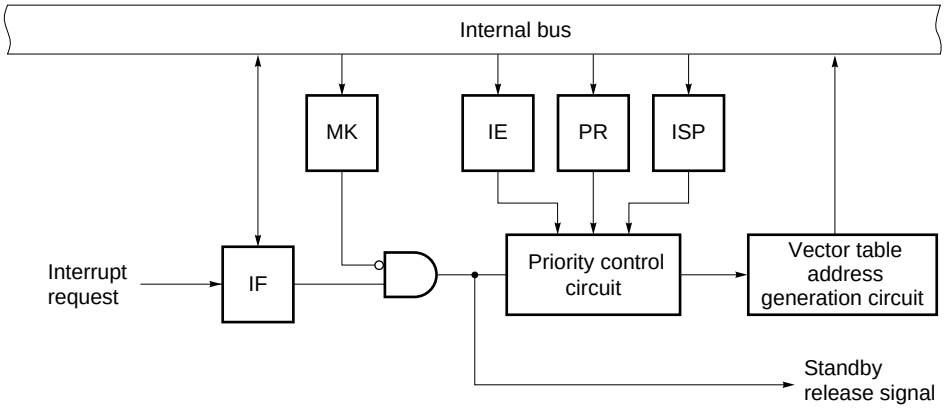
2. Basic configuration types (A) to (E) respectively correspond to (A) to (E) on the following pages.

Figure 15-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

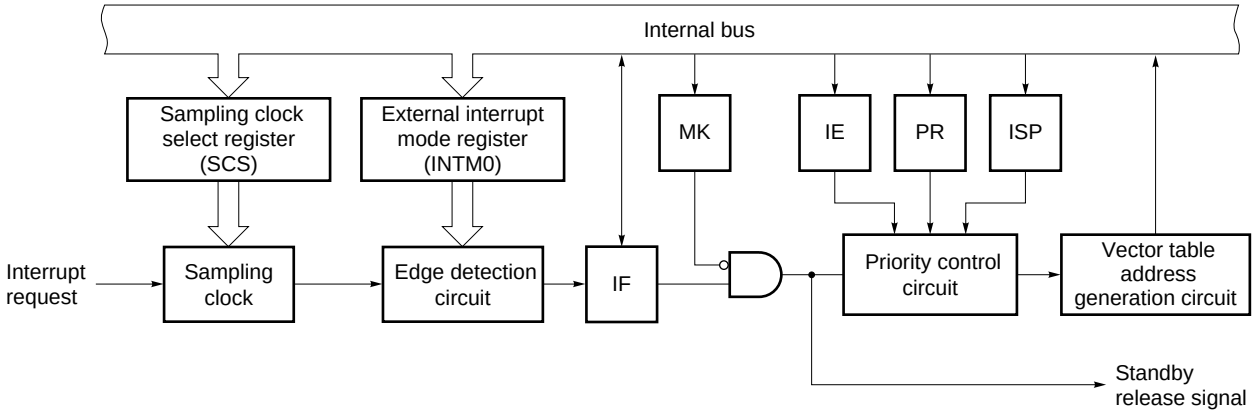
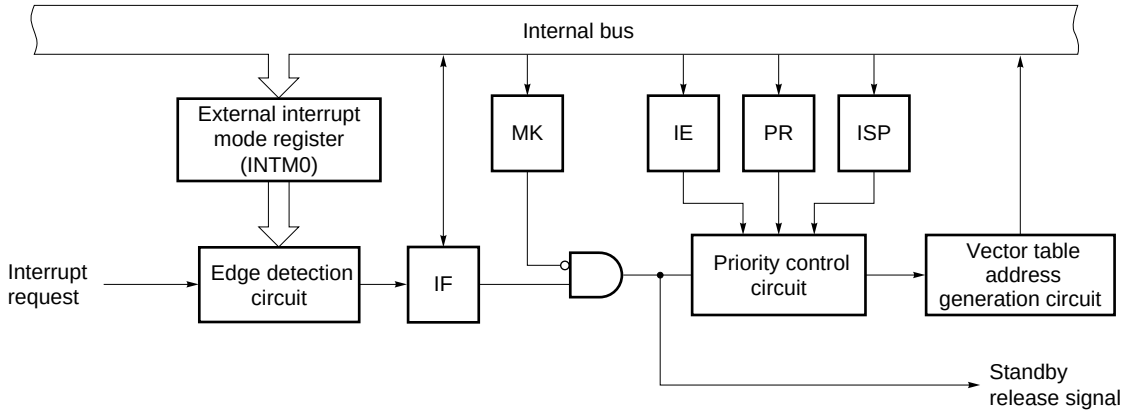
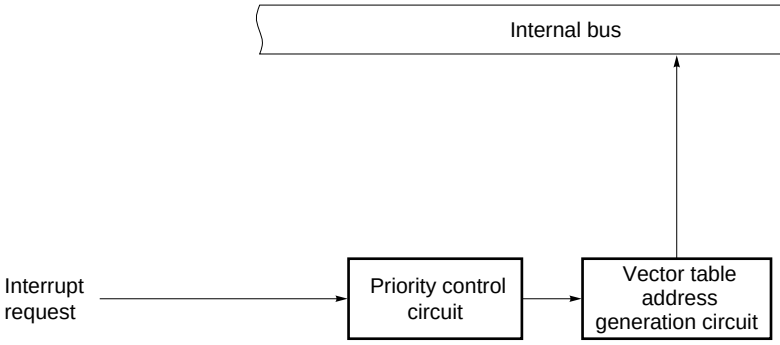


Figure 15-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



IF : interrupt request flag
IE : interrupt enable flag
ISP : in-service priority flag
MK : interrupt mask flag
PR : priority specification flag

15.3 Registers Controlling Interrupt Function

The following six types of registers control the interrupt function:

- Interrupt request flag registers (IF0L, IF0H)
- Interrupt mask flag registers (MK0L, MK0H)
- Priority specification flag registers (PR0L, PR0H)
- External interrupt mode register (INTM0)
- Sampling clock select register (SCS)
- Program status word (PSW)

Table 15-2 shows the names of the interrupt request flags, interrupt mask flags, and priority specification flags corresponding to the respective interrupt request sources.

Table 15-2. Flags Corresponding to Respective Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDT	TMIF4	IF0L	TMMK4	MK0L	TMPR4	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTCSI0	CSIIF0	IF0H	CSIMK0	MK0H	CSIPR0	PR0H
INTCSI1	CSIIF1		CSIMK1		CSIPR1	
INTTM3	TMIF3		TMMK3		TMPR3	
INTTM0	TMIF0		TMMK0		TMPR0	
INTTM1	TMIF1	IF1L	TMMK1	MK1L	TMPR1	PR1L
INTTM2	TMIF2		TMMK2		TMPR2	
INTAD	ADIF		ADMK		ADPR	

(1) Interrupt request flag registers (IF0L, IF0H)

An interrupt request flag is set to 1 when the corresponding interrupt request is generated or when an instruction is executed, and is cleared to 0 when the interrupt request is accepted, when the $\overline{\text{RESET}}$ signal is input, or when an instruction is executed.

IF0L and IF0H are set by a 1-bit or 8-bit memory manipulation instruction. When using IF0L and IF0H as a 16-bit register IF0, it is set by a 16-bit memory manipulation instruction.

These registers are set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 15-2. Format of Interrupt Request Flag Registers

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	After reset	R/W
IF0L	TMIF3	CSIIF1	CSIIF0	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W

	7	6	〈5〉	4	〈3〉	〈2〉	〈1〉	〈0〉			
IF0H	0	0	Note WTIF	0	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

××IF	Interrupt request flag
0	Interrupt request signal is not generated
1	Interrupt request signal is generated and interrupt is requested

Note The WTIF is a test input flag and does not generate a vectored interrupt request.

Cautions 1. The TMIF4 flag can be read/written only when the watchdog timer is used as an interval timer. Clear the TMIF4 flag to 0 when the watchdog timer mode 1 is used.

2. Always set bits 4, 6, and 7 of IF0H to 0.

(2) Interrupt mask flag registers (MK0L, MK0H)

An interrupt mask flag enables or disables the corresponding maskable interrupt processing and releasing the standby mode.

MK0L and MK0H are set by a 1-bit or 8-bit memory manipulation instruction. When using MK0L and MK0H as a 16-bit register MK0, it is set by a 16-bit memory manipulation instruction.

These registers are reset to FFH when the $\overline{\text{RESET}}$ signal is input.

Figure 15-3. Format of Interrupt Mask Flag Register

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	After reset	R/W
MK0L	TMMK3	CSIMK1	CSIMK0	PMK3	PMK2	PMK1	PMK0	TMMK4	FFE4H	FFH	R/W

	7	6	〈5〉	4	〈3〉	〈2〉	〈1〉	〈0〉			
MK0H	1	1	Note WTMK	1	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

××MK	Controls interrupt processing
0	Enables interrupt processing
1	Disables interrupt processing

Note The WTMK controls enabling/disabling the release of the standby mode. It does not control interrupt processing.

Cautions 1. The TMMK4 flag is undefined when it is read while the watchdog timer is used in the watchdog timer mode 1.

2. Because port 0 is shared with external interrupt request inputs, the corresponding interrupt request flag is set when the output mode is specified and output level of a port pin is changed.

To use the port in the output mode, therefore, set the corresponding interrupt mask flag to 1 in advance.

3. Always set bits 4, 6, and 7 of MK0H to 1.

(3) Priority specification flag registers (PR0L, PR0H)

A priority specification flag sets the priority of the corresponding maskable interrupt.

PR0L and PR0H are set by a 1-bit or 8-bit memory manipulation instruction. When using PR0L and PR0H as a 16-bit register PR0, it is set by a 16-bit memory manipulation instruction.

These registers are set to FFH when the $\overline{\text{RESET}}$ signal is input.

Figure 15-4. Format of Priority Specification Flag Register

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	After reset	R/W
PR0L	TMPR3	CSIPR1	CSIPR0	PPR3	PPR2	PPR1	PPR0	TMPR4	FFE8H	FFH	R/W

	7	6	5	4	〈3〉	〈2〉	〈1〉	〈0〉			
PR0H	1	1	1	1	ADPR	TMPR2	TMPR1	TMPR0	FFE9H	FFH	R/W

××PR	Selects priority level
0	High priority level
1	Low priority level

Cautions 1. Set the TMPR4 flag to 1 when using the watchdog timer in the watchdog timer mode 1.

2. Always set bits 4 to 7 of PR0H to 1.

(4) External interrupt mode register (INTM0)

This register sets the valid edges of INTP0 to INTP2.

INTM0 is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Remarks 1. The INTP0 pin is shared with TI0/P00.

2. INTP3 is fixed to the falling edge.

Figure 15-5. Format of External Interrupt Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES31	ES30	ES21	ES20	ES11	ES10	0	0	FFECH	00H	R/W

ES11	ES10	Selects valid edge of INTP0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES21	ES20	Selects valid edge of INTP1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	Selects valid edge of INTP2
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Caution Set the valid edge of the INTP0/TI0/P00 pin after setting bits 1 through 3 (TMC01 through TMC03) of the 16-bit timer mode control register (TMC0) to 0,0,0, and stopping the timer operation.

(5) Sampling clock select register (SCS)

This register sets the clock with which the valid edge input to INTP0 is sampled. When receiving a remote controller signal by using INTP0, digital noise can be eliminated by the sampling clock.

SCS is set by an 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 15-6. Format of Sampling Clock Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SCS	0	0	0	0	0	0	SCS1	SCS0	FF47H	00H	R/W

SCS1	SCS0	Selects sampling clock of INTP0
0	0	$f_X/2^{N+1}$
0	1	Setting prohibited
1	0	$f_X/2^6$ (156 kHz)
1	1	$f_X/2^7$ (78.1 kHz)

Caution $f_X/2^{N+1}$ is the clock supplied to the CPU, $f_X/2^6$ and $f_X/2^7$ are the clocks supplied to the peripheral hardware. $f_X/2^{N+1}$ is stopped in the HALT mode.

Remarks 1. N : value (N = 0 to 4) set to bits 0 to 2 (PCC0 to PCC2) of processor clock control register (PCC)

2. f_X : main system clock oscillation frequency

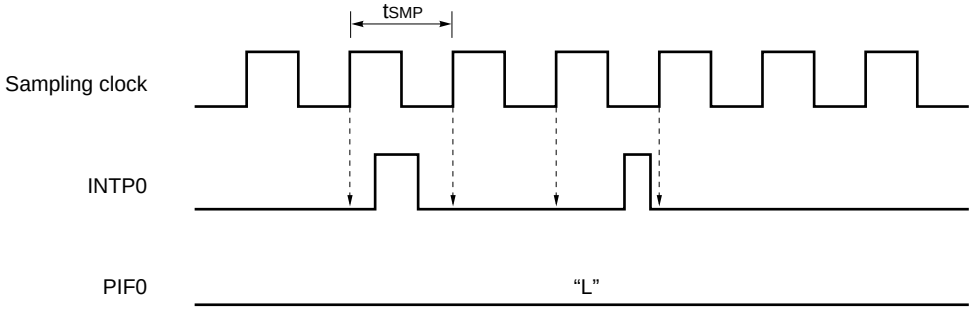
3. ( ) : at f_X = 10.0-MHz operation

The noise eliminating circuit sets the interrupt request flag (PIF0) to 1 when the input level of INTP0 sampled is active two times in succession.

Figure 15-7 shows the input/output timings of the noise eliminator.

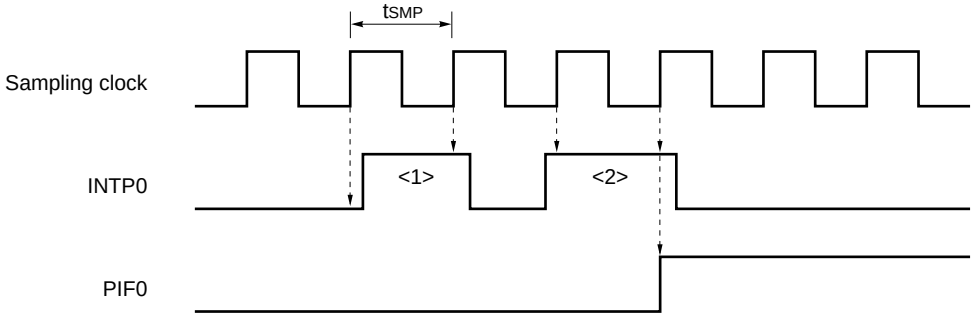
Figure 15-7. I/O Timing of Noise Eliminating Circuit (when rising edge is detected)

(a) When input is equal to sampling cycle (t_{SMP}) or lower



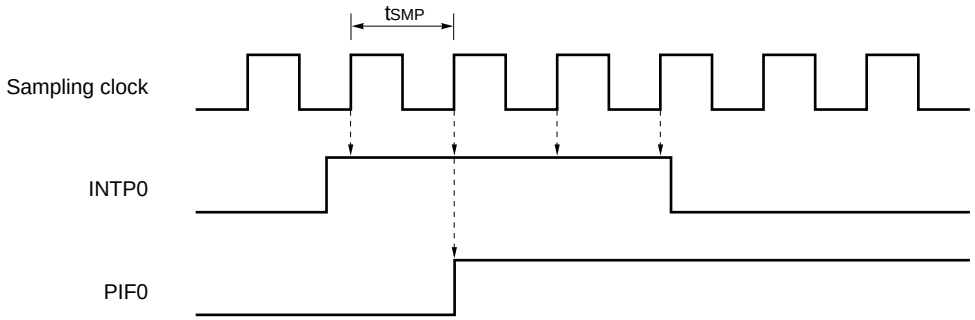
Because the level of INTP0 is not active at times of sampling, the output of PIF0 remains low.

(b) When input is 1 to 2 times the frequency of the sampling cycle (t_{SMP})



Because the level of INTP0 sampled is high two times in succession in <2>, the PIF0 flag is set to 1.

(c) When input is two or more times the frequency of the sampling cycle (t_{SMP})



In this case the level of INTP0 is high two times in succession, PIF0 flag is set to 1 at the second high-level detection.

(6) Program status word (PSW)

The program status word is a register that holds the instruction execution result and current status of interrupt request. An IE flag that enables/disables the maskable interrupts and an ISP flag that controls multiple interrupt processing are mapped to this register.

This register can be read or written in 8-bit units. In addition, it can also be manipulated by using a bit manipulation instruction or dedicated instructions (EI and DI). When a vectored interrupt request is accepted, and when the BRK instruction is executed, the PSW content is automatically saved to the stack. At this time, the IE flag is reset to 0. If a maskable interrupt request has been accepted the content of the priority flag of that interrupt is transferred to ISP flag. The content of PSW can also be saved to the stack by the PUSH PSW instruction, and restored from the stack by RETI, RETB or POP PSW instruction.

PSW is set to 02H when the $\overline{\text{RESET}}$ signal is input.

Figure 15-8. Configuration of Program Status Word

Symbol	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY	02H

→ Used when normal instruction is executed

ISP	Priority of interrupt currently processed
0	Interrupt with higher priority is processed (interrupt with lower priority is disabled)
1	Interrupt request is not accepted, or interrupt with lower priority is processed (all maskable interrupts are enabled)

IE	Enables/disables accepting interrupt requests
0	Disables
1	Enables

15.4 Interrupt Processing Operation

15.4.1 Non-maskable interrupt request acceptance operation

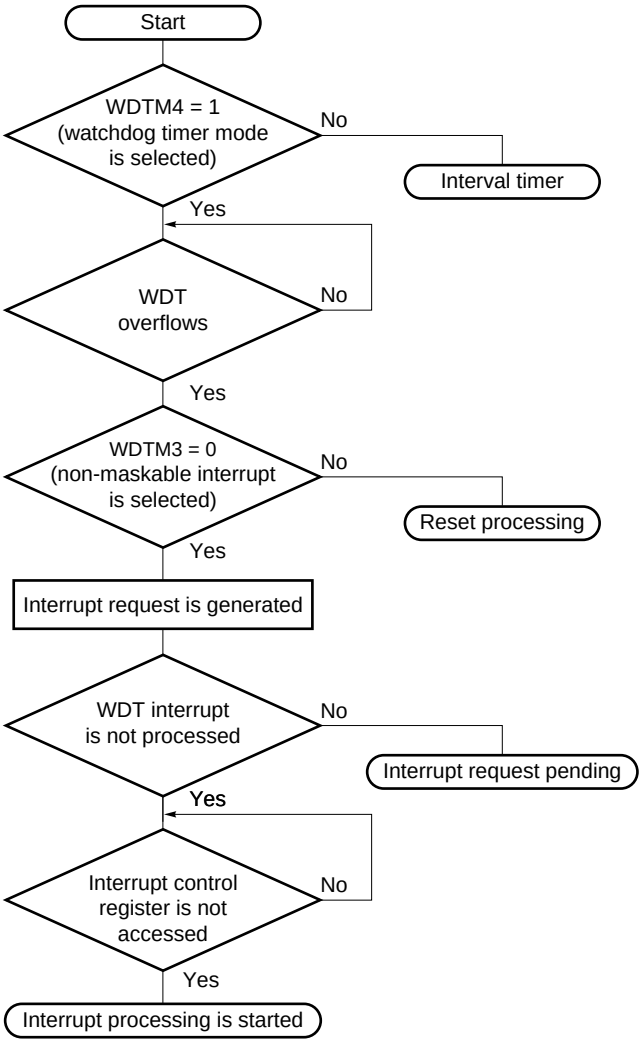
The non-maskable interrupt request is unconditionally accepted even when interrupt request input is disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the non-maskable interrupt request is acknowledged, the contents of the program status word (PSW) and program counter (PC) are saved to the stack in that order. Then, the IE flag and ISP flag are reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

If a new non-maskable interrupt request is generated while the non-maskable interrupt service program is executed, the interrupt request is accepted when the current execution of the non-maskable interrupt service program has been completed (after the RETI instruction has been executed) and one instruction in the main routine has been executed. If two or more new non-maskable interrupt requests are generated while the non-maskable interrupt service program is executed, only one non-maskable interrupt request is accepted after execution of the non-maskable interrupt service program has been completed.

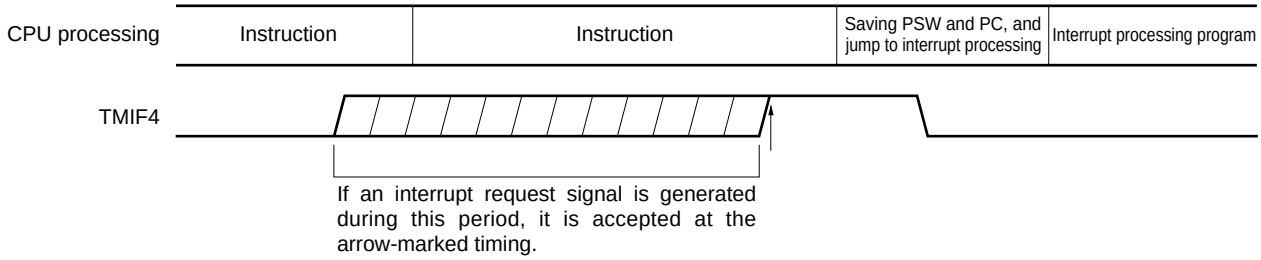
Figure 15-9 shows the flowchart from the generation of a non-maskable interrupt request signal to the acceptance of the signal. Figure 15-10 is the timing chart of the acceptance of a non-maskable interrupt request. If two or more non-maskable interrupt requests are generated at the same time, these requests are processed as illustrated in Figure 15-11.

Figure 15-9. Flowchart from Non-maskable Interrupt Request Generation to Acceptance



WDTM : watchdog timer mode register
WDT : watchdog timer

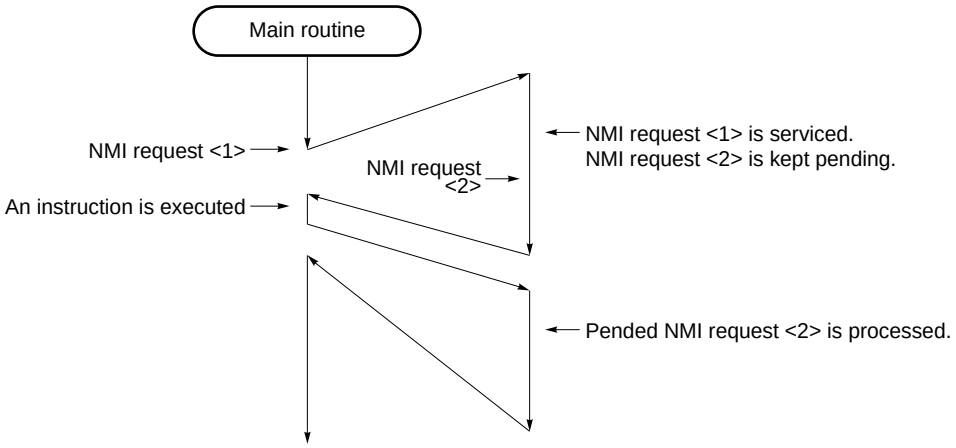
Figure 15-10. Timing of Non-maskable Interrupt Request Acceptance



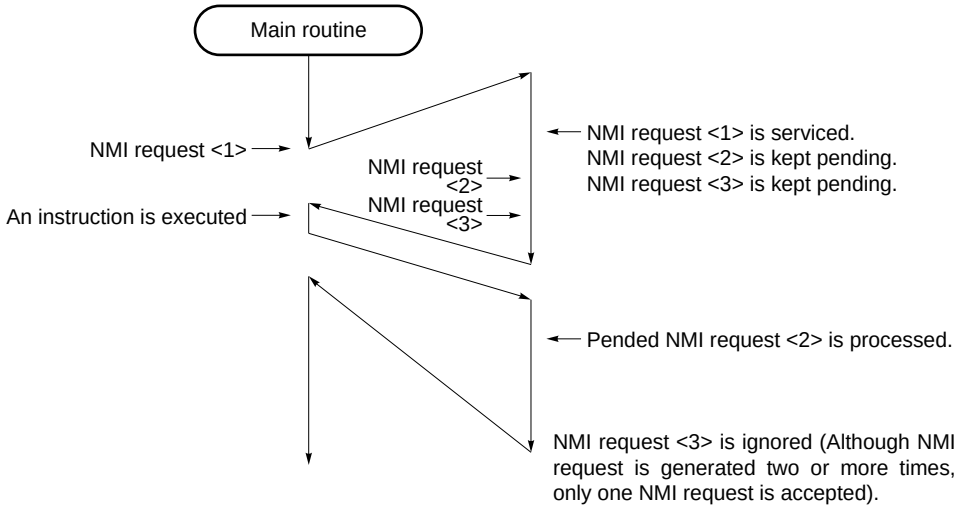
TMIF4 : watchdog timer interrupt request flag

Figure 15-11. Acceptance of Multiple Non-maskable Interrupt Requests

(a) When new non-maskable interrupt request is generated while non-maskable interrupt service program is executed



(b) If two new non-maskable interrupt requests are generated while non-maskable interrupt service program is executed



15.4.2 Maskable interrupt request acceptance operation

A maskable interrupt request can be accepted when the interrupt request flag is set to 1 and the corresponding interrupt mask flag (MK) is cleared to 0. A vectored interrupt request is accepted in the interrupt enabled status (when the IE flag is set to 1). However, an interrupt request with a lower priority cannot be accepted while an interrupt with a higher priority is being processed (when the ISP flag is reset to 0).

The time required to start the interrupt processing after a maskable interrupt request has been generated is shown in Table 15-3.

For the accepting timing of interrupt requests, refer to Figure 15-13 and 15-14.

Table 15-3. Time from Generation of Maskable Interrupt Request to Processing

	Minimum Time	Maximum Time[Note]
When xxPR = 0	13 clocks	63 clocks
When xxPR = 1	15 clocks	65 clocks

Note The wait time is maximum when an interrupt request is generated immediately before a division instruction.

Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU} : CPU clock)

When two or more maskable interrupt requests are generated at the same time, they are accepted starting from the one assigned the highest priority by the priority specification flag. When interrupts are assigned the same priority by the priority specification flag, interrupt requests given higher priority by default takes precedence.

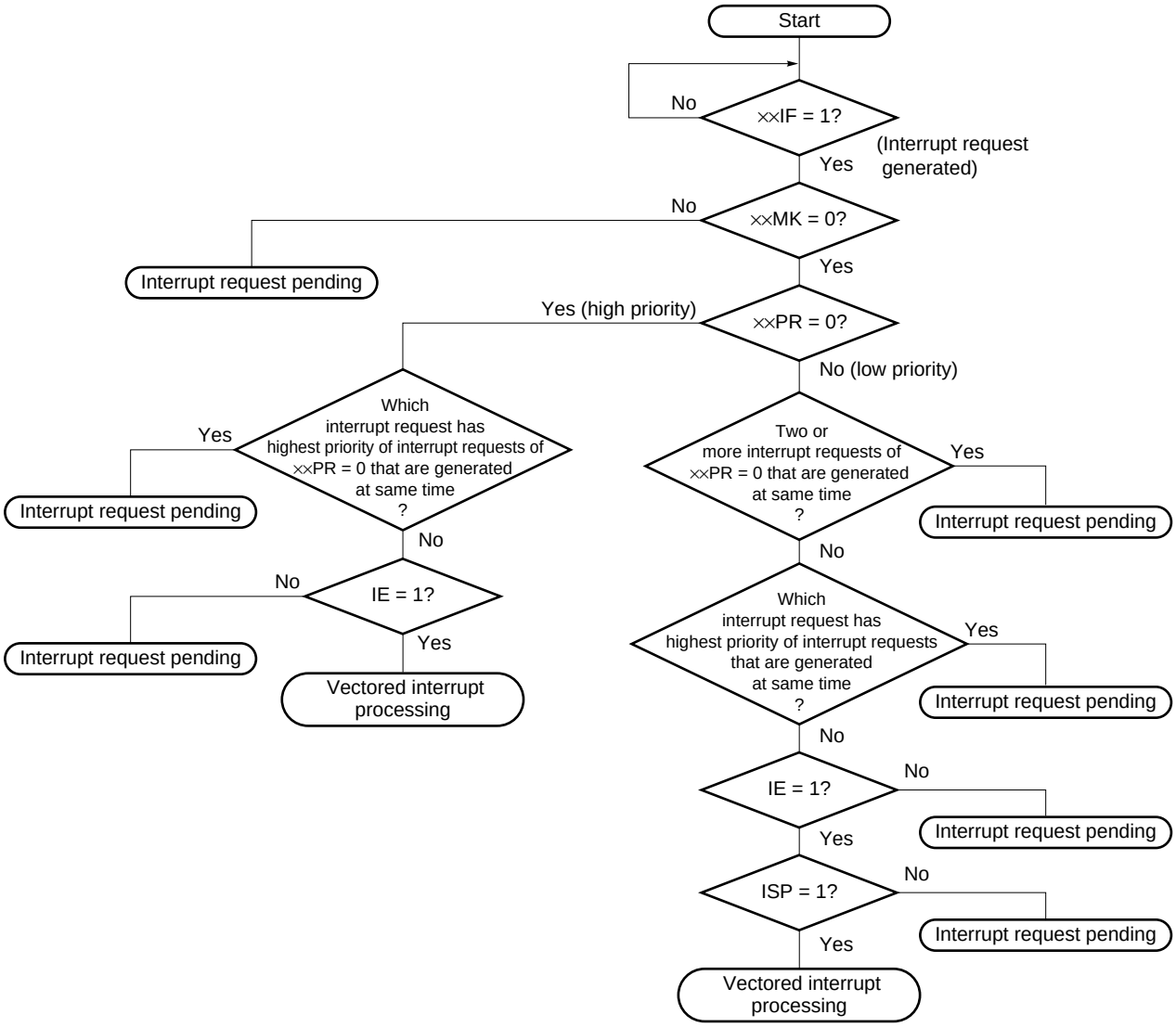
A pended interrupt request is accepted when the status where it can be accepted is set.

Figure 15-12 shows the algorithm of accepting interrupt requests.

When a maskable interrupt request is accepted, the contents of program status word (PSW) and program counter (PC) are saved to the stack in that order. Then, the IE flag is reset to 0, and the content of the interrupt request priority specification flag of the accepted interrupt is transferred to the ISP flag. In addition, the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

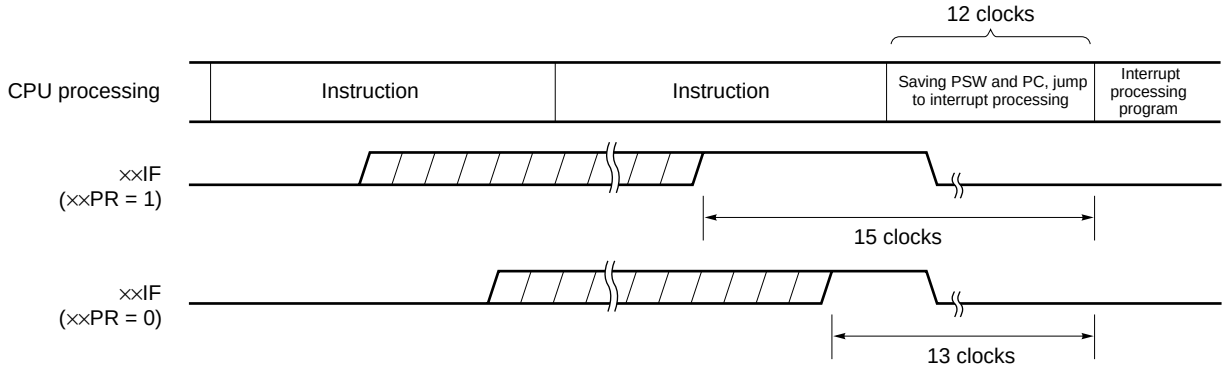
To return from interrupt processing, use the RETI instruction.

Figure 15-12. Interrupt Request Acceptance Program Algorithm



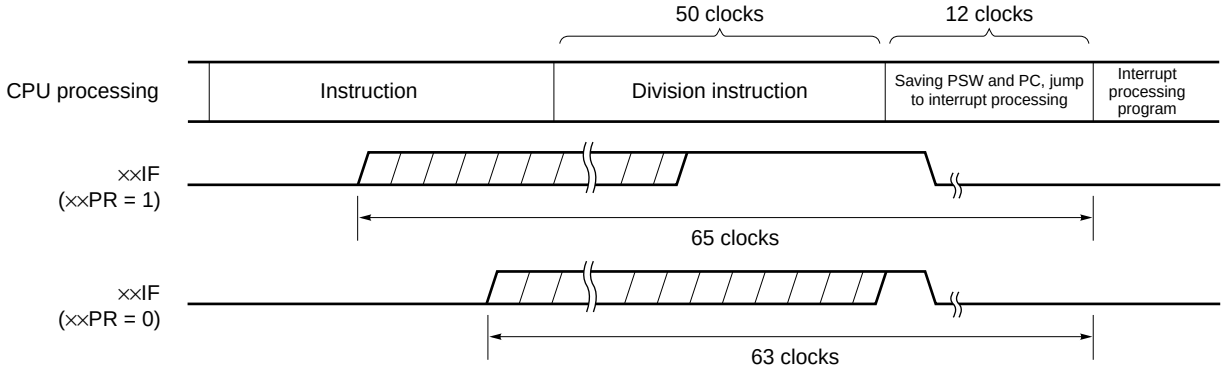
××IF : interrupt request flag
××MK : interrupt mask flag
××PR : priority specification flag
IE : flag controlling acceptance of maskable interrupt requests (1: enabled, 0: disabled)
ISP : flag indicating priority of interrupt being processed (0: in the course of processing high-priority interrupt, 1: no interrupt requests accepting or in the course of processing low-priority interrupt)

Figure 15-13. Interrupt Request Acceptance Timing (Minimum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU} : CPU clock)

Figure 15-14. Interrupt Request Acceptance Timing (Maximum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU} : CPU clock)

15.4.3 Software interrupt request acceptance operation

The software interrupt request can be accepted when the BRK instruction is executed. This interrupt request cannot be disabled.

When the software interrupt is accepted, the contents of the program status word (PSW) and program counter (PC) are saved to the stack in that order. Then, the IE flag is reset to 0, the contents of the vector table (003EH and 003FH) are loaded to the PC, and execution branches.

To return from the software interrupt processing, use the RETB instruction.

Caution Do not use the RETI instruction to return from the software interrupt.

15.4.4 Multiple interrupt processing

The status such that, in the course of an interrupt processing, the device accepts another interrupt request (starts new interrupt processing) is referred to as multiple interrupts.

Multiple interrupts occur only when the accceptance of interrupt requests is enabled (IE = 1) except for non-maskable interrupts. The IE flag is cleared to 0 (disables interrupt request accceptance) immediately after an interrupt request is accepted. Therefore, in order to enable multiple interrupt processing, an EI instruction must be executed to set the IE flag to 1, which enables interrupt accceptance.

Even if interrupt accceptance is enabled, some interrupt requests that occur during an interrupt processing may be or may not be accepted depending on their priority. There are two kinds of priority control criteria: default priority and programmable priority. Multiple interrupt control is performed by programmable priority.

The condition in which an interrupt request is accepted during an interrupt processing is that the IE flag has been set to 1 and the interrupt request has a priority higher than or the same as that of the interrupt being processed. If the priority of the interrupt request is lower than the interrupt currently being processed, it will not be accepted.

If an interrupt request is not accepted (because of disable interrupt or lower priority), it will be held pending until the current interrupt process completes, the processing goes back to the main program flow, and one instruction is executed.

Any interrupt requests will not be accepted during non-maskable interrupt processing.

Table 15-4 shows interrupt requests which can be accepted during interrupt processing; Figure 15-15 introduces some examples of multiple interrupts.

Table 15-4. Interrupt Requests the Device Accepts Even during Interrupt Processing

Interrupt being Processed \ Multiple Interrupt Requests		Non-maskable Interrupt Request	Maskable Interrupt Request			
			××PR = 0		××PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		×	×	×	×	×
Maskable interrupt	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt		○	○	×	○	×

Remarks 1. ○ : multiple interrupt processing possible

2. × : multiple interrupt processing impossible

3. ISP and IE are flags included in PSW.

ISP = 0 : interrupt with higher priority is processed.

ISP = 1 : interrupt request is not accepted or interrupt with lower priority is processed.

IE = 0 : accepting interrupt request is disabled.

IE = 1 : accepting interrupt request is enabled.

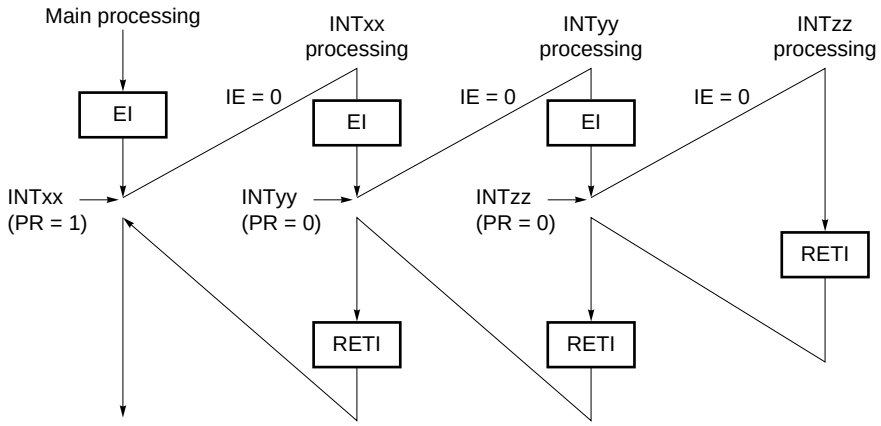
4. ××PR is flag included in PR0L, PR0H.

××PR = 0 : higher priority flag

××PR = 1 : lower priority flag

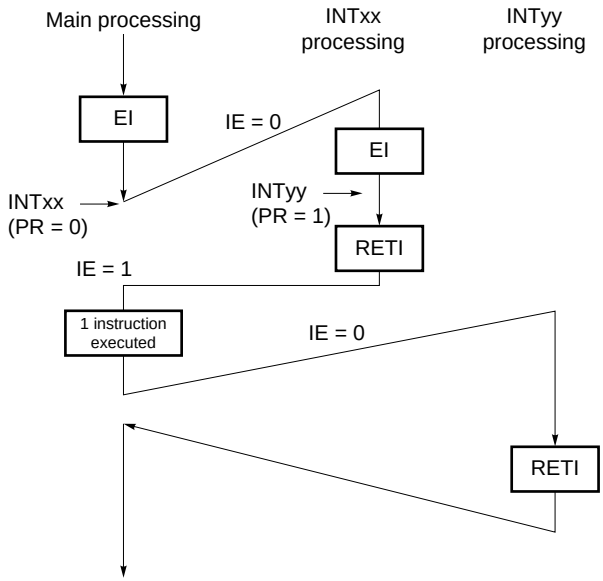
Figure 15-15. Example of Multiple Interrupt Processing (1/2)

Example 1. When another interrupt request is issued during a multiple interrupt processing



Multiple interrupts occur when the two interrupt requests INTyy and INTzz are accepted in the course of processing the interrupt INTxx. Prior to each interrupt request acceptance, an EI instruction is always executed, enabling acceptance of interrupt requests.

Example 2. Multiple interrupts do not occur based on the priority order

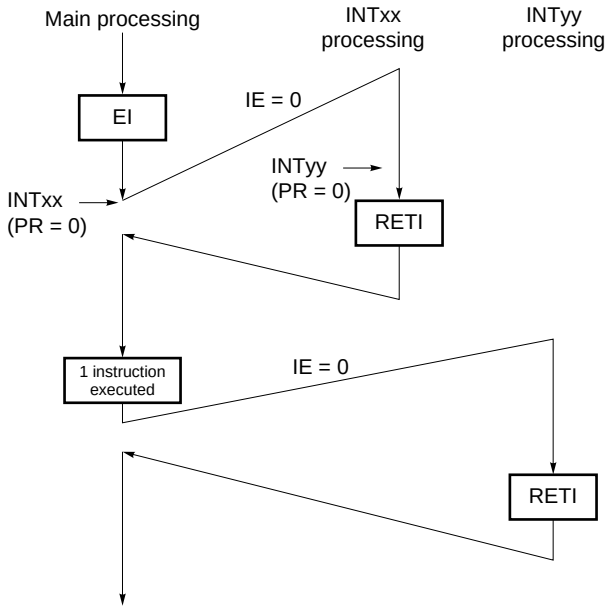


Because the interrupt request INTyy that occurs during processing of the interrupt INTxx has a lower priority than INTxx, multiple interrupts do not occur.

PR = 0 : higher priority level
PR = 1 : lower priority level
IE = 0 : interrupt request acceptance disabled

Figure 15-15. Example of Multiple Interrupt Processing (2/2)

Example 3. Because interrupt request acceptance is not enabled, multiple interrupts do not occur



Because interrupt request acceptance is not enabled (no EI instruction is executed) during the process for the interrupt INTxx, the interrupt request INTyy is not accepted. Therefore, no multiple interrupts occur. The INTyy request is held pending for a while, and is accepted after the execution of one instruction in the main program flow.

PR = 0 : higher priority
IE = 0 : interrupt request acceptance disabled

15.4.5 Pending interrupt request

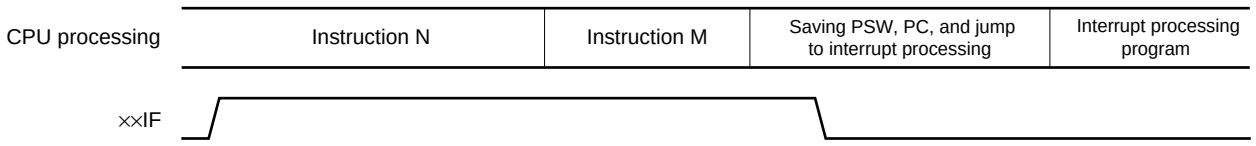
Some of the instructions keep the interrupt request generated during an interrupt processing pending until the next instruction execution is completed. These instructions (called interrupt request pending instructions) are shown below.

- 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 instruction to IF0L, IF0H, MK0L, MK0H, PR0L, PR0H, INTM0 registers

Caution The BRK instruction is not one of the interrupt request pending instructions mentioned above. However, execution of the BRK instruction triggers a software interrupt, which clears the IE flag to 0. Therefore, if a maskable interrupt request is generated during execution of the BRK instruction, the request will not be accepted due to the IE value. However, non-maskable interrupt requests will be accepted even in this situation.

Figure 15-16 shows the timing of pending interrupt request.

Figure 15-16. Pending Interrupt Request



Remarks 1. Instruction N : instruction that keeps interrupt request pending

2. Instruction M : instruction other than that which keeps interrupt request pending

3. The operation of interrupt request (××IF) is not affected by the value of priority level (××PR).

15.5 Test Functions

When an overflow occurs in the watch timer or the rising edge of port 4 is detected, the corresponding test input flag is set to 1 and the standby release signal is generated.

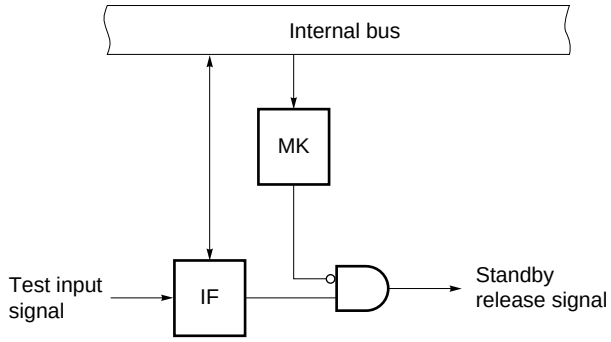
Unlike the interrupt function, vectored operation is not performed in this function.

Two test input sources are available as shown in Table 15-5. The basic configuration is as shown in Figure 15-17.

Table 15-5. Test Input Sources

Test Input Source		Internal/External
Name	Trigger	
INTWT	Overflow of watch timer	Internal
INTPT4	Detection of falling edge of port 4	External

Figure 15-17. Basic Configuration of Test Function



IF : test input flag
MK : test mask flag

15.5.1 Registers controlling test functions

The test function is controlled by the following three types of registers:

- Interrupt request flag register 0H (IF0H)
- Interrupt mask flag register 0H (MK0H)
- Key return mode register (KRM)

Table 15-6 shows the names of the test input flags and test mask flags corresponding to the respective test input signals.

Table 15-6. Flags Corresponding to Test Input Signals

Test Input Signal Name	Test Input Flag	Test Mask Flag
INTWT	WTIF	WTMK
INTPT4	KRIF	KRMK

(1) Interrupt request flag register 0H (IF0H)

This register indicates whether an overflow in the watch timer is detected or not.

IF0H is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Figure 15-18. Format of Interrupt Request Flag Register 0H

Symbol	7	6	〈5〉	4	〈3〉	〈2〉	〈1〉	〈0〉	Address	After reset	R/W
IF0H	0	0	WTIF	0	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

WTIF	Watch timer overflow detection flag
0	Not detected
1	Detected

Caution Be sure to set bits 4, 6, and 7 to 0.

(2) Interrupt mask flag register 0H (MK0H)

This register enables or disables releasing the standby mode by the watch timer.

MK0H is set by a 1-bit or 8-bit memory manipulation instruction.

This register is set to FFH when the $\overline{\text{RESET}}$ signal is input.

Figure 15-19. Format of Interrupt Mask Flag Register 0H

Symbol	7	6	〈5〉	4	〈3〉	〈2〉	〈1〉	〈0〉	Address	After reset	R/W
MK0H	1	1	WTMK	1	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

WTMK	Controls standby mode by watch timer
0	Enables releasing standby mode
1	Disables releasing standby mode

Caution Be sure to set bits 4, 6, and 7 to 1.

(3) Key return mode register (KRM)

This register enables or disables releasing the standby mode by using the key return signal (detection of the falling edge of port 4).

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

This register is set to 02H when the $\overline{\text{RESET}}$ signal is input.

Figure 15-20. Format of Key Return Mode Register

Symbol	7	6	5	4	3	2	〈1〉	〈0〉	Address	After reset	R/W
KRM	0	0	0	0	0	0	KRMK	KRIF	FFF6H	02H	R/W

KRIF	Key return signal detection flag
0	Not detected
1	Detected (detection of falling edge of port 4)

KRMK	Controls standby mode by key return signal
0	Enables releasing standby mode
1	Disables releasing standby mode

Caution Be sure to clear KRIF to 0 by program when using the falling edge detection of port 4. This bit is not automatically cleared by hardware.

15.5.2 Acceptance of test input signal

(1) Internal test input signal (INTWT)

The internal test input signal (INTWT) is generated at an overflow of the watch timer, which causes the WTIF flag to be set. At this time, if interrupt requests are not masked by the interrupt mask flag (WTMK), the standby release signal is generated. Therefore, if the WTIF flag is checked at an interval shorter than the watch timer overflow cycle, the watch function can be realized.

(2) External test input signal (INTPT4)

The external test input signal (INTPT4) is generated at the falling edge of the port 4 pins (P40 to P47), which causes the KRIF flag to be set. At this time, if the interrupt mask flag (KRMK) is not set for masking, the standby signal is generated. Therefore, based on the KRIF flag, whether key inputs exist or not can be checked by using port 4 as key return signal inputs from the key matrix.

[MEMO]

16.1 External Device Extension Function

The external device extension function is to connect an external device to areas other than the internal ROM, RAM, and SFR areas. To connect an external device, ports 4 to 6 are used. These ports control address/data, read/write strobe, wait, and address strobe signals.

Table 16-1. Pin Functions in External Memory Extension Mode

Pin Function when External Device is Connected		Alternate Function
Name	Function	
AD0 to AD7	Multiplexed address/data bus	P40 to P47
A8 to A15	Address bus	P50 to P57
$\overline{RD}$	Read strobe signal	P64
$\overline{WR}$	Write strobe signal	P65
$\overline{WAIT}$	Wait signal	P66
ASTB	Address strobe signal	P67

Table 16-2. Status of Ports 4 to 6 in External Memory Extension Mode

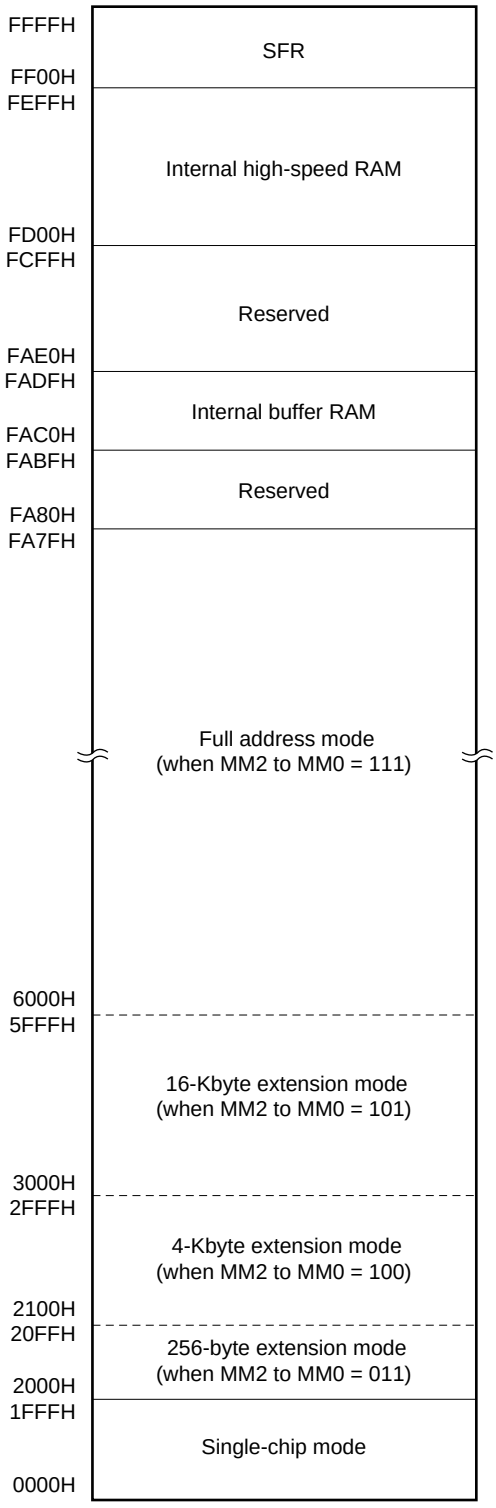
Port / External Extension Mode	Port 4	Port 5								Port 6							
	0 to 7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Single-chip mode	Port	Port								Port							
256-byte extension mode	Address/data	Port								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
4-Kbyte extension mode	Address/data	Address				Port				Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
16-Kbyte extension mode	Address/data	Address						Port		Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			
Full address mode	Address/data	Address								Port				$\overline{RD}$, $\overline{WR}$, $\overline{WAIT}$, ASTB			

Caution When the external wait function is not used, the $\overline{WAIT}$ pin can be used as a port pin in all the modes.

The memory map is as follows when the external device extension function is used.

Figure 16-1. Memory Map when External Device Extension Function Is Used (1/2)

(a) Memory map of μPD78011H



(b) Memory map of μPD78012H

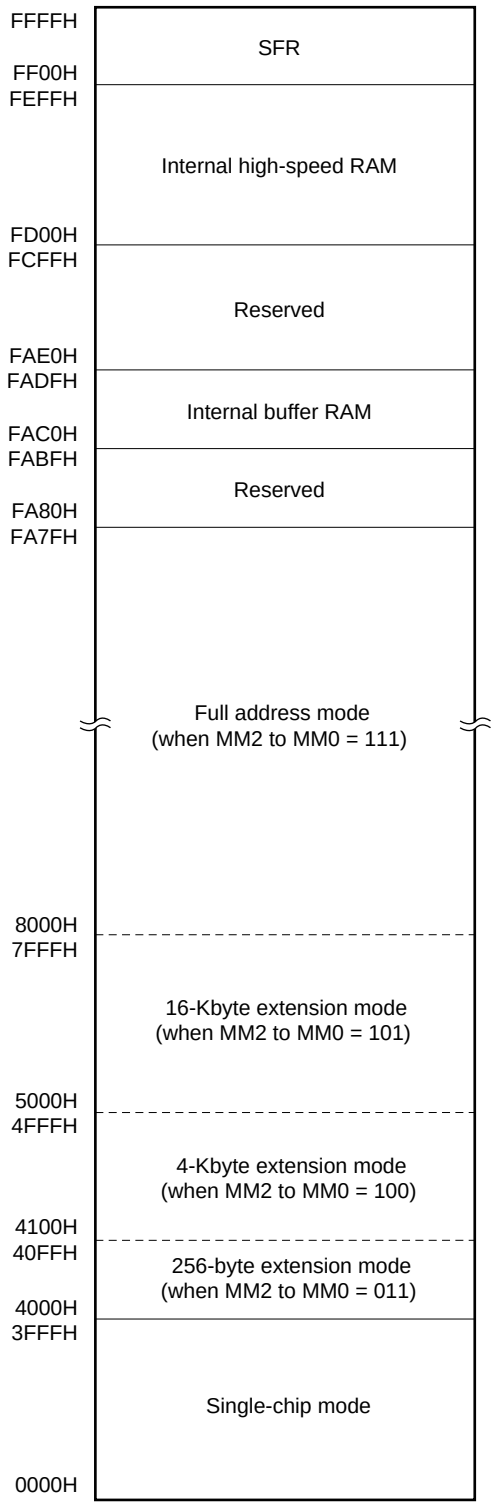
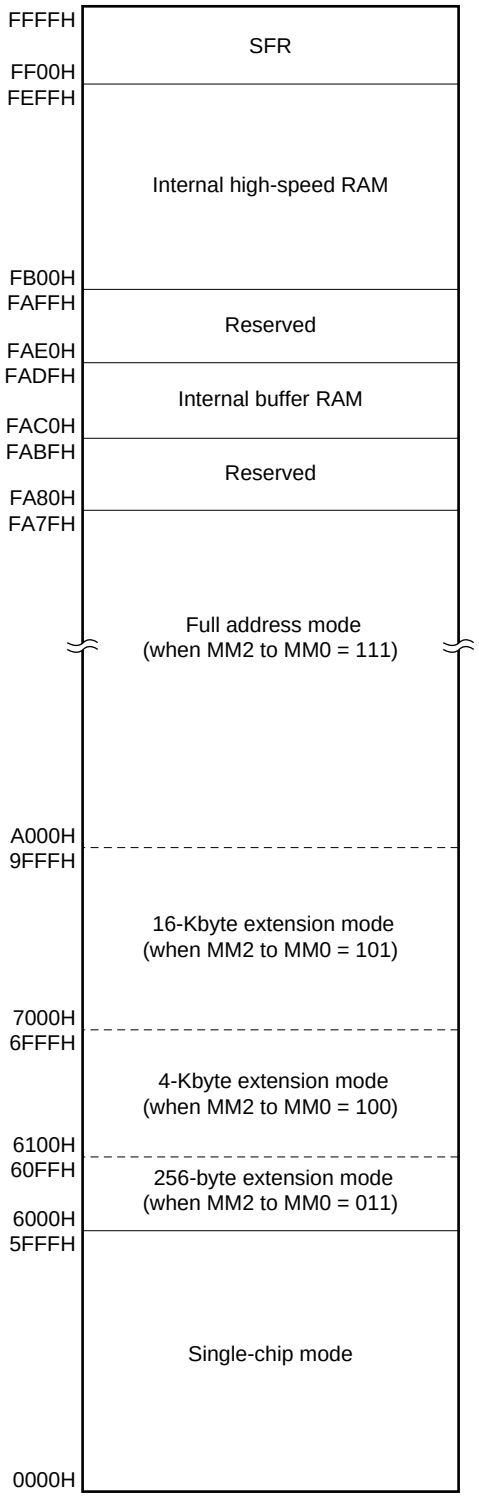
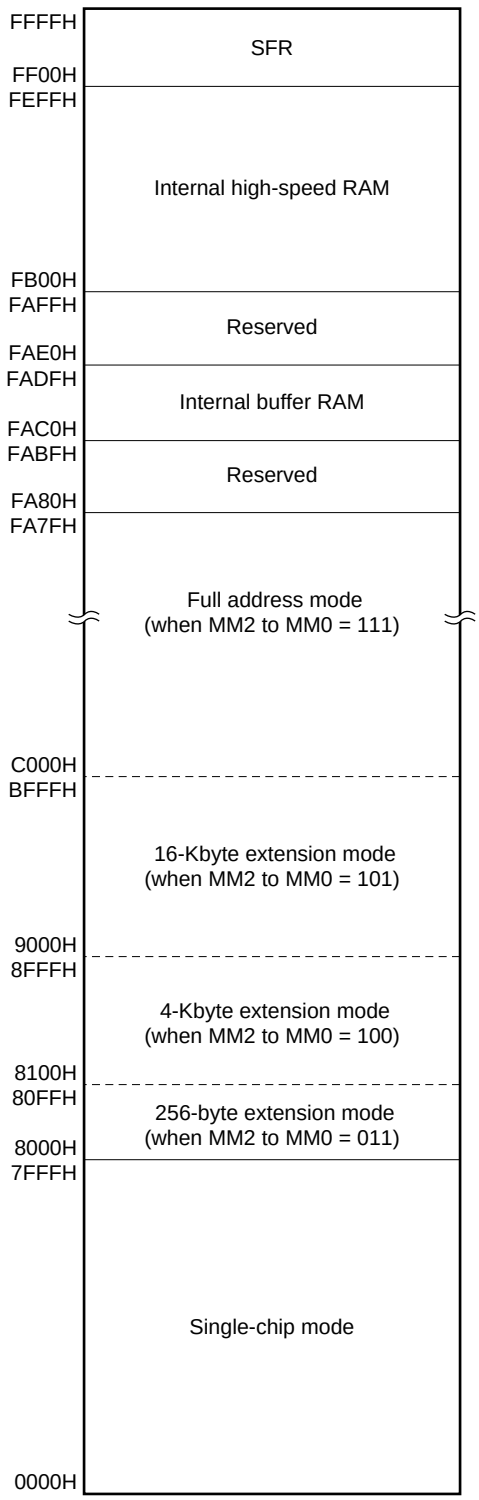


Figure 16-1. Memory Map when External Device Extension Function Is Used (2/2)

(c) Memory map of μPD78013H



(d) Memory map of μPD78014H



16.2 Registers Controlling External Device Extension Function

The external device expansion function is controlled by the memory expansion mode register (MM) and memory size select register (IMS).

(1) Memory expansion mode register (MM)

MM is a register that sets the number of wait states and an external expansion area It also sets the input or output mode of port 4.

MM is set by using an 8-bit memory manipulation instruction.

Its value is set to 10H at $\overline{\text{RESET}}$.

Figure 16-2. Format of Memory Extension Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
MM	0	0	PW1	PW0	0	MM2	MM1	MM0	FFF8H	10H	R/W

MM2	MM1	MM0	Selects single-chip/ memory extension mode		Status of P40 to P47, P50 to P57, P64 to P67 pins				
					P40 to P47	P50 to P53	P54, P55	P56, P57	P64 to P67
0	0	0	Single-chip mode		Port mode: Input	Port mode			
0	0	1			Port mode: Output				
0	1	1	Memory extension mode	256-byte mode	AD0 to AD7	Port mode			P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = $\overline{\text{WAIT}}$ P67 = ASTB
1	0	0		4-Kbyte mode		A8 to A11	Port mode		
1	0	1		16-Kbyte mode			A12, A13	Port mode	
1	1	1		Full address mode[Note]				A14, A15	
Others			Setting prohibited						

PW1	PW0	Controls wait state
0	0	No wait
0	1	Wait (1 wait state is inserted)
1	0	Setting prohibited
1	1	Wait control by external wait pin

Note The full address mode is a mode in which the entire area of the 64-Kbyte address space, except the internal ROM, RAM, SFR, and unused areas, can be externally extended.

Remark The P60 to P63 pins can be used in the port mode, regardless of the single-chip mode and memory extension mode.

(2) Memory size select register (IMS)

This register sets the capacities of the internal ROM and internal high-speed RAM. Set IMS to the value at reset.

IMS is set by using an 8-bit memory manipulation instruction.

The value of this register is as shown in Table 16-3 at $\overline{\text{RESET}}$.

Figure 16-3. Format of Memory Size Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	**Note**	W

ROM3	ROM2	ROM1	ROM0	Selects internal ROM capacity
0	0	1	0	8 Kbytes
0	1	0	0	16 Kbytes
0	1	1	0	24 Kbytes
1	0	0	0	32 Kbytes
Others				Setting prohibited

RAM2	RAM1	RAM0	Selects internal high-speed RAM capacity
0	1	0	512 bytes
1	1	0	1024 bytes
Others			Setting prohibited

Note The value of this register at reset differs depending on the model (refer to **Table 16-3**).

Table 16-3. Value of Memory Size Select Register on Reset

Part Number	Set Value of IMS
μPD78011H	42H
μPD78012H	44H
μPD78013H	C6H
μPD78014H	C8H

16.3 Timing of External Device Extension Function

The timing control signal output pins used in the external memory extension mode are as follows:

(1) $\overline{\text{RD}}$ pin (shared by P64)

This pin outputs a read strobe signal when an instruction is fetched or data is accessed from the external memory.

When the internal memory is accessed, the read strobe signal is not output (instead, this pin holds the high level).

(2) $\overline{\text{WR}}$ pin (shared by P65)

This pin outputs a write strobe signal when the external memory is accessed for data.

When the internal memory is accessed, the write strobe signal is not output (this pin holds the high level).

(3) $\overline{\text{WAIT}}$ pin (shared by P66)

This pin inputs an external wait signal.

When the external wait signal is not used, the $\overline{\text{WAIT}}$ pin can be used as an I/O port pin.

When the internal memory is accessed, the external wait signal is ignored.

(4) ASTB pin (shared by P67)

This pin outputs an address strobe signal which is always output regardless of instruction fetch or data access from the external memory.

(the address strobe signal is also output when the internal memory is accessed)

(5) AD0 to AD7, A8 to A15 pins (shared by P40 to P47, P50 to P57)

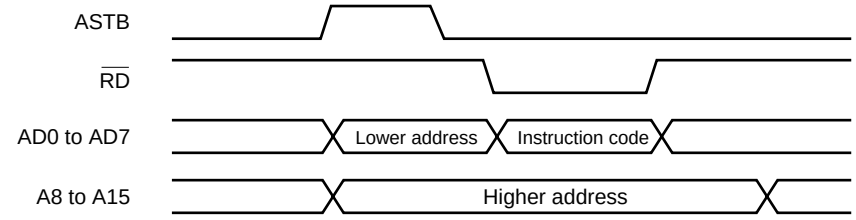
These pins output address and data signals. The valid signals are output or input when instructions are fetched or data is accessed from the external memory. The status of the signal also changes when the internal memory is accessed.

(the output contents are undefined)

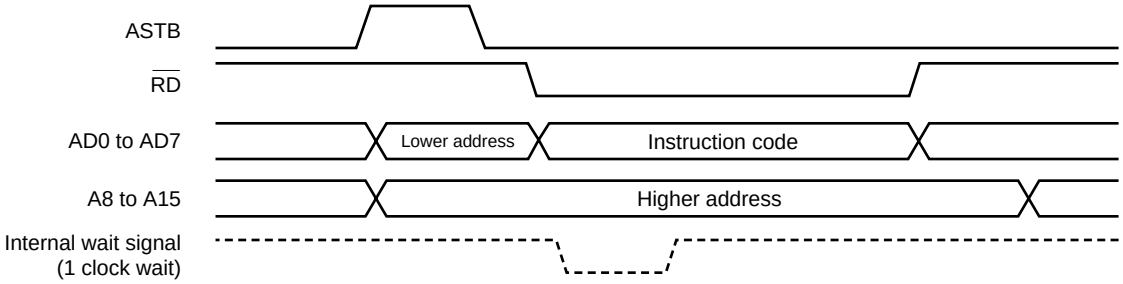
Figures 16-4 through 16-7 show the timing charts.

Figure 16-4. Instruction Fetch from External Memory

(a) When no wait state is set (PW1, PW0 = 0, 0)



(b) When wait state is set (PW1, PW0 = 0, 1)



(c) When external wait state is set (PW1, PW0 = 1, 1)

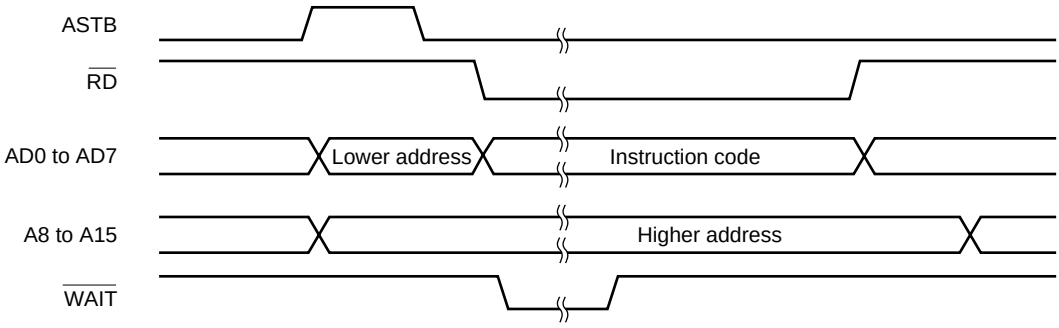
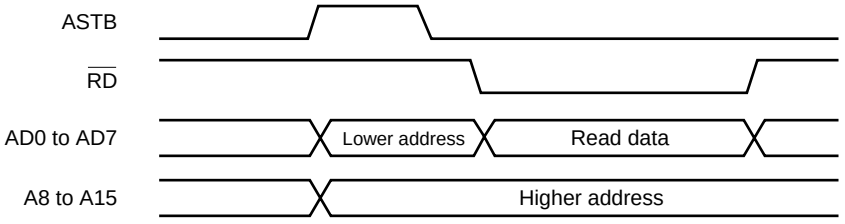
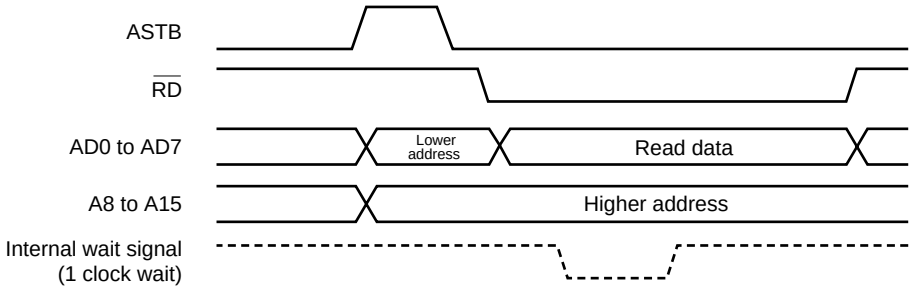


Figure 16-5. Read Timing of External Memory

(a) When no wait state is set (PW1, PW0 = 0, 0)



(b) When wait state is set (PW1, PW0 = 0, 1)



(c) When external wait state is set (PW1, PW0 = 1, 1)

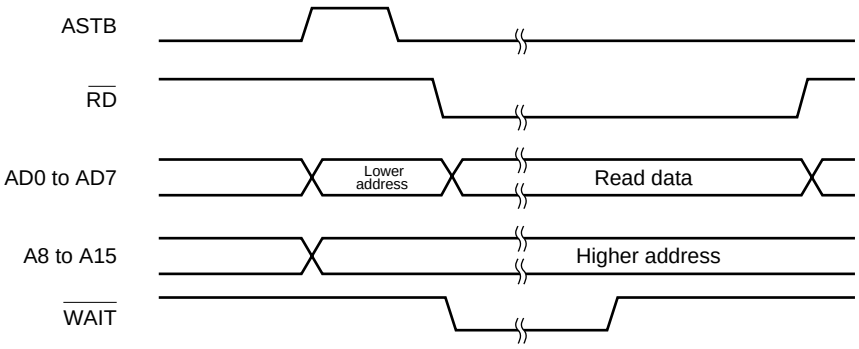
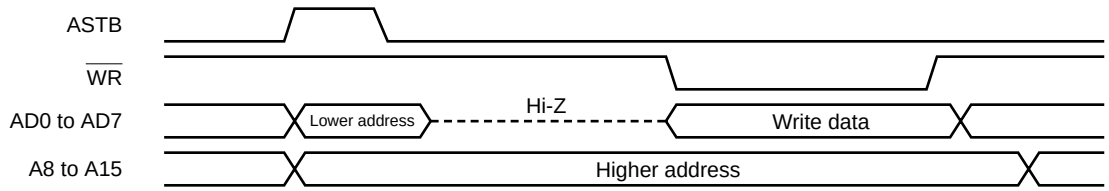
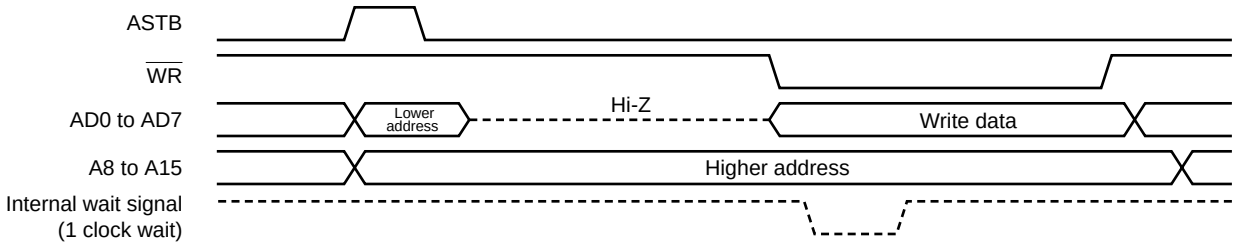


Figure 16-6. Write Timing of External Memory

(a) When no wait state is set (PW1, PW0 = 0, 0)



(b) When wait state is set (PW1, PW0 = 0, 1)



(c) When external wait state is set (PW1, PW0 = 1, 1)

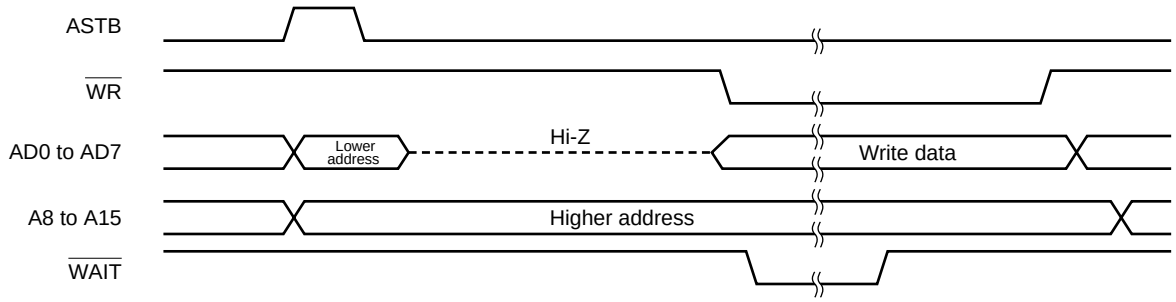
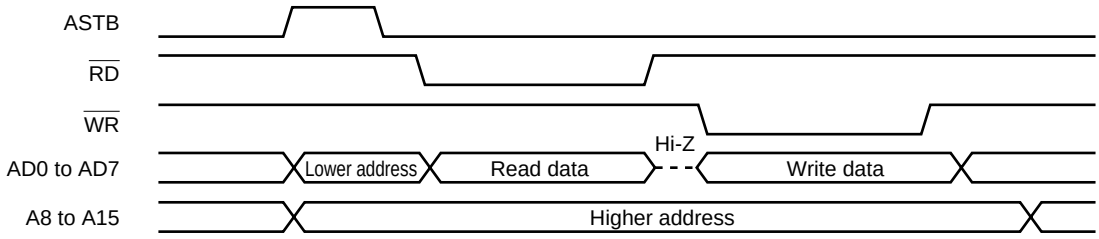
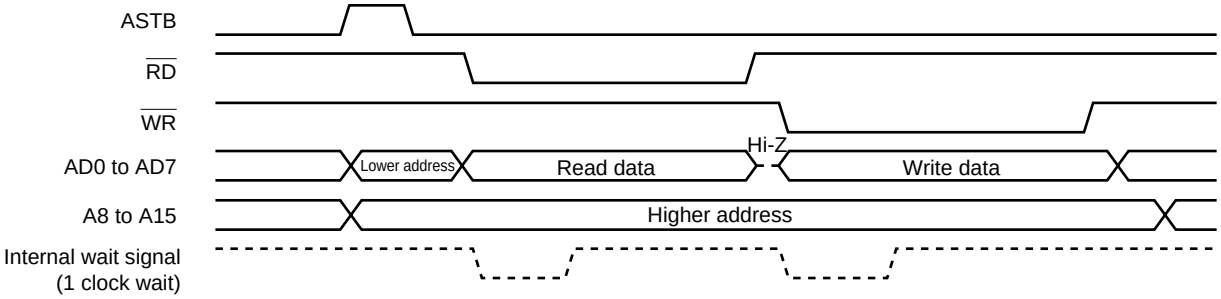


Figure 16-7. Read-Modify-Write Timing of External Memory

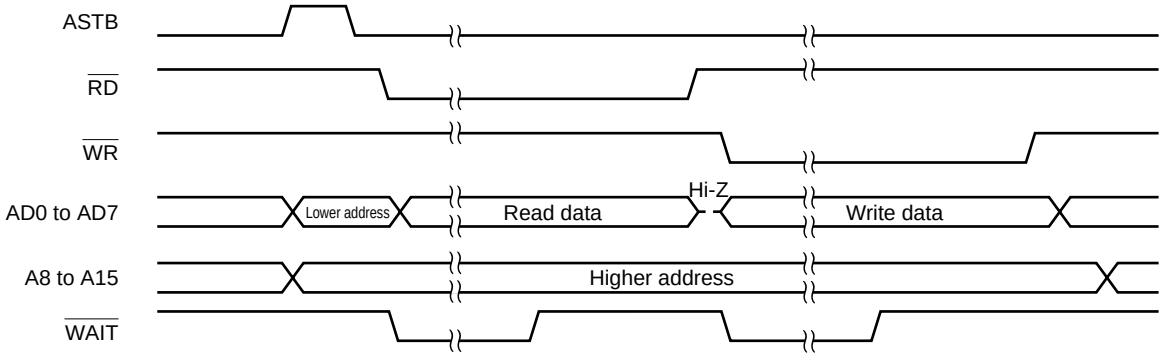
(a) When no wait state is set (PW1, PW0 = 0, 0)



(b) When wait state is set (PW1, PW0 = 0, 1)



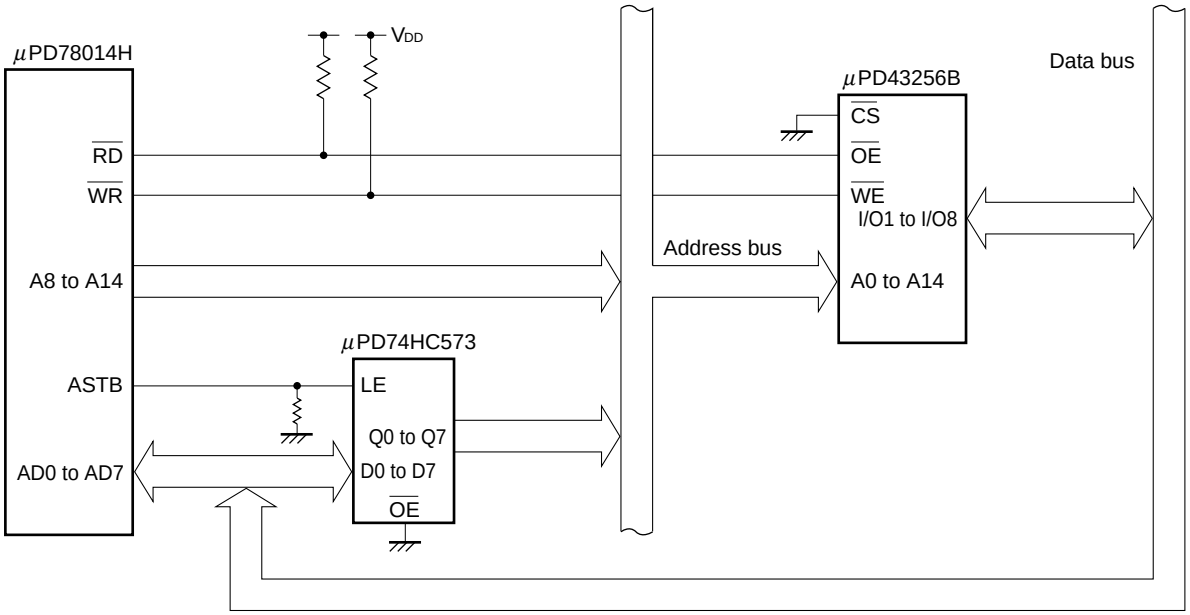
(c) When external wait state is set (PW1, PW0 = 1, 1)



16.4 Example of Connection with Memory

Figure 16-8 shows an example of connecting the μPD78014H and external memories. In this application example, SRAM is connected. In addition, the external device expansion function is used in the full address mode, and 32 Kbytes of addresses, 0000H through 7FFFH, are allocated to internal ROM; addresses 8000H and higher are allocated to SRAM.

Figure 16-8. Example of Connecting μPD78014H and Memories



[MEMO]

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is to reduce the power dissipation of the system and can be effected in the following two modes:

(1) HALT mode

This mode is set when the HALT instruction is executed. The HALT mode stops the operation clock of the CPU. The system clock oscillation circuit continues oscillating. This mode does not reduce the power dissipation as much as the STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations such as a watch operation.

(2) STOP mode

This mode is set when the STOP instruction is executed. The STOP mode stops the main system clock oscillation circuit and stops the entire system. The power dissipation of the CPU can be substantially reduced in this mode.

The low voltage (V_{DD} = 1.8 V) of the data memory can be retained. Therefore, this mode is useful for retaining the contents of the data memory at an extremely low current.

The STOP mode can be released by an interrupt request, so that this mode can be used for the intermittent operation. However, certain time is required until the system clock oscillation circuit stabilizes after the STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting the standby mode are all retained. In addition, the statuses of the output latch of the I/O ports and output buffer are also retained.

Cautions 1. The STOP mode can be used only when the system operates on the main system clock (this mode cannot be used to stop the oscillation of the subsystem clock). The HALT mode can be used regardless of whether the system operates on the main system clock or subsystem clock.

2. To set the STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

3. To reduce the power dissipation of the A/D converter, clear bit 7 (CS) of A/D converter mode register (ADM) to 0 to stop the A/D conversion, and then execute the HALT or STOP instruction.

17.1.2 Registers controlling standby function

The wait time during which oscillation is stabilized after the STOP mode has been released by an interrupt request is controlled by the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

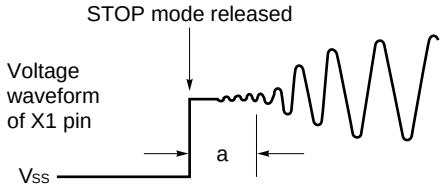
This register is set to 04H when the $\overline{\text{RESET}}$ signal is input. Therefore, to release the STOP mode by inputting the $\overline{\text{RESET}}$ signal, the time required to release the mode is $2^{18}/f_X$.

Figure 17-1. Format of Oscillation Stabilization Time Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selects oscillation stabilization time when released the STOP mode
0	0	0	$2^{13}/f_X$ (819 μs)
0	0	1	$2^{15}/f_X$ (3.28 ms)
0	1	0	$2^{16}/f_X$ (6.55 ms)
0	1	1	$2^{17}/f_X$ (13.1 ms)
1	0	0	$2^{18}/f_X$ (26.2 ms)
Others			Setting prohibited

Caution The wait time when the STOP mode is released does not include the time required for the clock oscillation to start after the STOP mode has been released (see "a" in the figure below), regardless of whether the mode has been released by the $\overline{\text{RESET}}$ signal or an interrupt request.



Remarks 1. f_X : main system clock oscillation frequency

2. ( ) : at f_X = 10.0-MHz operation

17.2 Operation of Standby Function

17.2.1 HALT mode

(1) Setting and operation status of HALT mode

The HALT mode is set by executing the HALT instruction. This mode can be set regardless of whether the system has been operating on the main system clock or subsystem clock.

The operation status in the HALT mode is shown in the table below.

Table 17-1. Operation Status in HALT Mode (1/2)

(a) When HALT instruction is executed while system operates on main system clock

Item \ Setting of HALT Mode		Without Subsystem Clock[Note 1]	With Subsystem Clock[Note 2]
Clock generation circuit		Both main system clock and subsystem clock can oscillate. Supply of clock to CPU is stopped.	
CPU		Stops operation	
Port (output latch)		Retains previous status before setting HALT mode	
16-bit timer/event counter		Operable	
8-bit timer/event counter			
Watchdog timer			
A/D converter			
Watch timer		Operable when $f_X/2^8$ selected as count clock	Operable
Serial interface	Other than automatic transmission/reception function	Operable	
	Automatic transmission/ reception function	Stops operation	
External interrupt	INTP0	Operable when clock to peripheral hardware ($f_X/2^6$, $f_X/2^7$) selected as sampling clock	
	INTP1 to INTP3	Operable	
Externally extended bus line	AD0 to AD7	High impedance	
	A8 to A15	Retains previous status before setting HALT mode	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

Notes 1. Includes the case where an external clock is not supplied as the subsystem clock.

2. Includes the case where an external clock is supplied as the subsystem clock.

Table 17-1. Operation Status in HALT Mode (2/2)

(b) When HALT instruction is executed while system operates on subsystem clock

Item \ Setting of HALT Mode		When Main System Clock Oscillation Continues	When Main System Clock Oscillation Stops
Clock generation circuit		Both main system clock and subsystem clock can oscillate. Supply of clock to CPU is stopped.	
CPU		Stops operation	
Port (output latch)		Retains previous status before setting HALT mode	
16-bit timer/event counter		Operable	Stops operation
8-bit timer/event counter			Operable when TI1 or TI2 selected as count clock
Watchdog timer		Stops operation	
A/D converter		Operable	Stops operation
Watch timer			Operable when f_{XT} selected as count clock
Serial interface	Other than automatic transmission/reception function	Operable	Operable when external clock selected
	Automatic transmission/ reception function	Stops operation	
External interrupt	INTP0	Operable when clock to peripheral hardware ($f_X/2^6$, $f_X/2^7$) selected as sampling clock	Stops operation
	INTP1 to INTP3	Operable	
Externally extended bus line	AD0 to AD7	High impedance	
	A8 to A15	Retains previous status before setting HALT mode	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

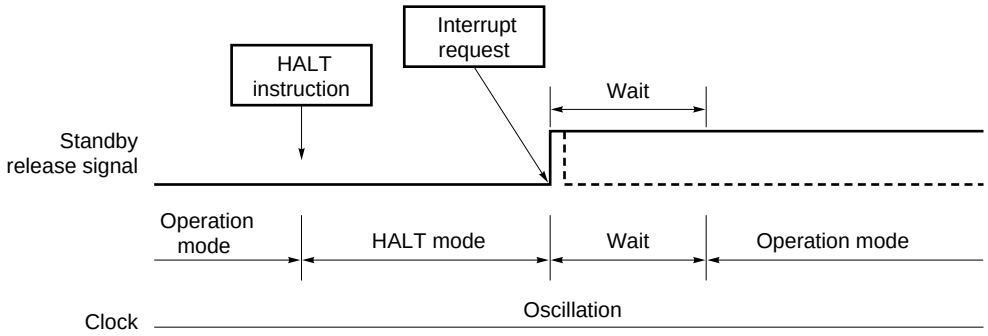
(2) Releasing HALT mode

The HALT mode can be released by the following four types of sources:

(a) Releasing by unmasked interrupt request

The HALT mode is released by the generation of an unmasked interrupt request. In this case, if the interrupt request is enabled to be accepted, vectored interrupt processing is performed. If the interrupt request is disabled, the instruction at the next address is executed.

Figure 17-2. Releasing HALT Mode by Interrupt Request



Remarks 1. The broken line indicates the case where the interrupt request that has released the standby mode is accepted.

2. The wait time is as follows:

- When vectored interrupt processing is performed : 16.5 to 17.5 clocks
- When vectored interrupt processing is not performed : 4.5 to 5.5 clocks

(b) Releasing by non-maskable interrupt request

When a non-maskable interrupt request is generated, the HALT mode is released regardless of whether the interrupt request is enabled or disabled, and vectored interrupt processing is performed.

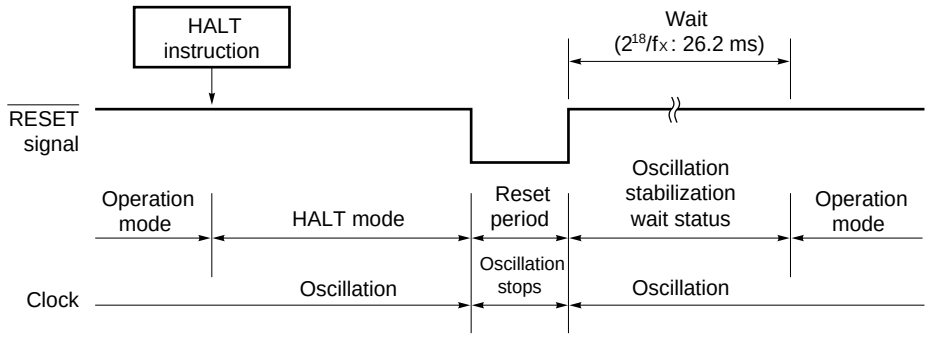
(c) Releasing by unmasked test input

When the HALT mode has been released by an unmasked test signal input, the instruction at the address next to that of the HALT instruction is executed.

(d) Releasing by $\overline{\text{RESET}}$ input

The HALT mode is released by $\overline{\text{RESET}}$ signal input. Then execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 17-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remarks 1. f_X : main system clock oscillation frequency

2. ( ): at f_X = 10.0-MHz operation

Table 17-2. Operation after Release of HALT Mode

Releasing Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Executes next address instruction
	0	0	1	×	Executes interrupt processing
	0	1	0	1	Executes next address instruction
	0	1	×	0	
	0	1	1	1	Executes interrupt processing
	1	×	×	×	Retains HALT mode
Non-maskable interrupt request	–	–	×	×	Executes interrupt processing
Test input	0	–	×	×	Executes next address instruction
	1	–	×	×	Retains HALT mode
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark × : don't care

17.2.2 STOP mode

(1) Setting and operation status of STOP mode

The STOP mode is set by executing the STOP instruction. This mode can be set only when the system operates on the main system clock.

Cautions 1. When the STOP mode is set, X2 pin is internally pulled up circuited to V_{DD} to suppress the current leakage of the crystal oscillation circuit block. Therefore, do not use the STOP mode in a system where the external clock is used as the main system clock.

2. Because the standby mode can be released by an interrupt request signal, the standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. When the STOP mode is set, therefore, the HALT mode is set immediately after the STOP instruction has been executed, the wait times set by the oscillation stabilization time select register (OSTS) elapses, and then an operation mode is set.

The following table shows the operation status in the STOP mode.

Table 17-3. Operation Status in STOP Mode

Item \ Setting of STOP Mode		When Subsystem Clock is Used	When Subsystem Clock is Not Used
Clock generation circuit		Only main system clock stops oscillation	
CPU		Stops operation	
Output port (output latch)		Retains previous status immediately before STOP instruction execution	
16-bit timer/event counter		Stops operation	
8-bit timer/event counter		Operable only when TI1 or TI2 is selected as count clock	
Watchdog timer		Stops operation	
A/D converter			
Watch timer		Operable only when f_{XT} is selected as count clock	Stops operation
Serial interface	Other than automatic transmission/ reception function	Operable only when external input clock is selected as serial clock	
	Automatic transmission/ reception function	Stops operation	
External interrupt	INTP0	Cannot operate	
	INTP1 to INTP3	Operable	
Externally extended bus line	AD0 to AD7	High impedance	
	A8 to A15	Retains previous status immediately before STOP instruction execution	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

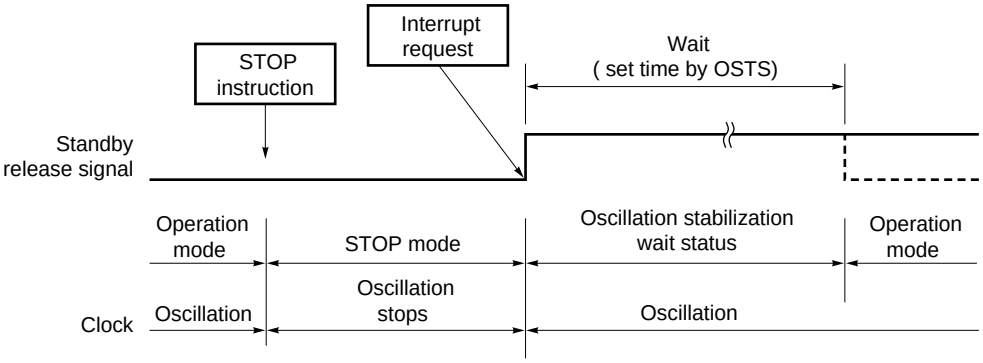
(2) Releasing STOP mode

The STOP mode can be released by the following three types of sources:

(a) Releasing by unmasked interrupt request

The STOP mode can be released by the generation of an unmasked interrupt request. In this case, if the interrupt request is enabled to be accepted, vectored interrupt processing is performed, after the oscillation stabilization time has elapsed. If the interrupt request is disabled to be accepted, the instruction at the next address is executed.

Figure 17-4. Releasing STOP Mode by Interrupt Request



Remark The broken line indicates the case where the interrupt request that has released the standby mode is accepted.

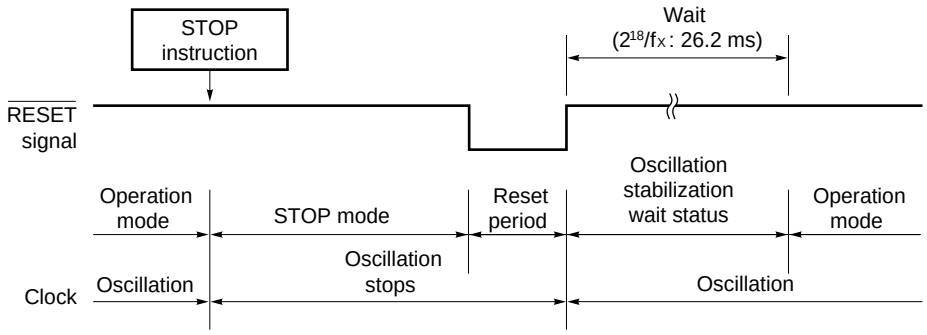
(b) Releasing by unmasked test input

When the STOP mode has been released by an unmasked test signal input, the instruction at the address next to that of the STOP instruction is executed, after the oscillation stabilization time has elapsed.

(c) Releasing by $\overline{\text{RESET}}$ input

The STOP mode is released by $\overline{\text{RESET}}$ signal input. Then the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 17-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx : main system clock oscillation frequency

2. ( ): at fx = 10.0-MHz operation

Table 17-4. Operation after Release of STOP Mode

Releasing Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Executes next address instruction
	0	0	1	×	Executes interrupt processing
	0	1	0	1	Executes next address instruction
	0	1	×	0	
	0	1	1	1	Executes interrupt processing
	1	×	×	×	Retains STOP mode
Test input	0	–	×	×	Executes next address instruction
	1	–	×	×	Retains STOP mode
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark × : don't care

CHAPTER 18 RESET FUNCTION

18.1 Reset Function

The reset signal can be effected by the following two methods:

(1) External reset input from $\overline{\text{RESET}}$ pin
(2) Internal reset by runaway time detection by watchdog timer

There is no functional difference between the external reset and internal reset, and execution of the program is started from addresses written to addresses 0000H and 0001H when the $\overline{\text{RESET}}$ signal is input.

The reset function is effected when a low-level signal is input to the $\overline{\text{RESET}}$ pin or when an overflow occurs in the watchdog timer. As a result, each hardware enters the status shown in Table 18-1. Each pin goes into a high-impedance state while the $\overline{\text{RESET}}$ signal is input, and during the oscillation stabilization time immediately after the reset function has been released.

When a high-level signal is input to the $\overline{\text{RESET}}$ pin, the reset function is released, and program execution is started after oscillation stabilization time ($2^{18}/f_X$) has elapsed. The reset function effected by an overflow in the watchdog timer is automatically released after reset, and program execution is started after the oscillation stabilization time ($2^{18}/f_X$) has elapsed (refer to **Figures 18-2** through **18-4**).

Cautions 1. Input a low-level signal to the $\overline{\text{RESET}}$ pin for 10 μs or longer when executing external reset input.
2. Oscillation of the main system clock is stopped while the $\overline{\text{RESET}}$ signal is input. Oscillation of the subsystem clock is not stopped but continues.
3. To release the STOP mode by the $\overline{\text{RESET}}$ input, the contents in the STOP mode are retained while the $\overline{\text{RESET}}$ signal is input. However, the port pins go into a high-impedance state.

Figure 18-1. Block Diagram of Reset Function

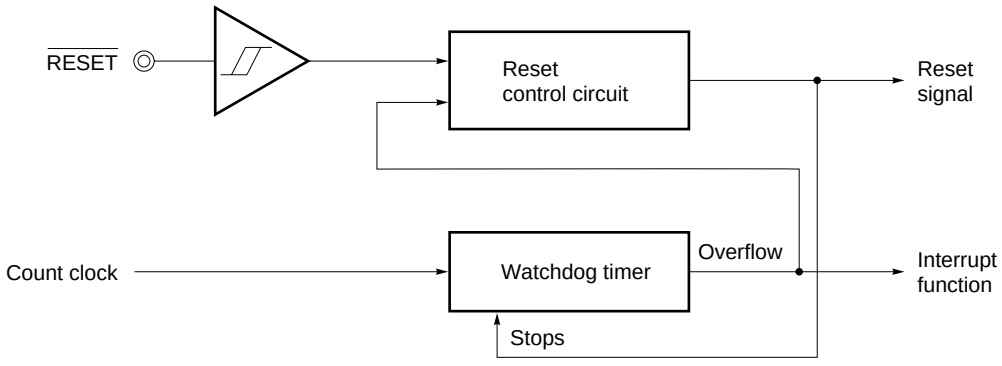


Figure 18-2. Reset Timing by $\overline{\text{RESET}}$ Input

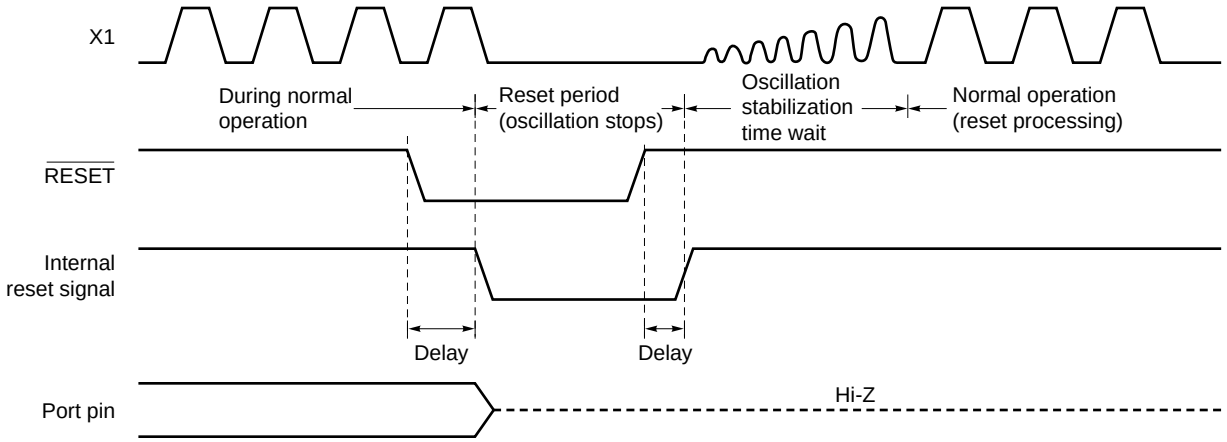


Figure 18-3. Reset Timing by Overflow in Watchdog Timer

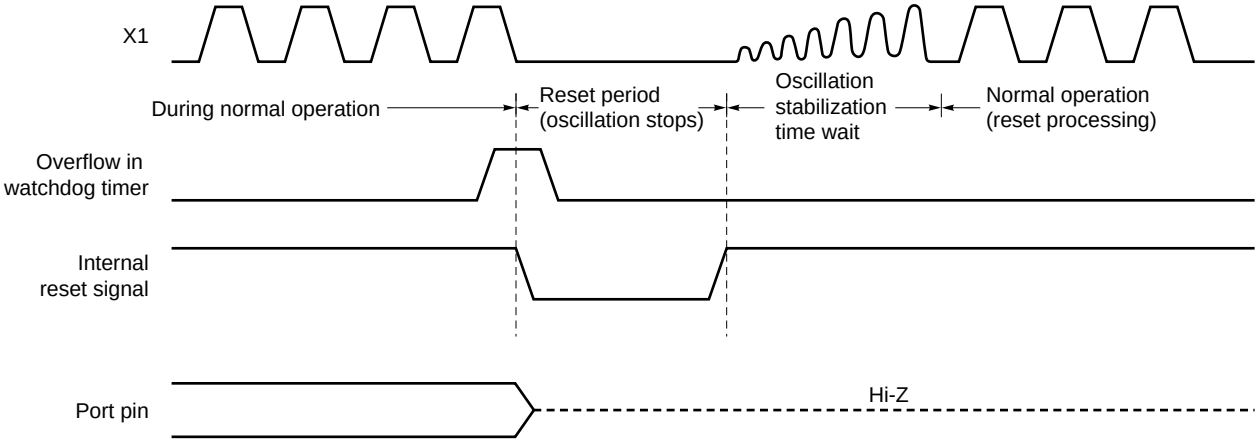


Figure 18-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

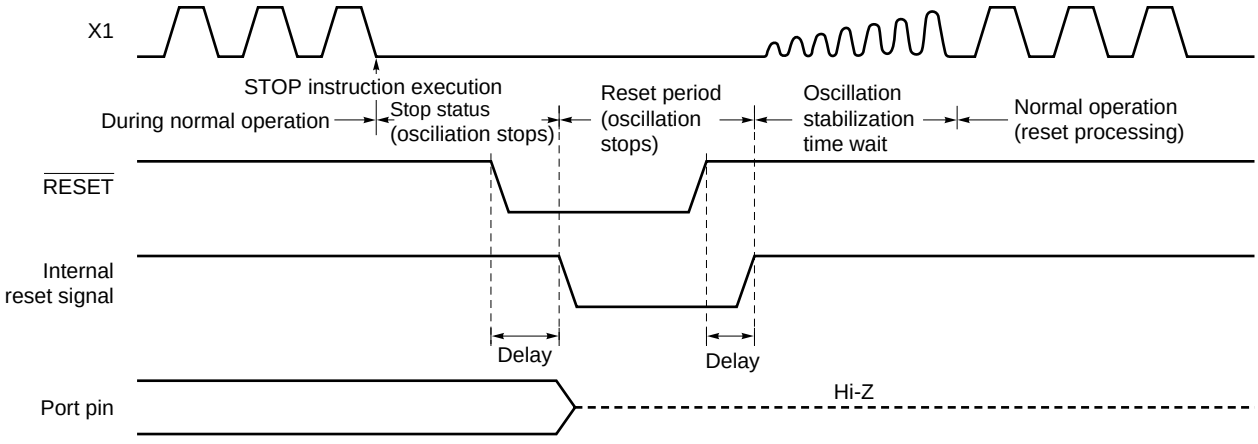


Table 18-1. Status of Each Hardware after Reset (1/2)

Hardware		Status after Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) are set
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Port (output latch)	Ports 0 to 3 (P0 to P3)	00H
	Ports 4 to 6 (P4 to P6)	Undefined
Port mode register	(PM0)	1FH
	(PM1, PM2, PM3, PM5, PM6)	FFH
Pull-up resistor option register (PUO)		00H
Processor clock control register (PCC)		04H
Memory extension mode register (MM)		10H
Memory size select register (IMS) **Note 3**		
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	0000H
	Compare register (CR00)	Undefined
	Capture register (CR01)	Undefined
	Clock select register (TCL0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
8-bit timer/event counter	Timer register (TM1, TM2)	00H
	Compare register (CR10, CR20)	Undefined
	Clock select register (TCL1)	00H
	Mode control register (TMC1)	00H
	Output control register (TOC1)	00H

Notes 1. Only the contents of the PC among hardware become undefined during reset input and oscillation stabilization time wait. The other status is not different from that after reset as above.

2. If reset is executed during the standby mode, the status before the reset is retained.

3. The value at reset of the memory size select register (IMS) differs depending on the model, as follows: μPD78011H: 42H, μPD78012H: 44H, μPD78013H: C6H, μPD78014H: C8H

Table 18-1. Status of Each Hardware after Reset (2/2)

Hardware		Status after Reset
Watch timer	Mode control register (TMC2)	00H
	Clock select register (TCL2)	
Watchdog timer	Mode register (WDTM)	00H
Serial interface	Clock select register (TCL3)	88H
	Shift register (SIO0, SIO1)	Undefined
	Mode register (CSIM0, CSIM1)	00H
	Serial bus interface control register (SBIC)	00H
	Slave address register (SVA)	Undefined
	Automatic data transmission/reception control register (ADTC)	00H
	Automatic data transmission/reception time interval specification register (ADTI)	00H
	Automatic data transmission/reception address pointer (ADTP)	00H
	Interrupt timing specification register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
Interrupt	Request flag register (IF0L, IF0H)	00H
	Mask flag register (MK0L, MK0H)	FFH
	Priority specification flag register (PR0L, PR0H)	FFH
	External interrupt mode register (INTM0)	00H
	Key return mode register (KRM)	02H
	Sampling clock select register (SCS)	00H

CHAPTER 19 OUTLINE OF μPD78P018F

The μPD78P018F is a product which incorporates a one-time PROM which can be written only one time or an EPROM capable of program write, erase, and rewrite. The μPD78P018F is a μPD78018F Subseries product but it can also cope with the μPD78014H Subseries.

19.1 Ordering Information

Part Number	Package	Internal ROM
μPD78P018FCW	64-pin plastic shrink DIP (750 mils)	One-time PROM
μPD78P018FDW	64-pin ceramic shrink DIP (with window) (750 mils)	EPROM
μPD78P018FGC-AB8	64-pin plastic QFP (14 x 14 mm)	One-time PROM
μPD78P018FGK-8A8	64-pin plastic LQFP (12 x 12 mm)	One-time PROM
μPD78P018FKK-S	64-pin ceramic WQFN (14 x 14 mm)	EPROM
μPD78P018FCW(A)	64-pin plastic shrink DIP (750 mils)	One-time PROM
μPD78P018FGC(A)-AB8	64-pin plastic QFP (14 x 14 mm)	One-time PROM

19.2 Quality Grade

Part Number	Package	Quality Grade
μPD78P018FCW	64-pin plastic shrink DIP (750 mils)	Standard
μPD78P018FDW	64-pin ceramic shrink DIP (with window) (750 mils)	Not applicable
μPD78P018FGC-AB8	64-pin plastic QFP (14 x 14 mm)	Standard
μPD78P018FGK-8A8	64-pin plastic LQFP (12 x 12 mm)	Standard
μPD78P018FKK-S	64-pin ceramic WQFN (14 x 14 mm)	Not applicable
μPD78P018FCW(A)	64-pin plastic shrink DIP (750 mils)	Special
μPD78P018FGC(A)-AB8	64-pin plastic QFP (14 x 14 mm)	Special

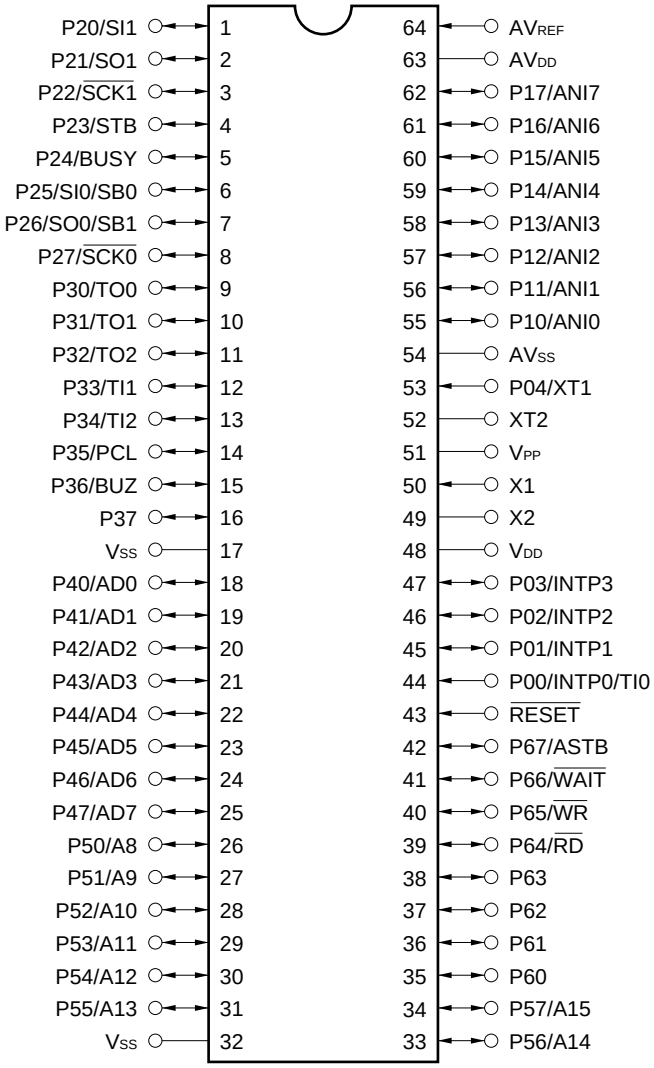
Caution The μPD78P018FDW and 78P018FKK-S versions do not have reliabilities intended for mass production of your systems. Use these models for experiment or function evaluation only.

Please refer to “Quality Grades on NEC Semiconductor Devices” (Document number C11531E) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

19.3 Pin Configuration (Top View)

(1) Normal operating mode

- 64-pin plastic shrink DIP (750 mils)
  μPD78P018FCW, 78P018FCW(A)
- 64-pin ceramic shrink DIP (with window) (750 mils)
  μPD78P018FDW

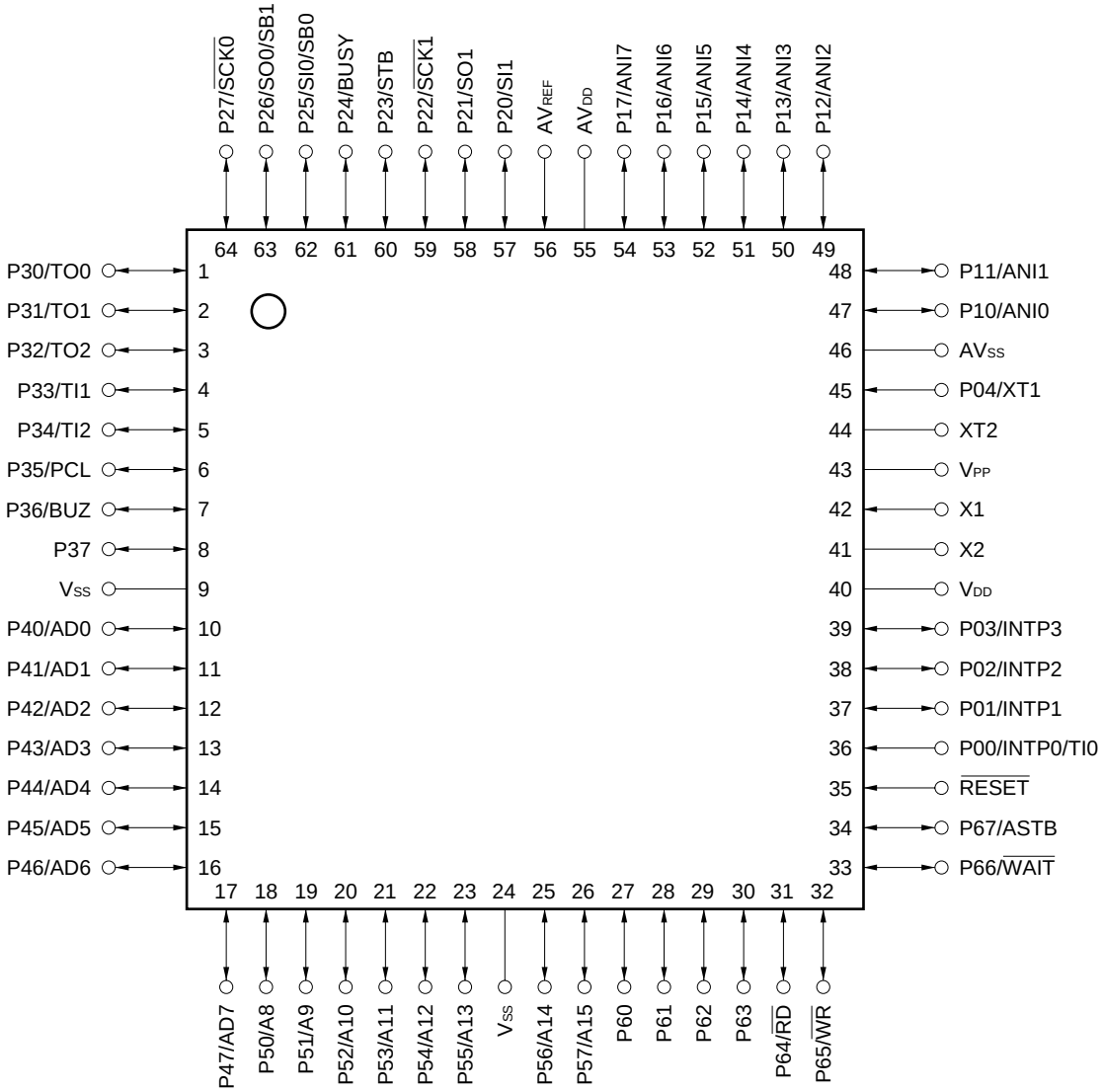


Cautions 1. Connect directly AVDD pin to VDD.

2. Connect directly AVSS pin to VSS.

★ **3. Connect directly VPP pin to VSS.**

- 64-pin plastic QFP (14 x 14 mm)
  μPD78P018FGC-AB8, 78P018FGC(A)-AB8
- 64-pin plastic LQFP (12 x 12 mm)
  μPD78P018FGK-8A8
- 64-pin ceramic WQFN (14 x 14 mm)
  μPD78P018FKK-S



Cautions 1. Connect directly AV_{DD} pin to V_{DD}.

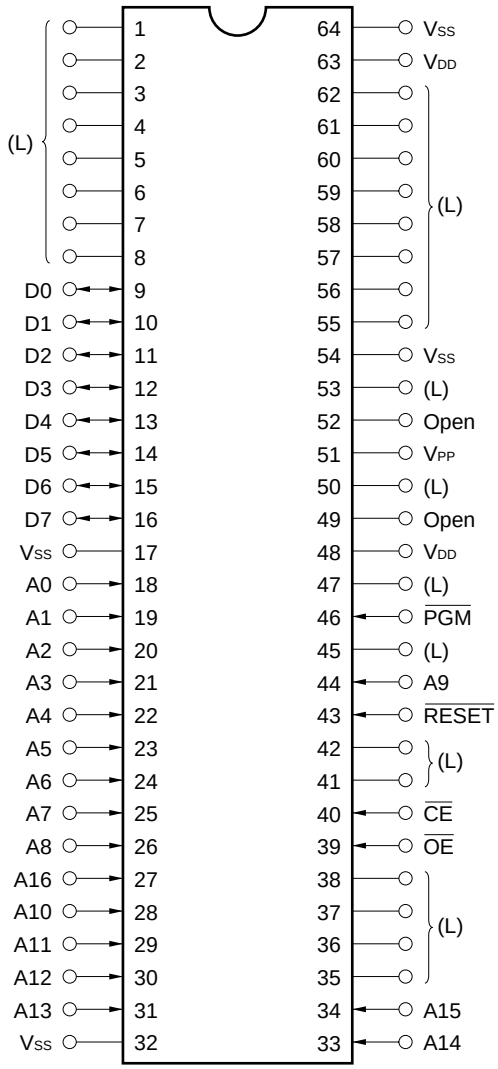
2. Connect directly AV_{SS} pin to V_{SS}.

★ **3. Connect directly V_{PP} pin to V_{SS}.**

A8 to A15	:	Address Bus
AD0 to AD7	:	Address/Data Bus
ANI0 to ANI7	:	Analog Input
ASTB	:	Address Strobe
AV_{DD}	:	Analog Power Supply
AV_{REF}	:	Analog Reference Voltage
AV_{SS}	:	Analog Ground
BUSY	:	Busy
BUZ	:	Buzzer Clock
INTP0 to INTP3	:	Interrupt from Peripherals
P00 to P04	:	Port 0
P10 to P17	:	Port 1
P20 to P27	:	Port 2
P30 to P37	:	Port 3
P40 to P47	:	Port 4
P50 to P57	:	Port 5
P60 to P67	:	Port 6
PCL	:	Programmable Clock
$\overline{RD}$	:	Read Strobe
$\overline{RESET}$	:	Reset
SB0, SB1	:	Serial Bus
$\overline{SCK0}$, $\overline{SCK1}$	:	Serial Clock
SI0, SI1	:	Serial Input
SO0, SO1	:	Serial Output
STB	:	Strobe
TI0 to TI2	:	Timer Input
TO0 to TO2	:	Timer Output
V_{DD}	:	Power Supply
V_{PP}	:	Programming Power Supply
V_{SS}	:	Ground
$\overline{WAIT}$	:	Wait
$\overline{WR}$	:	Write Strobe
X1, X2	:	Crystal (Main System Clock)
XT1, XT2	:	Crystal (Subsystem Clock)

(2) PROM programming mode

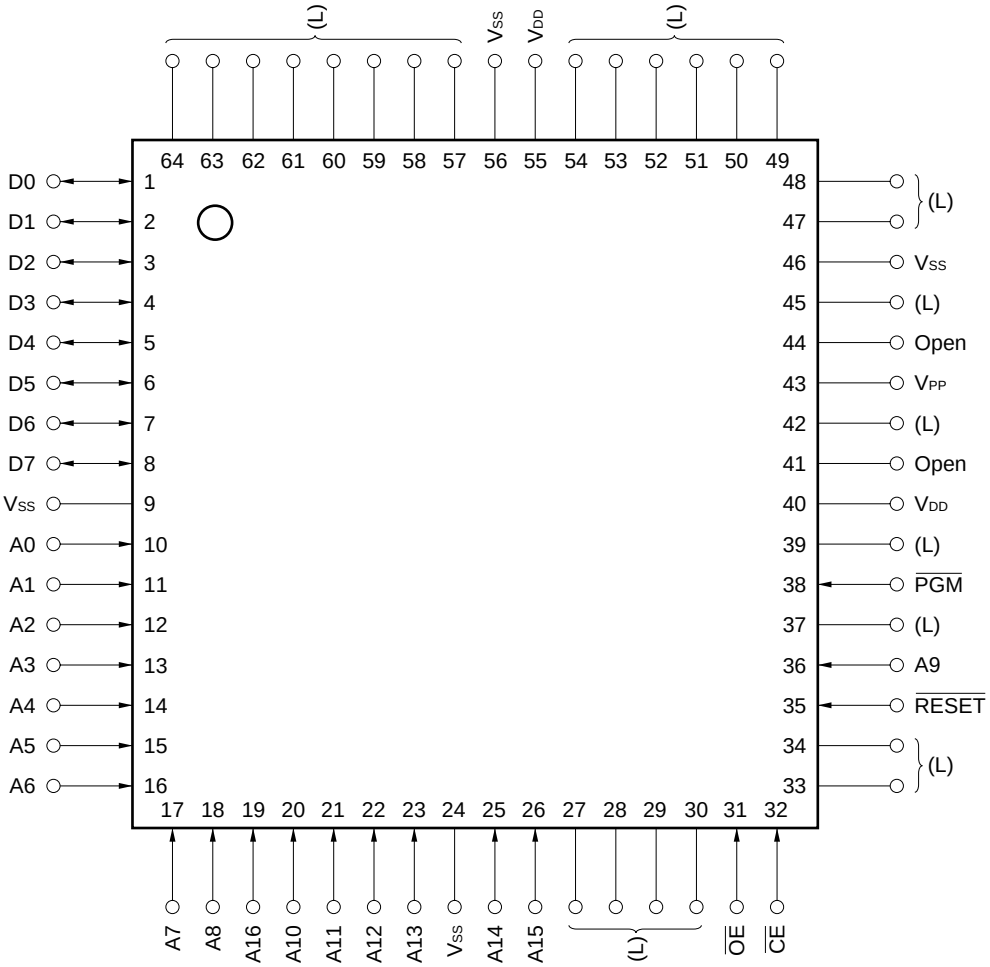
- 64-pin plastic shrink DIP (750 mils)
  μPD78P018FCW, 78P018FCW(A)
- 64-pin ceramic shrink DIP (with window) (750 mils)
  μPD78P018FDW



Cautions 1. (L) : Connect independently to V_{SS} via a pull-down resistor.
2. V_{SS} : Connect to the ground.
3. $\overline{RESET}$: Set to the low level.
4. Open : Do not connect anything.

Phase-out/Discontinued

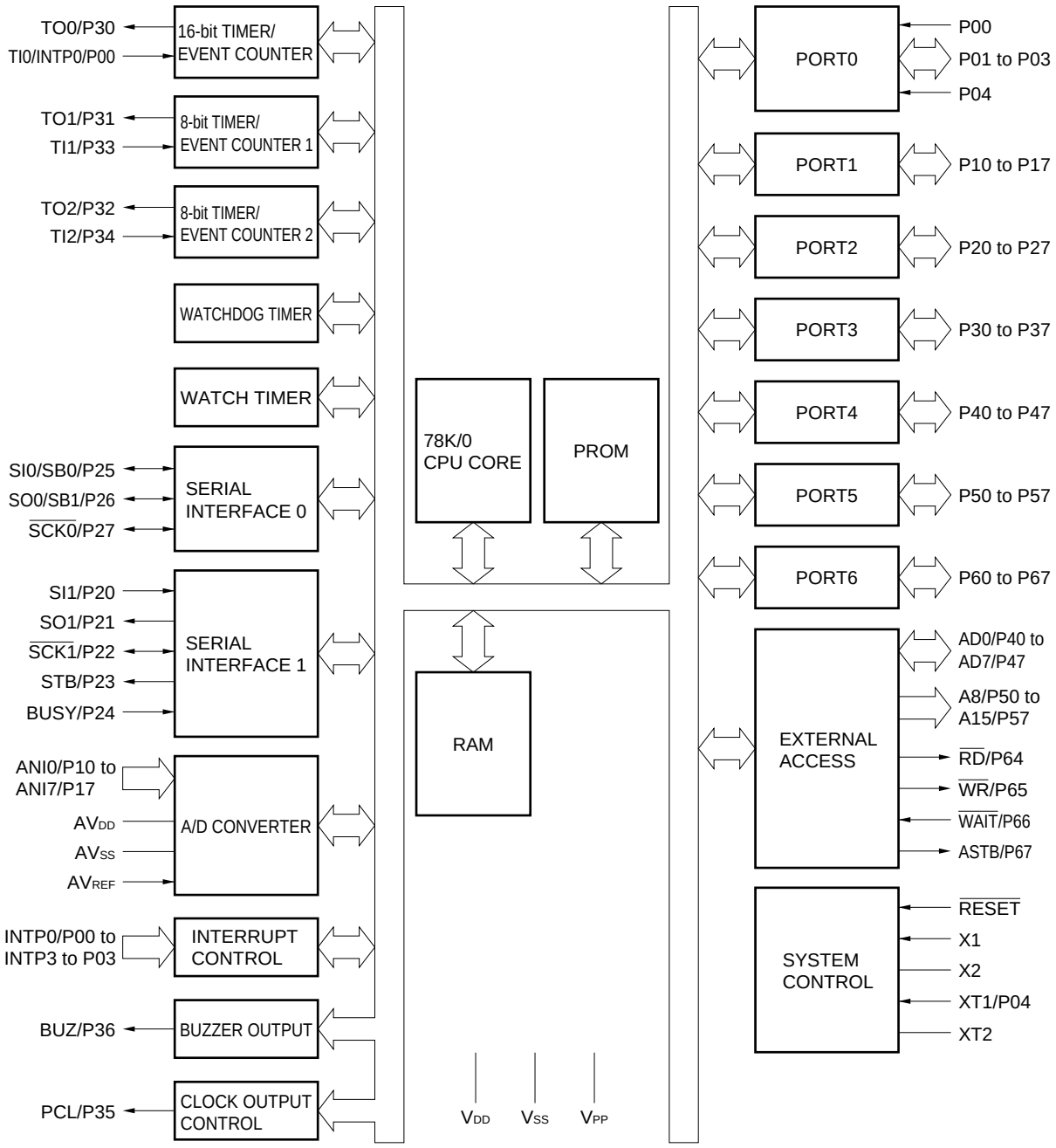
- 64-pin plastic QFP (14 x 14 mm)
  μPD78P018FGC-AB8, 78P018FGC(A)-AB8
- 64-pin plastic LQFP (12 x 12 mm)
  μPD78P018FGK-8A8
- 64-pin ceramic WQFN (14 x 14 mm)
  μPD78P018FKK-S



Cautions 1. (L) : Connect independently to V_{SS} via a pull-down resistor.
2. V_{SS} : Connect to the ground.
3. $\overline{RESET}$: Set to the low level.
4. Open : Do not connect anything.

A0 to A16	:	Address Bus	$\overline{RESET}$	:	Reset
$\overline{CE}$	:	Chip Enable	V_{DD}	:	Power Supply
D0 to D7	:	Data Bus	V_{PP}	:	Programming Power Supply
$\overline{OE}$	:	Output Enable	V_{SS}	:	Ground
$\overline{PGM}$	:	Program			

19.4 Block Diagram



19.5 Pin Function List

19.5.1 Normal operating mode pins

(1) Port pins (1/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P00	Input	Port 0.	Dedicated to input.	Input	INTP0/TI0
P01	I/O	5-bit I/O port.	Bit-wise I/O specifiable.		INTP1
P02			When used as an input port, an on-chip		INTP2
P03			pull-up resistor can be used by software.		INTP3
P04[Note1]	Input		Dedicated to input.		XT1
P10 to P17	I/O	Port 1. 8-bit I/O port. Bit-wise I/O specifiable. When used as an input port, an on-chip pull-up resistor can be used by software.[Note2]			ANI0 to ANI7
P20	I/O	Port 2.			SI1
P21		8-bit I/O port.			SO1
P22		Bit-wise I/O specifiable.			$\overline{SCK1}$
P23		When used as an input port, an on-chip pull-up resistor			STB
P24		can be used by software.			BUSY
P25					SI0/SB0
P26					SO0/SB1
P27					$\overline{SCK0}$
P30	I/O	Port 3.			TO0
P31		8-bit I/O port.			TO1
P32		Bit-wise I/O specifiable.			TO2
P33		When used as an input port, an on-chip pull-up resistor			TI1
P34		can be used by software.			TI2
P35					PCL
P36					BUZ
P37					—

Notes 1. When the P04/XT1 pin is used as an input port, set processor control register (PCC) bit 6 (FRC) to 1. Do not use the on-chip feedback resistor of the subsystem clock oscillator.

2. When the P10/ANI0 to P17/ANI7 pins are used as analog pins of the A/D converter, set port 1 to the input mode.
However, the pull-up resistor is automatically disabled.

(1) Port pins (2/2)

Pin Name	I/O	Function		After Reset	Alternate Function
P40 to P47	I/O	Port 4. 8-bit I/O port. I/O specifiable in 8-bit units. When used as an input port, an on-chip pull-up resistor can be used by software. Falling edge detection sets the test input flag (KRIF) to 1.		Input	AD0 to AD7
P50 to P57	I/O	Port 5. 8-bit I/O port. Can drive the LED directly. Bit-wise I/O specifiable. When used as an input port, an on-chip pull-up resistor can be used by software.			A8 to A15
P60	I/O	Port 6. 8-bit I/O port. Bit-wise I/O specifiable.	N-ch open-drain I/O port. An on-chip pull-up resistor can be used for only the mask ROM version by mask option. Can drive the LED directly.		—
P61					
P62					
P63					
P64			When used as an input port, an on-chip pull-up resistor can be used by software.		$\overline{RD}$
P65					$\overline{WR}$
P66					$\overline{WAIT}$
P67					ASTB

(2) Non-port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	Valid edges (rising edge, falling edge, and both rising and falling edges) specifiable external interrupt request input.	Input	P00/TI0
INTP1				P01
INTP2				P02
INTP3		Falling edge detected external interrupt request input.		P03
SI0	Input	Serial data input of serial interface.		P25/SB0
SI1				P20
SO0	Output	Serial data output of serial interface.		P26/SB1
SO1				P21
SB0	I/O	Serial data I/O of serial interface.		P25/SI0
SB1				P26/SO0
$\overline{SCK0}$	I/O	Serial clock I/O of serial interface.		P27
$\overline{SCK1}$				P22
STB	Output	Automatic transmit/receive strobe output of serial interface.		P23
BUSY	Input	Automatic transmit/receive busy input of serial interface.		P24
TI0	Input	External count clock input to 16-bit timer (TM0).		P00/INTP0
TI1		External count clock input to 8-bit timer (TM1).		P33
TI2		External count clock input to 8-bit timer (TM2).		P34
TO0	Output	16-bit timer output (dual-function as 14-bit PWM output).		P30
TO1		8-bit timer (TM1) output.		P31
TO2		8-bit timer (TM2) output.		P32
PCL	Output	Clock output (for main system clock and subsystem clock trimming).		P35
BUZ	Output	Buzzer output.		P36
AD0 to AD7	I/O	Lower address/data bus for external memory expansion.		P40 to P47
A8 to A15	Output	Higher address bus for external memory expansion.		P50 to P57
$\overline{RD}$	Output	External memory read strobe signal output.		P64
$\overline{WR}$		External memory write strobe signal output.		P65
$\overline{WAIT}$	Input	External memory access wait insertion.		P66
ASTB	Output	Strobe output to externally latch the address information output to ports 4 and 5 to access the external memory.		P67

(2) Non-port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
ANI0 to ANI7	Input	A/D converter analog input.	Input	P10 to P17
AV_{REF}	Input	A/D converter reference voltage input.	—	—
AV_{DD}	—	A/D converter analog power supply. Connect to V_{DD}.	—	—
AV_{SS}	—	A/D converter ground potential. Connect to V_{SS}.	—	—
$\overline{RESET}$	Input	System reset input.	—	—
X1	Input	Main system clock oscillation crystal connection.	—	—
X2	—		—	—
XT1	Input	Subsystem clock oscillation crystal connection.	Input	P04
XT2	—		—	—
V_{DD}	—	Positive power supply.	—	—
V_{PP}	—	High voltage application for program write/verify. Connect to V_{SS} directly in normal operating mode.	—	—
V_{SS}	—	Ground potential.	—	—

19.5.2 PROM programming mode pins

Pin Name	I/O	Function
$\overline{RESET}$	Input	PROM programming mode set. When +5 V or +12.5 V is applied to the V_{PP} pin and a low level is applied to the $\overline{RESET}$ pin, the PROM programming mode is set.
V_{PP}	Input	High voltage application for PROM programming mode set and program write/verify.
A0 to A16	Input	Address bus.
D0 to D7	I/O	Data bus.
$\overline{CE}$	Input	PROM enable input/program pulse input.
$\overline{OE}$	Input	Read strobe input to PROM.
$\overline{PGM}$	Input	Program/program inhibit inputs for PROM programming mode.
V_{DD}	—	Positive power supply.
V_{SS}	—	Ground potential.

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

Table 19-1 shows the I/O circuit type of each pin and how to handle unused pins.
For the configuration of each type of I/O circuit, refer to **Figure 19-1**.

Table 19-1. I/O Circuit Type of Each Pin (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP0/TI0	2	Input	Connect to V_{SS}.
P01/INTP1	8-A	I/O	Independently connect to V_{SS} via a resistor.
P02/INTP2			
P03/INTP3			
P04/XT1	16	Input	Connect to V_{DD}.
P10/ANI0 to P17/ANI7	11	I/O	Independently connect to V_{DD} or V_{SS} via a resistor.
P20/SI1	8-A		
P21/SO1	5-A		
P22/$\overline{SCK1}$	8-A		
P23/STB	5-A		
P24/BUSY	8-A		
P25/SI0/SB0	10-A		
P26/SO0/SB1			
P27/$\overline{SCK0}$			
P30/TO0	5-A		
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			
P37			
P40/AD0 to P47/AD7	5-E		Independently connect to V_{DD} via a resistor.
P50/A8 to P57/A15	5-A		Independently connect to V_{DD} or V_{SS} via a resistor.
P60 to P63 (Mask ROM version)	13-B		Independently connect to V_{DD} via a resistor.
P60 to P63 (PROM version)	13-D		
P64/$\overline{RD}$	5-A		Independently connect to V_{DD} or V_{SS} via a resistor.
P65/$\overline{WR}$			
P66/$\overline{WAIT}$			
P67/ASTB			

Table 19-1. I/O Circuit Type of Each Pin (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
$\overline{\text{RESET}}$	2	Input	—
XT2	16	—	Leave open.
AVREF	—		Connect to V_{SS}.
AVDD			Connect to V_{DD}.
AVSS			Connect to V_{SS}.
VPP			Directly connect to V_{SS}.

Figure 19-1. Pin I/O Circuit List (1/2)

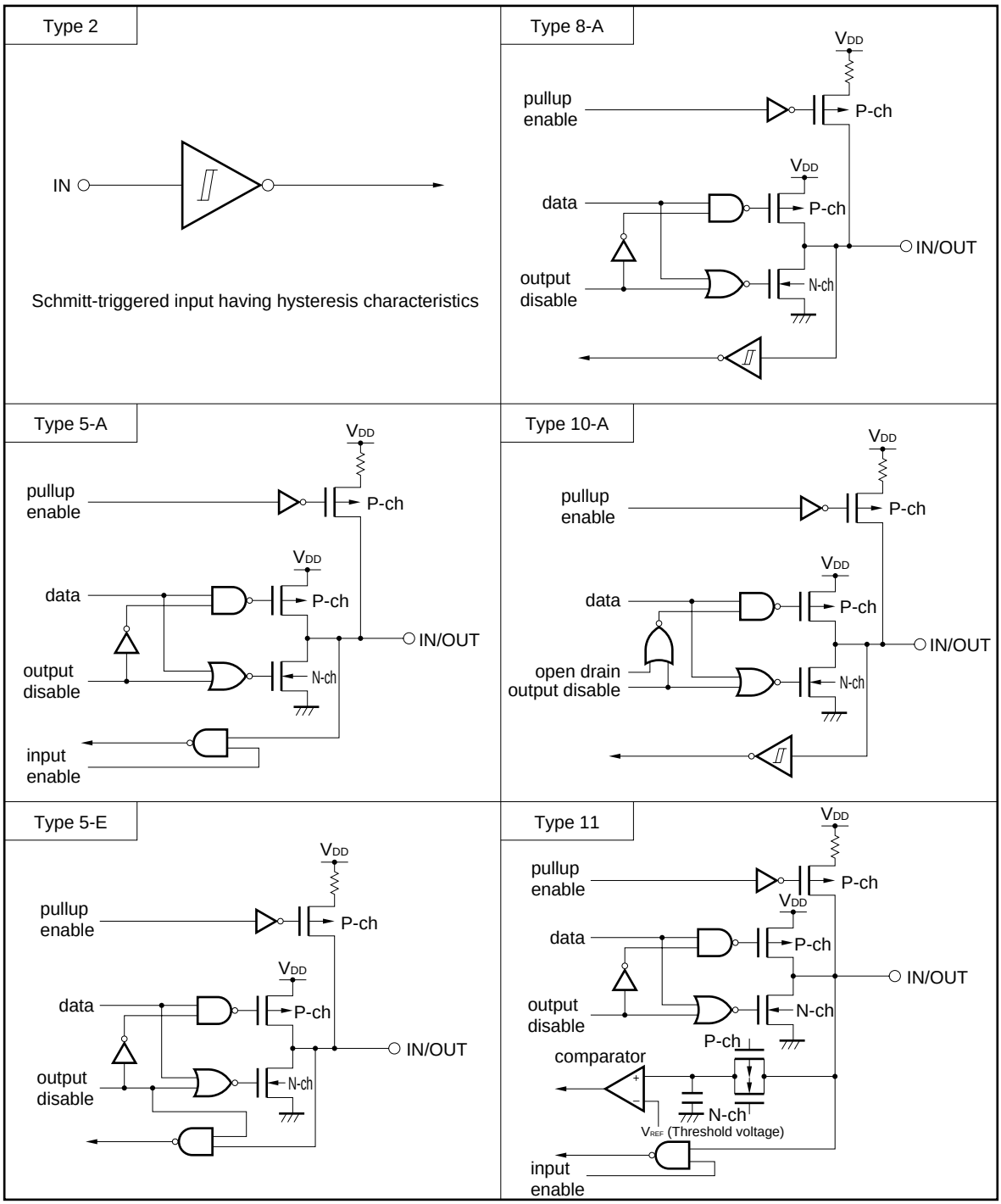
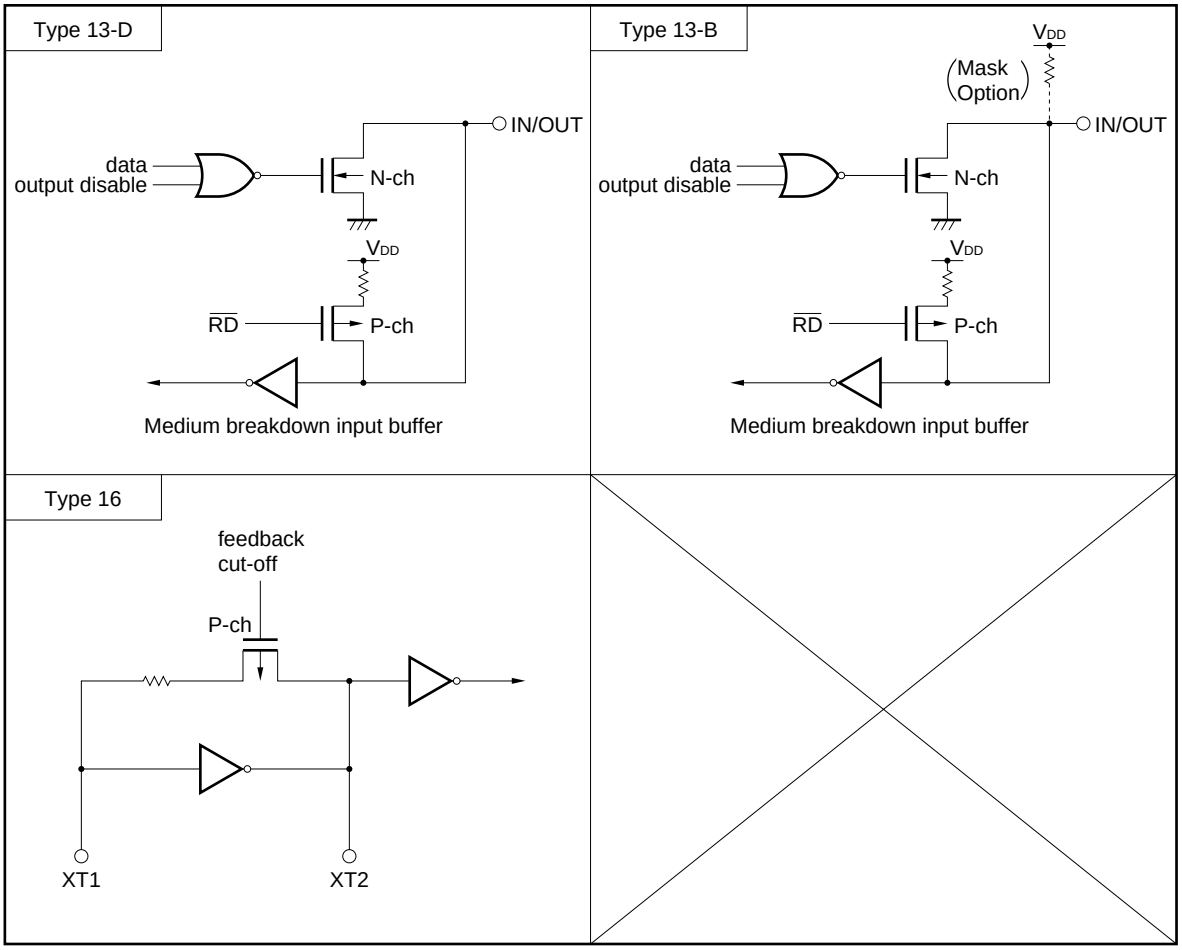
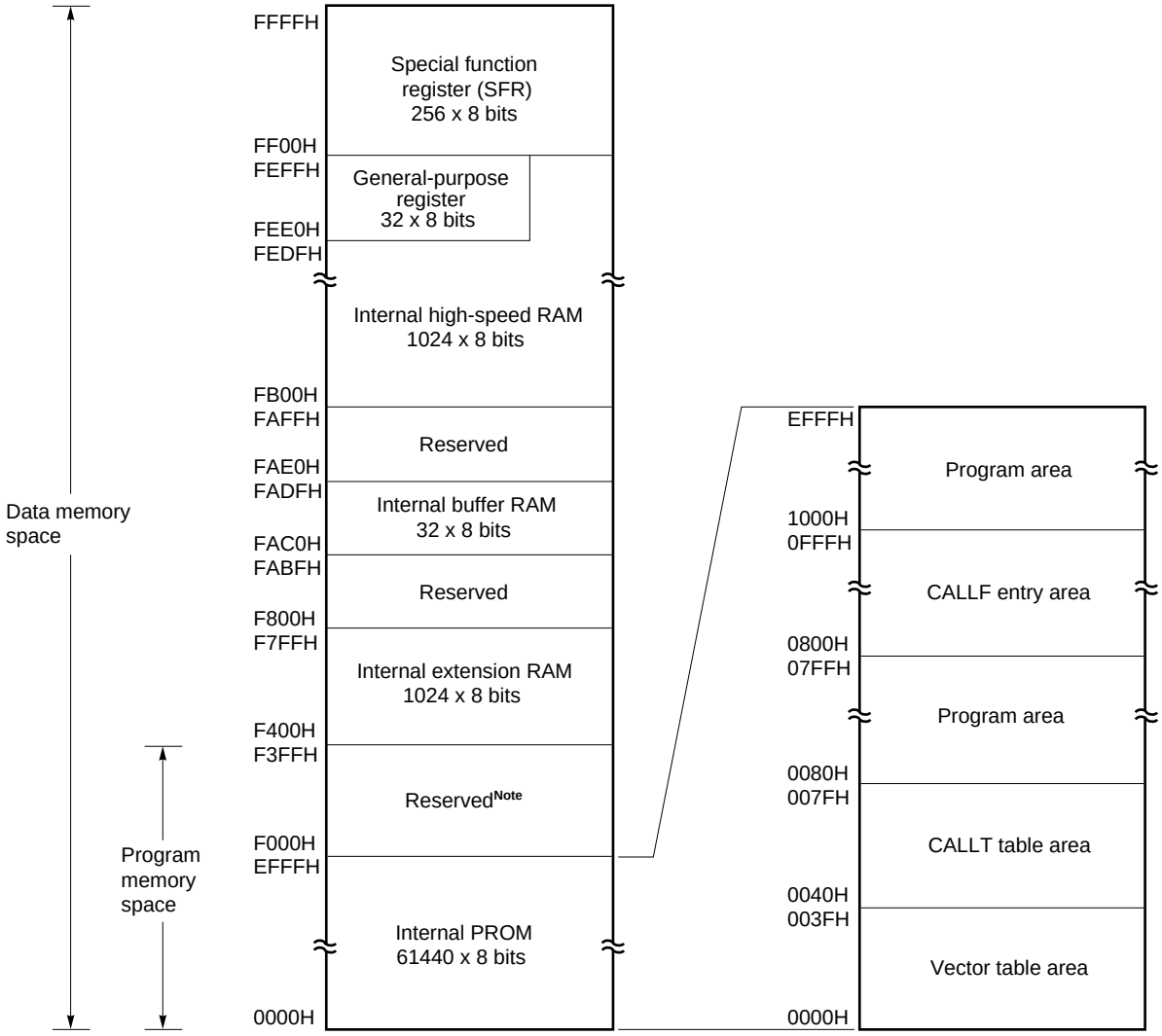


Figure 19-1. Pin I/O Circuit List (2/2)



19.7 Memory Map

Figure 19-2. Memory Map (μPD78P018F)



Note When the internal PROM capacity is 60 Kbytes, the area of F000H to F3FFH cannot be used. The area of F000H to F3FFH can be used as an external memory by setting the internal PROM capacity to 56 Kbytes or less with the memory size select register (IMS).

19.8 Memory Size Select Register

The μPD78P018F is provided with a memory size select register (IMS) that can select the internal memory. By setting IMS, the memory of the μPD78P018F can be mapped in the same manner as that of the mask ROM versions each having a different memory capacity from those of the others.

To map the memory of the μPD78P018F in the same manner as that of a mask ROM version, set the value of the mask ROM version at reset to IMS.

IMS of a mask ROM version need not to be set.

IMS is set by an 8-bit memory manipulation instruction.

This register is set as shown in Table 19-2 when the $\overline{\text{RESET}}$ signal is input.

Caution When using the mask ROM version, do not set a value other than the value at reset shown in Table 19-2 to IMS.

Figure 19-3. Format of Memory Size Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	CFH	W

ROM3	ROM2	ROM1	ROM0	Selects internal ROM capacity
0	0	1	0	8 Kbytes
0	1	0	0	16 Kbytes
0	1	1	0	24 Kbytes
1	0	0	0	32 Kbytes
Others				Setting prohibited

RAM2	RAM1	RAM0	Selects internal high-speed RAM capacity
0	1	0	512 bytes
1	1	0	1024 bytes
Others			Setting prohibited

The set value of IMS to map the memory of the μPD78P018F in the same manner as that of the mask ROM version is shown in Table 19-2.

Table 19-2. Set Value of Memory Size Select Register

Model	Set Value of IMS
μPD78011H	42H
μPD78012H	44H
μPD78013H	C6H
μPD78014H	C8H

19.9 Internal Extended RAM Size Select Register

The internal extended RAM of the μPD78P018F can be mapped in the same manner as the memory map of a mask ROM version with a different sized internal extended RAM by setting the internal extended RAM size select register (IXS).

The μPD78014H Subseries does not provide IXS.

IXS is set by an 8-bit memory manipulation instruction.

This register is set to 0AH when the $\overline{\text{RESET}}$ signal is input.

Caution The internal extended RAM size select register is provided to the μPD78P018F only.

Figure 19-4. Format of Internal Extended RAM Size Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
IXS	0	0	0	0	IX RAM3	IX RAM2	IX RAM1	IX RAM0	FFF4H	0AH	W

IX RAM3	IX RAM2	IX RAM1	IX RAM0	Selects internal extended RAM capacity
1	0	1	0	1024 bytes (F400H to F7FFH)
1	1	0	0	0 bytes
Others				Setting prohibited

To make the memory allocation of the μPD78P018F the same as that of the mask ROM versions, set the IXS register to 0CH[Note].

Note Even when a program for the μPD78P018F in which “MOV IXS, #0CH” is described is executed on the product which IXS is not provided for, the operation is not affected.

19.10 PROM Programming

The μPD78P018F is provided with 60-Kbyte PROM as a program memory. To write a program into this memory, the device needs to enter the PROM programming mode by manipulating the V_{PP} and $\overline{RESET}$ pins. When these pins are not used, process them by referring to **19.3 Pin Configuration (2) PROM programming mode**.

Caution Write a program to the program memory in an address range of 0000H to EFFFH (specify the last address EFFFH). The program cannot be written with a PROM programmer that cannot specify a write address.

19.10.1 Operation mode

The PROM programming mode is set when +5 V or +12.5 V is applied to the V_{PP} pin or when a low-level signal is applied to the $\overline{RESET}$ pin. In this mode, the operation modes shown in Table 19-3 can be set by using the $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins.

By setting the read mode, the contents of the PROM can be read.

Table 19-3. Operation Modes for PROM Programming

Pin / Operation Mode	$\overline{RESET}$	V_{PP}	V_{DD}	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	D0 to D7
Page data latch	L	+12.5 V	+6.5 V	H	L	H	Data input
Page write				H	H	L	High impedance
Byte write				L	H	L	Data input
Program verify				L	L	H	Data output
Program inhibit				×	H	H	High impedance
				×	L	L	
Read		+5 V	+5 V	L	L	H	Data output
Output disable				L	H	×	High impedance
Standby				H	×	×	High impedance

Remark ×: L or H

(1) Read mode

The read mode is set by setting $\overline{CE}$ = L and $\overline{OE}$ = L.

(2) Output disable mode

The data output goes into a high-impedance state and the output disable mode is set when $\overline{OE}$ = H.

When two or more μPD78P018Fs are connected to the data bus, therefore, data can be read from any one of the devices by controlling the $\overline{OE}$ pin.

(3) Standby mode

The standby mode is set when $\overline{CE}$ = H.

In this mode, the data output goes into a high-impedance state regardless of the status of $\overline{OE}$.

(4) Page data latch mode

The page data latch mode is set when $\overline{CE}$ = H, PGM = H, and $\overline{OE}$ = L at the beginning of the page write mode.

In this mode, 4-byte data of 1 page is latched to the internal address/data latch circuit.

(5) Page write mode

Page write is executed by applying a 0.1-ms program pulse (active low) to the $\overline{PGM}$ pin with $\overline{CE}$ = H and $\overline{OE}$ = H after latching 1-page, 4-byte address and data in the page data latch mode. After that, the program can be verified when $\overline{CE}$ = L and $\overline{OE}$ = L.

If the program cannot be written by one program pulse, repeatedly execute write and verify X times (X ≤ 10).

(6) Byte write mode

Byte write is executed when a 0.1-ms program pulse (active low) is applied to the $\overline{PGM}$ pin with $\overline{CE}$ = L and $\overline{OE}$ = H. After that, the program can be verified when $\overline{OE}$ = L.

If the program cannot be written by one program pulse, repeatedly execute write and verify X times (X ≤ 10).

(7) Program verify mode

The program verify mode is set when $\overline{CE}$ = L, $\overline{PGM}$ = H, and $\overline{OE}$ = L.

Use this mode to confirm that the program has written correctly.

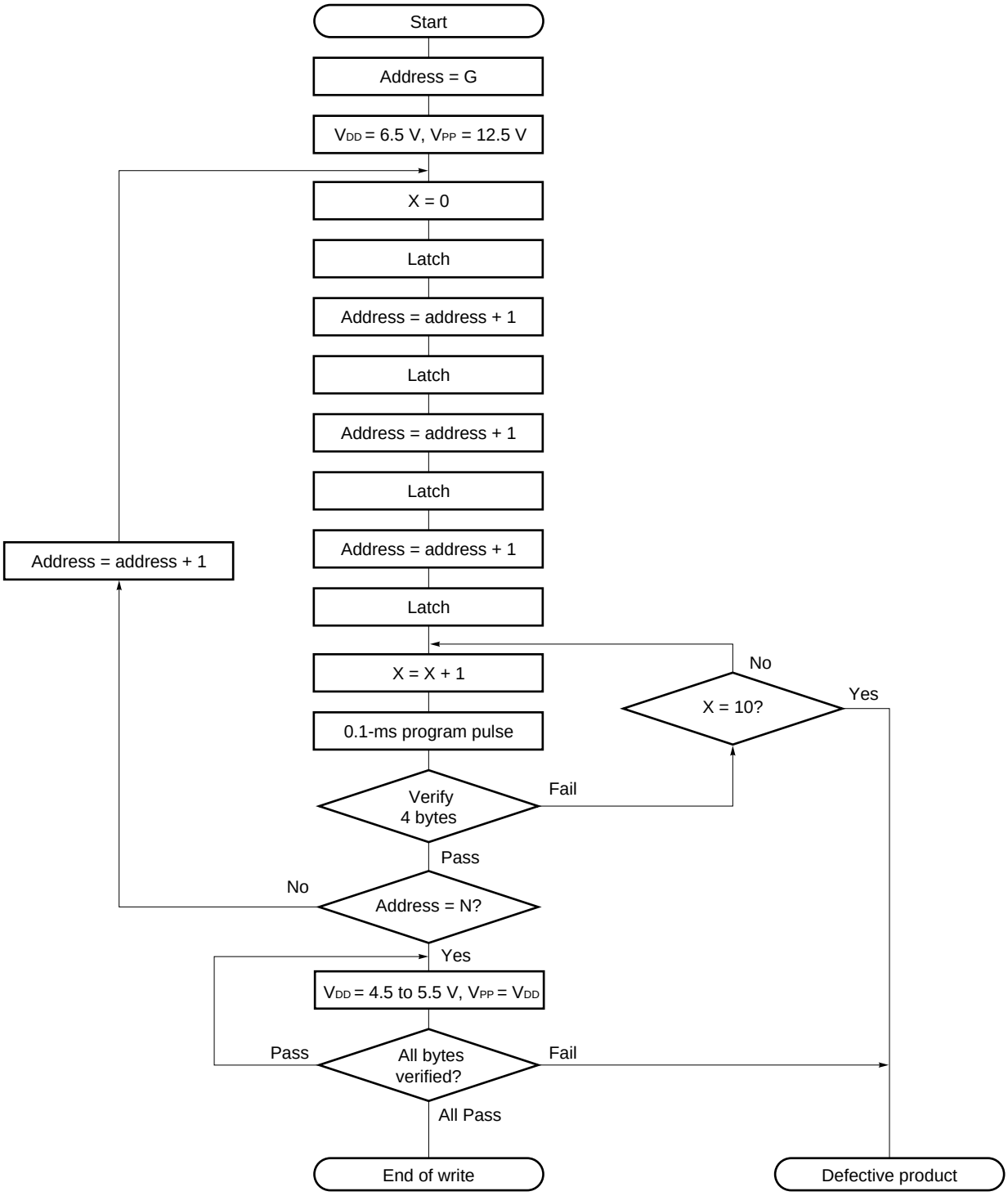
(8) Program inhibit mode

The program inhibit mode is used to write one device of the plural μPD78P018Fs whose $\overline{OE}$, V_{PP}, and D0 to D7 pins are connected in parallel.

To write a program, use the page write or byte write mode described above. At this time, the program is not written to a device whose $\overline{PGM}$ pin is high.

19.10.2 PROM write sequence

Figure 19-5. Page Program Mode Flowchart



G = start address

N = last address of program

Figure 19-6. Page Program Mode Timing

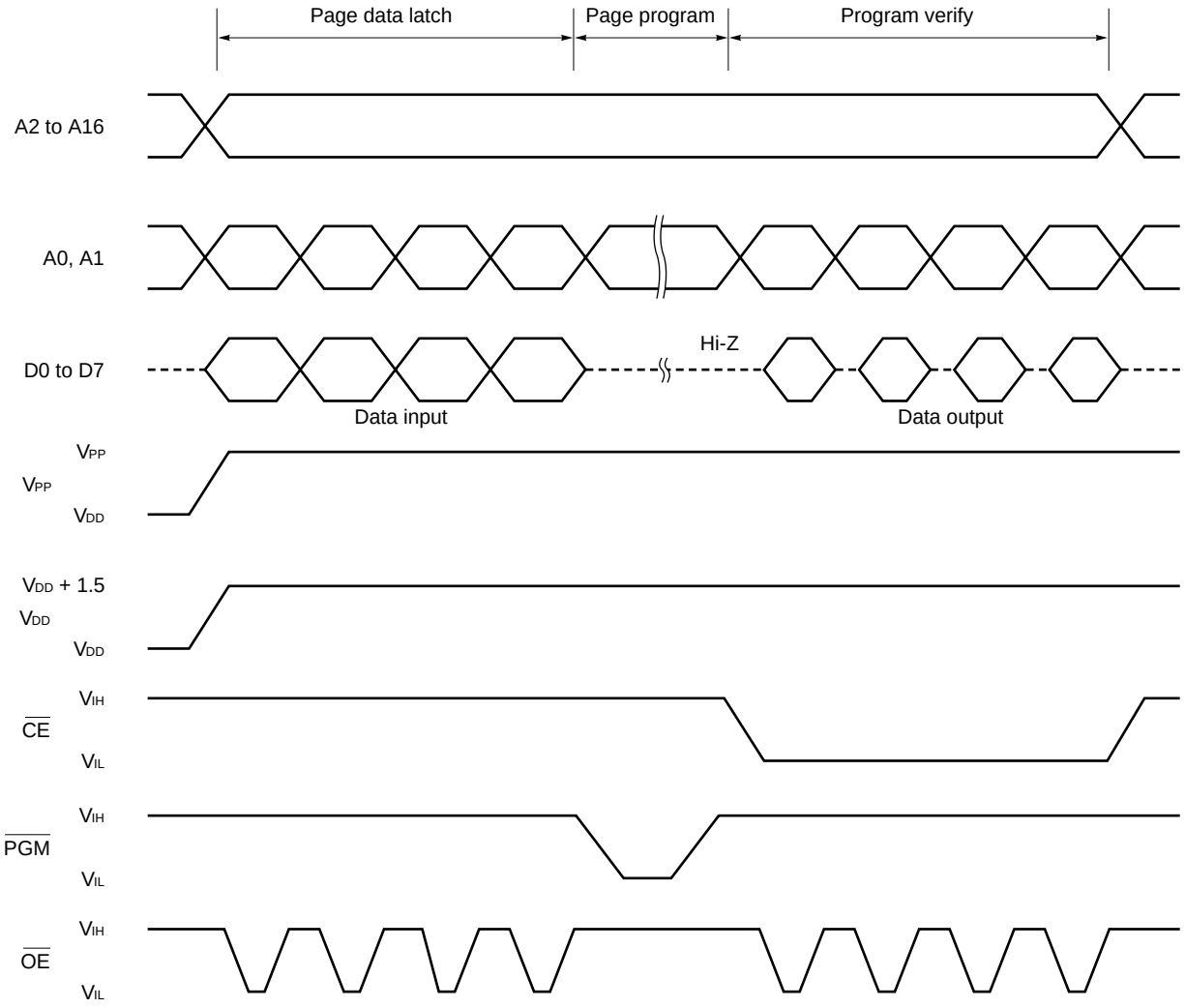
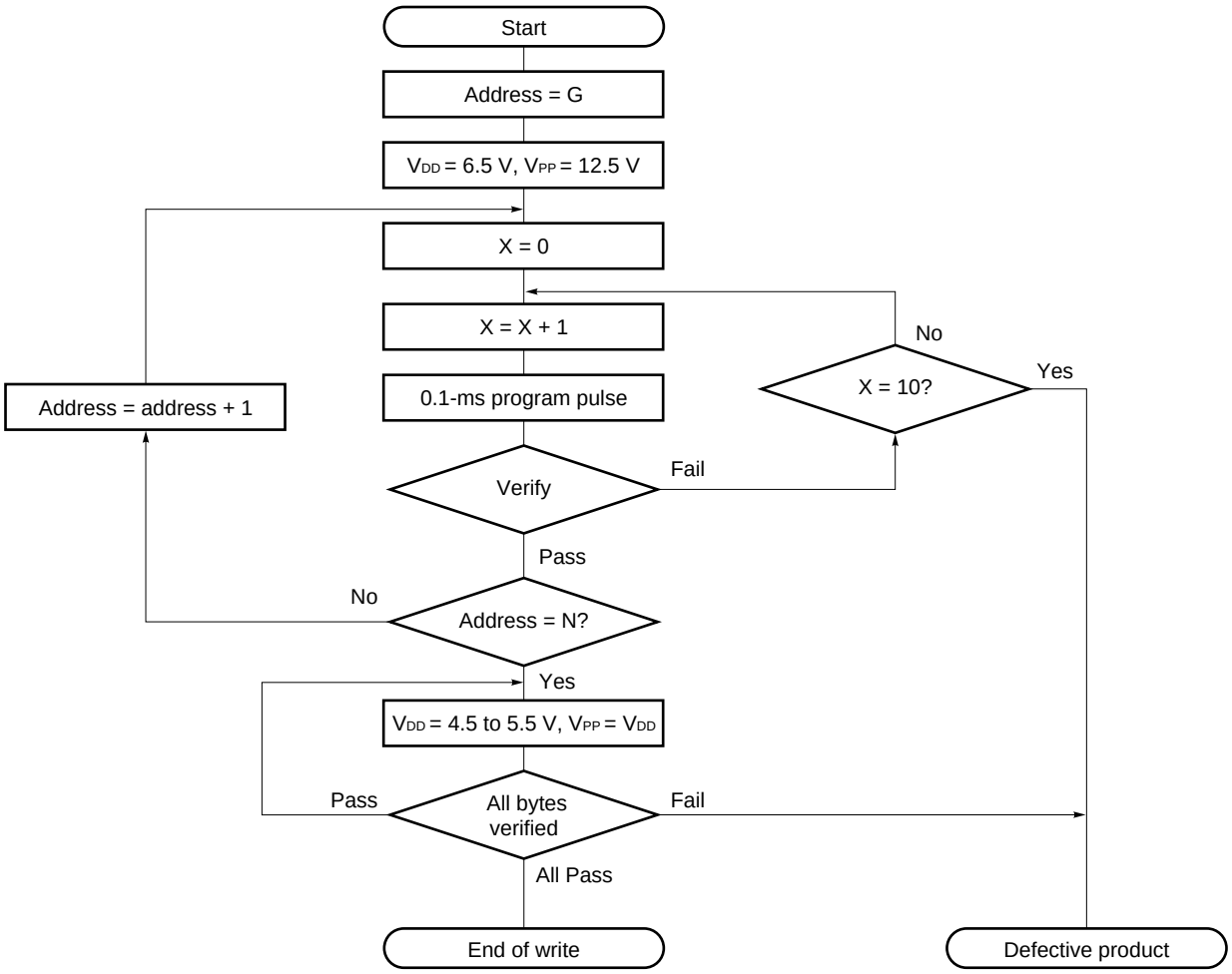


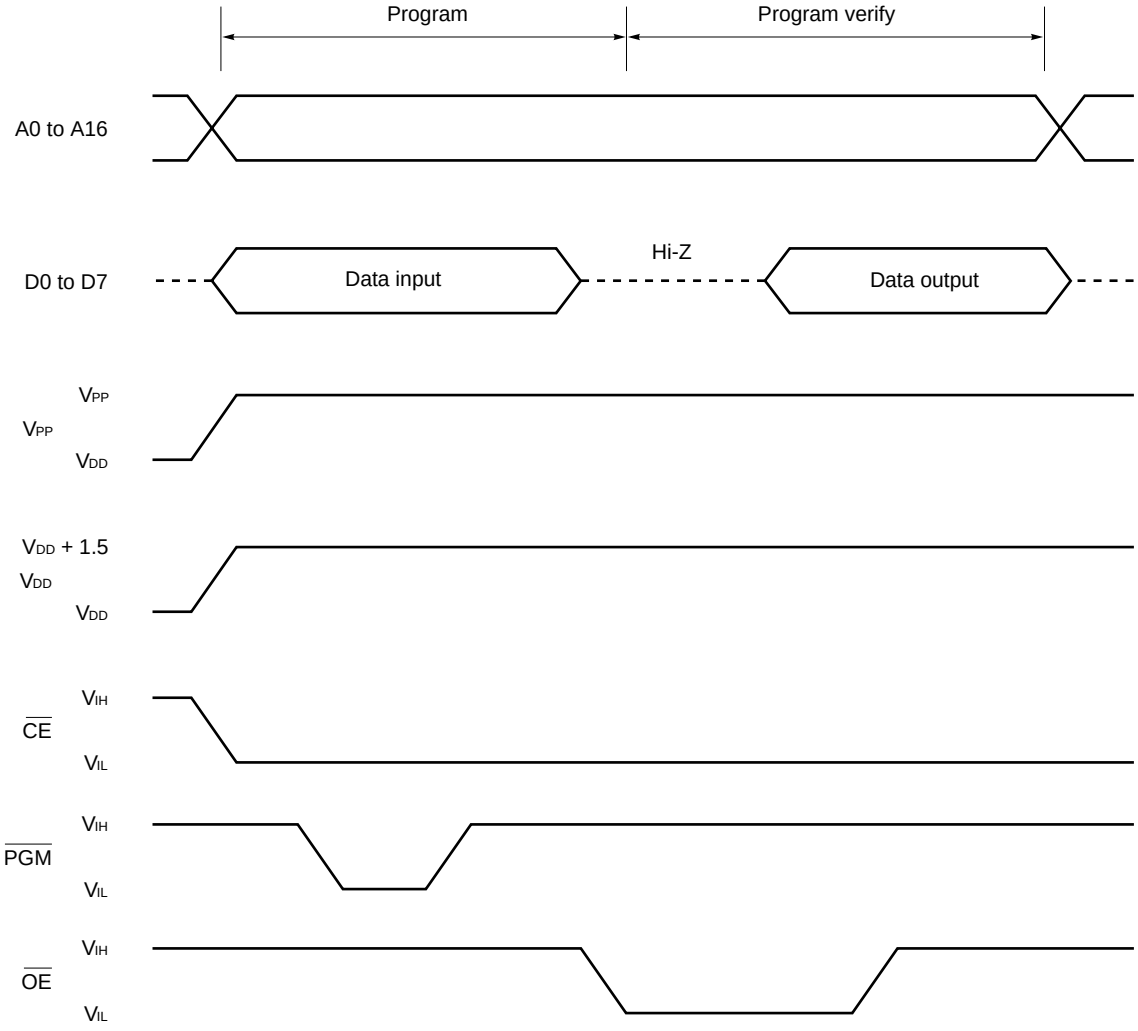
Figure 19-7. Byte Program Mode Flowchart



G = start address

N = last address of program

Figure 19-8. Byte Program Mode Timing



Cautions 1. Apply V_{DD} before V_{PP} and turn it off after V_{PP}.

2. Keep V_{PP} to within +13.5 V including the overshoot.

3. If the V_{PP} is disconnected from the socket while +12.5 V is applied to it, the reliability of the μPD78P018F may be degraded.

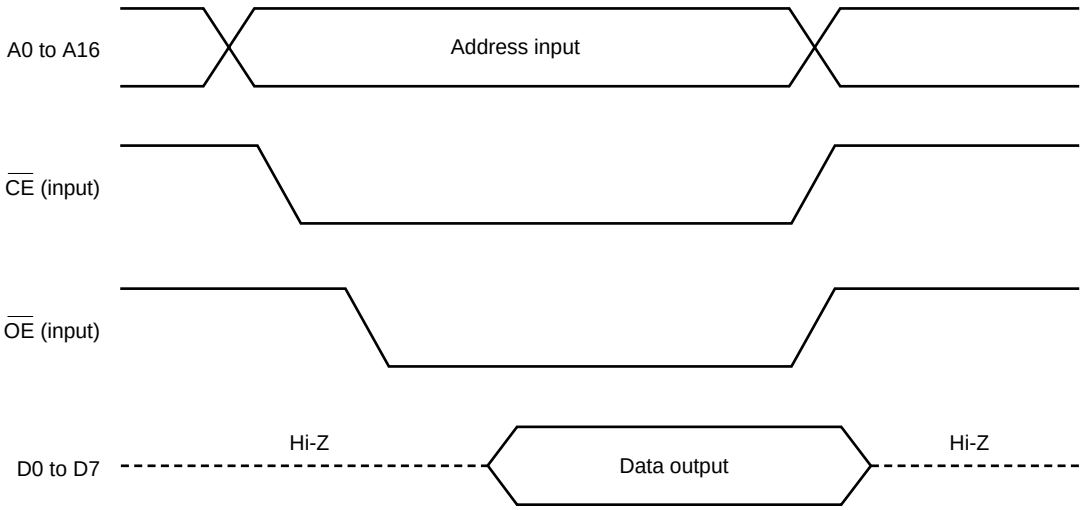
19.10.3 PROM read sequence

The contents of the PROM can be read to the external data bus (D0 to D7) in the following sequence:

(1) Fix the $\overline{\text{RESET}}$ pin to the low level. Supply +5 V to the V_{PP} pin. Connect all unused pins as specified in **19.3 Pin Configuration (2) PROM programming mode**.
(2) Supply +5 V to the V_{DD} and V_{PP} pins.
(3) Input the address of the data to be read to the A0 to A16 pins.
(4) Read mode.
(5) Data is output to the D0 to D7 pins.

Figure 19-9 shows the timing of steps (2) to (5) above.

Figure 19-9. PROM Read Timing



19.11 Erasure: μPD78P018FDW, 78P018FKK-S

The data written to the program memory of the μPD78P018FDW and 78P018FKK-S can be erased (to FFH) and new data can be rewritten to it.

To erase the data contents, cast a light whose wavelength is shorter than approximately 400 nm onto the erasure window. Usually, an ultraviolet ray of 254 nm is used. The light intensity and time required to completely erase the data contents are as follows:

- Intensity of ultraviolet ray × erasure time: 30 W•s/cm^2 min.
- Erasure time: 40 minutes or longer (with an ultraviolet lamp of 12 mW/cm^2. However, the erasure time may be extended if the performance of the ultraviolet lamp is degraded or if the erasure window is dirty.)

To erase the data, place the ultraviolet lamp at a distance of within 2.5 cm from the erasure window. If a filter is attached to the ultraviolet lamp, remove the filter before casting ultraviolet ray.

19.12 Opaque Film on Erasure Window: μPD78P018FDW, 78P018FKK-S

To protect the EPROM contents from being erased by light other than that of the erasure lamp and to prevent the internal circuit other than EPROM from malfunctioning due to light, attach an opaque film to the erasure window when the EPROM contents must be protected.

19.13 Screening of One-time PROM Version

Because of their structure, the one-time PROM versions (μPD78P018FCW, 78P018FGC-AB8, 78P018FGK-8A8, 78P018FCW(A), and 78P018FGC(A)-AB8) cannot be completely tested by NEC before shipment. It is recommended that screening be implemented to verify the PROM contents after necessary data have been written to the PROM and the product has been stored under the following temperature conditions:

Storage Temperature	Storage Time
125°C	24 hours

NEC offers a service, at a charge, called QTOP™ microcontroller, for writing, marking, screening, and verifying one-time PROMs. For details, consult NEC.

[MEMO]

CHAPTER 20 INSTRUCTION SET

This chapter lists the instruction set of the μPD78014H subseries. For the details of the operation and machine language (instruction code) of each instruction, refer to **78K/0 Series User's Manual - Instruction (U12326E)**.

20.1 Legend

20.1.1 Operand representation and description formats

In the operand field of each instruction, an operand is described according to the description format for operand representation of that instruction (for details, refer to the assembler specifications). Some operands may be described in two or more description formats. In this case, select one of them. Uppercase characters, #, !, $, and [ ] are keywords and must be described as is. The meanings of the symbols are as follows:

- # : immediate data
- ! : absolute address
- $: relative address
- [ ] : indirect address

To describe immediate data, also describe an appropriate numeric value or label. To describe a label, be sure to describe #, !, $, or [ ].

Register description formats r or rp for an operand can be described as a function name (such as X, A, or C) or absolute name (the name in parentheses in the table below, such as R0, R1, or R2).

Table 20-1. Operand Representation and Description Formats

Representation	Description Format
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 (only even address of register that can be manipulated in 16-bit units)[Note]
saddr	FE20H to FF1FH immediate data or label
saddrp	FE20H to FF1FH immediate data or label (even address only)
addr16	0000H to FFFFH immediate data or label
	(even address only for 16-bit data transfer instruction)
addr11	0800H to 0FFFH immediate data or label
addr5	0040H to 007FH immediate data or label (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 FFD0H to FFDFH cannot be addressed.

Remark For the symbols of the special function registers, refer to **Table 3-5 Special Function Register List**.

20.1.2 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
NMIS : non-maskable interrupt processing flag
( ) : memory contents indicated by contents of address or register in ( )
$\times_H$, $\times_L$: 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)
$\overline{\quad}$: inverted data
addr16 : 16-bit immediate data or label
jdisp8 : signed 8-bit data (displacement value)

20.1.3 Description in flag operation column

(Blank) : not affected
0 : cleared to 0
1 : set to 1
× : set/cleared according to result
R : value saved before is restored

20.2 Operation List

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	8	–	r ← byte	
		saddr, #byte	3	12	14	(saddr) ← byte	
		sfr, #byte	3	–	14	sfr ← byte	
		A, r **Note 3**	1	4	–	A ← r	
		r, A **Note 3**	1	4	–	r ← A	
		A, saddr	2	8	10	A ← (saddr)	
		saddr, A	2	8	10	(saddr) ← A	
		A, sfr	2	–	10	A ← sfr	
		sfr, A	2	–	10	sfr ← A	
		A, !addr16	3	16	18 + 2n	A ← (addr16)	
		!addr16, A	3	16	18 + 2m	(addr16) ← A	
		PSW, #byte	3	–	14	PSW ← byte	× × ×
		A, PSW	2	–	10	A ← PSW	
		PSW, A	2	–	10	PSW ← A	× × ×
		A, [DE]	1	8	10 + 2n	A ← (DE)	
		[DE], A	1	8	10 + 2m	(DE) ← A	
		A, [HL]	1	8	10 + 2n	A ← (HL)	
		[HL], A	1	8	10 + 2m	(HL) ← A	
		A, [HL + byte]	2	16	18 + 2n	A ← (HL + byte)	
		[HL + byte], A	2	16	18 + 2m	(HL + byte) ← A	
		A, [HL + B]	1	12	14 + 2n	A ← (HL + B)	
		[HL + B], A	1	12	14 + 2m	(HL + B) ← A	
		A, [HL + C]	1	12	14 + 2n	A ← (HL + C)	
		[HL + C], A	1	12	14 + 2m	(HL + C) ← A	

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

4. m indicates the number of wait states when the external memory extension area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**XCH**	A, r **Note 3**	1	4	–	A ↔ r	
		A, saddr	2	8	12	A ↔ (saddr)	
		A, sfr	2	–	12	A ↔ sfr	
		A, !addr16	3	16	20 + 2n + 2m	A ↔ (addr16)	
		A, [DE]	1	8	12 + 2n + 2m	A ↔ (DE)	
		A, [HL]	1	8	12 + 2n + 2m	A ↔ (HL)	
		A, [HL + byte]	2	16	20 + 2n + 2m	A ↔ (HL + byte)	
		A, [HL + B]	2	16	20 + 2n + 2m	A ↔ (HL + B)	
		A, [HL + C]	2	16	20 + 2n + 2m	A ↔ (HL + C)	
16-bit data transfer	**MOVW**	rp, #word	3	12	–	rp ← word	
		saddrp, #word	4	16	20	(saddrp) ← word	
		sfrp, #word	4	–	20	sfrp ← word	
		AX, saddrp	2	12	16	AX ← (saddrp)	
		saddrp, AX	2	12	16	(saddrp) ← AX	
		AX, sfrp	2	–	16	AX ← sfrp	
		sfrp, AX	2	–	16	sfrp ← AX	
		AX, rp **Note 4**	1	8	–	AX ← rp	
		rp, AX **Note 4**	1	8	–	rp ← AX	
		AX, !addr16	3	20	24 + 4n	AX ← (addr16)	
		!addr16, AX	3	20	24 + 4m	(addr16) ← AX	
	XCHW	AX, rp **Note 4**	1	8	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	8	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	12	16	(saddr), CY ← (saddr) + byte	× × ×
		A, r **Note 3**	2	8	–	A, CY ← A + r	× × ×
		r, A	2	8	–	r, CY ← r + A	× × ×
		A, saddr	2	8	10	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	16	18 + 2n	A, CY ← A + (saddr16)	× × ×
		A, [HL]	1	8	10 + 2n	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	16	18 + 2n	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	16	18 + 2n	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	16	18 + 2n	A, CY ← A + (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

4. Only when rp = BC, DE, HL

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

4. m indicates the number of wait states when the external memory extension area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**ADDC**	A, #byte	2	8	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	12	16	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r **Note 3**	2	8	–	A, CY ← A + r + CY	×	×	×
		r, A	2	8	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	8	10	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	16	18 + 2n	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	8	10 + 2n	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	16	18 + 2n	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	16	18 + 2n	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	16	18 + 2n	A, CY ← A + (HL + C) + CY	×	×	×
	SUB	A, #byte	2	8	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	12	16	(saddr), CY ← (saddr) – byte	×	×	×
		A, r **Note 3**	2	8	–	A, CY ← A – r	×	×	×
		r, A	2	8	–	r, CY ← r – A	×	×	×
		A, saddr	2	8	10	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	16	18 + 2n	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	8	10 + 2n	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	16	18 + 2n	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	16	18 + 2n	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	16	18 + 2n	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	8	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	12	16	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r **Note 3**	2	8	–	A, CY ← A – r – CY	×	×	×
		r, A	2	8	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	8	10	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	16	18 + 2n	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	8	10 + 2n	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	16	18 + 2n	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	16	18 + 2n	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	16	18 + 2n	A, CY ← A – (HL + C) – CY	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**AND**	A, #byte	2	8	–	A ← A ∧ byte	×
		saddr, #byte	3	12	16	(saddr) ← (saddr) ∧ byte	×
		A, r **Note 3**	2	8	–	A ← A ∧ r	×
		r, A	2	8	–	r ← r ∧ A	×
		A, saddr	2	8	10	A ← A ∧ (saddr)	×
		A, !addr16	3	16	18 + 2n	A ← A ∧ (addr16)	×
		A, [HL]	1	8	10 + 2n	A ← A ∧ (HL)	×
		A, [HL + byte]	2	16	18 + 2n	A ← A ∧ (HL + byte)	×
		A, [HL + B]	2	16	18 + 2n	A ← A ∧ (HL + B)	×
		A, [HL + C]	2	16	18 + 2n	A ← A ∧ (HL + C)	×
	OR	A, #byte	2	8	–	A ← A ∨ byte	×
		saddr, #byte	3	12	16	(saddr) ← (saddr) ∨ byte	×
		A, r **Note 3**	2	8	–	A ← A ∨ r	×
		r, A	2	8	–	r ← r ∨ A	×
		A, saddr	2	8	10	A ← A ∨ (saddr)	×
		A, !addr16	3	16	18 + 2n	A ← A ∨ (addr16)	×
		A, [HL]	1	8	10 + 2n	A ← A ∨ (HL)	×
		A, [HL + byte]	2	16	18 + 2n	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	16	18 + 2n	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	16	18 + 2n	A ← A ∨ (HL + C)	×
	XOR	A, #byte	2	8	–	A ← A ∀ byte	×
		saddr, #byte	3	12	16	(saddr) ← (saddr) ∀ byte	×
		A, r **Note 3**	2	8	–	A ← A ∀ r	×
		r, A	2	8	–	r ← r ∀ A	×
		A, saddr	2	8	10	A ← A ∀ (saddr)	×
		A, !addr16	3	16	18 + 2n	A ← A ∀ (addr16)	×
		A, [HL]	1	8	10 + 2n	A ← A ∀ (HL)	×
		A, [HL + byte]	2	16	18 + 2n	A ← A ∀ (HL + byte)	×
		A, [HL + B]	2	16	18 + 2n	A ← A ∀ (HL + B)	×
		A, [HL + C]	2	16	18 + 2n	A ← A ∀ (HL + C)	×

Notes **1.** When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks **1.** One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**CMP**	A, #byte	2	8	–	A – byte	× × ×
		saddr, #byte	3	12	16	(saddr) – byte	× × ×
		A, r **Note 3**	2	8	–	A – r	× × ×
		r, A	2	8	–	r – A	× × ×
		A, saddr	2	8	10	A – (saddr)	× × ×
		A, !addr16	3	16	18 + 2n	A – (addr16)	× × ×
		A, [HL]	1	8	10 + 2n	A – (HL)	× × ×
		A, [HL + byte]	2	16	18 + 2n	A – (HL + byte)	× × ×
		A, [HL + B]	2	16	18 + 2n	A – (HL + B)	× × ×
		A, [HL + C]	2	16	18 + 2n	A – (HL + C)	× × ×
16-bit operation	**ADDW**	AX, #word	3	12	–	AX, CY ← AX + word	× × ×
	SUBW	AX, #word	3	12	–	AX, CY ← AX – word	× × ×
	CMPW	AX, #word	3	12	–	AX – word	× × ×
Multiply/ divide	**MULU**	X	2	32	–	AX ← A × X	
	DIVUW	C	2	50	–	AX (quotient), C (remainder) ← AX ÷ C	
Increment/ decrement	**INC**	r	1	4	–	r ← r + 1	× ×
		saddr	2	8	12	(saddr) ← (saddr) + 1	× ×
	DEC	r	1	4	–	r ← r – 1	× ×
		saddr	2	8	12	(saddr) ← (saddr) – 1	× ×
	INCW	rp	1	8	–	rp ← rp + 1	
	DECW	rp	1	8	–	rp ← rp – 1	
Rotate	**ROR**	A, 1	1	4	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1 time	×
	ROL	A, 1	1	4	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1 time	×
	RORC	A, 1	1	4	–	(CY ← A_0, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1 time	×
	ROLC	A, 1	1	4	–	(CY ← A_7, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1 time	×
	ROR4	[HL]	2	20	24 + 2n + 2m	$A_{3-0} \leftarrow (HL)_{3-0}$, $(HL)_{7-4} \leftarrow A_{3-0}$, $(HL)_{3-0} \leftarrow (HL)_{7-4}$	
	ROL4	[HL]	2	20	24 + 2n + 2m	$A_{3-0} \leftarrow (HL)_{7-4}$, $(HL)_{3-0} \leftarrow A_{3-0}$, $(HL)_{7-4} \leftarrow (HL)_{3-0}$	

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

3. Except r = A

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

4. m indicates the number of wait states when the external memory extension area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
BCD adjustment	**ADJBA**		2	8	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	8	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulation	**MOV1**	CY, saddr.bit	3	12	14	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	14	CY ← sfr.bit			×
		CY, A.bit	2	8	–	CY ← A.bit			×
		CY, PSW.bit	3	–	14	CY ← PSW.bit			×
		CY, [HL].bit	2	12	14 + 2n	CY ← (HL).bit			×
		saddr.bit, CY	3	12	16	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	16	sfr.bit ← CY			
		A.bit, CY	2	8	–	A.bit ← CY			
		PSW.bit, CY	3	–	16	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	12	16 + 2n + 2m	(HL).bit ← CY			
	AND1	CY, saddr.bit	3	12	14	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	14	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	8	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	14	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	12	14 + 2n	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	12	14	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	14	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	8	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	14	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	12	14 + 2n	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	12	14	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	14	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	8	–	CY ← CY ⊻ A.bit			×
		CY, PSW. bit	3	–	14	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	12	14 + 2n	CY ← CY ⊻ (HL).bit			×

Notes **1.** When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks **1.** One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

4. m indicates the number of wait states when the external memory extension area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulation	**SET1**	saddr.bit	2	8	12	(saddr.bit) ← 1	
		sfr.bit	3	–	16	sfr.bit ← 1	
		A.bit	2	8	–	A.bit ← 1	
		PSW.bit	2	–	12	PSW.bit ← 1	× × ×
		[HL].bit	2	12	16 + 2n + 2m	(HL).bit ← 1	
	CLR1	saddr.bit	2	8	12	(saddr.bit) ← 0	
		sfr.bit	3	–	16	sfr.bit ← 0	
		A.bit	2	8	–	A.bit ← 0	
		PSW.bit	2	–	12	PSW.bit ← 0	× × ×
		[HL].bit	2	12	16 + 2n + 2m	(HL).bit ← 0	
	SET1	CY	1	4	–	CY ← 1	1
	CLR1	CY	1	4	–	CY ← 0	0
	NOT1	CY	1	4	–	CY ← $\overline{CY}$	×
Call/return	**CALL**	!addr16	3	14	–	(SP – 1) ← (PC + 3)$_H$, (SP – 2) ← (PC + 3)$_L$, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	10	–	(SP – 1) ← (PC + 2)$_H$, (SP – 2) ← (PC + 2)$_L$, PC$_{15-11}$ ← 00001, PC$_{10-0}$ ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	12	–	(SP – 1) ← (PC + 1)$_H$, (SP – 2) ← (PC + 1)$_L$, PC$_H$ ← (00000000, addr5 + 1), PC$_L$ ← (00000000, addr5), SP ← SP – 2	
	BRK		1	12	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)$_H$, (SP – 3) ← (PC + 1)$_L$, PC$_H$ ← (003FH), PC$_L$ ← (003EH), SP ← SP – 3, IE ← 0	
	RET		1	12	–	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), SP ← SP + 2	
	RETI		1	12	–	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R R R
	RETB		1	12	–	PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R

Notes **1.** When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks **1.** One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

4. m indicates the number of wait states when the external memory extension area is written.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Stack manipulation	**PUSH**	PSW	1	4	–	(SP – 1) ← PSW, SP ← SP – 1	
		rp	1	8	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2	
	POP	PSW	1	4	–	PSW ← (SP), SP ← SP + 1	R R R
		rp	1	8	–	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2	
	MOVW	SP, #word	4	–	20	SP ← word	
		SP, AX	2	–	16	SP ← AX	
		AX, SP	2	–	16	AX ← SP	
Unconditional branch	**BR**	!addr16	3	12	–	PC ← addr16	
		$addr16	2	12	–	PC ← PC + 2 + jdisp8	
		AX	2	16	–	PC_H ← A, PC_L ← X	
Conditional branch	**BC**	$addr16	2	12	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	12	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	12	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	12	–	PC ← PC + 2 + jdisp8 if Z = 0	
	BT	saddr.bit, $addr16	3	16	18	PC ← PC + 3 + jdisp8 if(saddr.bit) = 1	
		sfr.bit, $addr16	4	–	22	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	16	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	18	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	20	22 + 2n	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	20	22	PC ← PC + 4 + jdisp8 if(saddr.bit) = 0	
		sfr.bit, $addr16	4	–	22	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	16	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	22	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	20	22 + 2n	PC ← PC + 3 + jdisp8 if (HL).bit = 0	

Notes 1. When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks 1. One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

Instruction Group	Mnemonic	Operand	Byte	Clock		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BTCLR**	saddr.bit, $addr16	4	20	24	PC ← PC + 4 + jdisp8 if(saddr.bit) = 1 then reset(saddr.bit)	
		sfr.bit, $addr16	4	–	24	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	16	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	24	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	20	24 + 2n + 2m	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	12	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	12	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr, $addr16	3	16	20	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if(saddr) ≠ 0	
CPU control	**SEL**	RBn	2	8	–	RBS1, 0 ← n	
	NOP		1	4	–	No operation	
	EI		2	–	12	IE ← 1(Enable interrupt)	
	DI		2	–	12	IE ← 0(Disable interrupt)	
	HALT		2	12	–	Set HALT mode	
	STOP		2	12	–	Set STOP mode	

Notes **1.** When the internal high-speed RAM area is accessed or when an instruction that does not access data is executed

2. When an area other than the internal high-speed RAM area is accessed

Remarks **1.** One clock of an instruction is equal to one CPU clock (f_{CPU}) selected by processor clock control register (PCC).

2. The number of clocks shown is when the program is stored in the internal ROM area.

3. n indicates the number of wait states when the external memory extension area is read.

4. m indicates the number of wait states when the external memory extension area is written.

20.3 Instruction List by Addressing

(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

1st Operand \ 2nd 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 for r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ 2nd 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

1st Operand \ 2nd 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/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

1st Operand \ 2nd 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

APPENDIX A DIFFERENCES BETWEEN μPD78014, 78014H, AND 78018F SUBSERIES

Table A-1 shows the major differences between the μPD78014, 78014H, and 78018F Subseries.

Table A-1. Major Differences between μPD78014, 78014H, and 78018F Subseries (1/2)

Item \ Part Number	μPD78014 Subseries	μPD78014H Subseries	μPD78018F Subseries
EMI noise prevention	Not provided	Provided	Not provided
On-chip I^2C bus	Provided	Not provided	Provided
PROM version	μPD78P014	μPD78P018F	
Supply voltage	V_{DD} = 2.7 to 6.0 V	V_{DD} = 1.8 to 5.5 V	
Internal high-speed RAM	μPD78011B : 512 bytes μPD78012B : 512 bytes μPD78013 : 1024 bytes μPD78014 : 1024 bytes μPD78P014 : 1024 bytes	μPD78011H : 512 bytes μPD78012H : 512 bytes μPD78013H : 1024 bytes μPD78014H : 1024 bytes	μPD78011F : 512 bytes μPD78012F : 512 bytes μPD78013F : 1024 bytes μPD78014F : 1024 bytes μPD78015F : 1024 bytes μPD78016F : 1024 bytes μPD78018F : 1024 bytes μPD78P018F: 1024 bytes
Internal extended RAM	Not provided		μPD78011F : Not provided μPD78012F : Not provided μPD78013F : Not provided μPD78014F : Not provided μPD78015F : 512 bytes μPD78016F : 512 bytes μPD78018F : 1024 bytes μPD78P018F: 1024 bytes
Serial interface operating mode (Y Subseries)	• 3-wire/2-wire/SBI/I^2C: 1ch • 3-wire (automatic transmission/reception): 1ch	–	• 3-wire/2-wire/I^2C: 1ch • 3-wire (automatic transmission/reception): 1ch
Bit 5 (SIC) of interrupt timing specification register (SINT) in SBI mode (selection of INTCSI0 interrupt source)	When SIC = 1 is set: CSIIF0 (interrupt request flag) is set on detection of bus release	When SIC = 1 is set: CSIIF0 (interrupt request flag) is set on detection of bus release and termination of transfer	
Bit 5 (SIC) of interrupt timing specification register (SINT) in I^2C bus mode (selection of INTCSI0 interrupt source)	When SIC = 1 is set: CSIIF0 (interrupt request flag) is set on detection of stop condition	–	When SIC = 1 is set: CSIIF0 (interrupt request flag) is set on detection of stop condition and termination of transfer

Table A-1. Major Differences between μPD78014, 78014H, and 78018F Subseries (2/2)

Item \ Part Number	μPD78014 Subseries	μPD78014H Subseries	μPD78018F Subseries
Function of bit 7 (BSYE) of serial bus interface control register (SBIC) (Y Subseries)	Control of synchronous busy signal output • When BSYE = 0 Disables busy signal synchronized with the falling edge of $\overline{SCK0}$ clock immediately after the instruction to be cleared (0) in SBI mode. BSYE = 0 must be set in I^2C bus mode. • When BSYE = 1 Starts outputting busy signal at the falling edge of $\overline{SCK0}$ clock following acknowledge signal in SBI mode.	–	Control of N-ch open drain output for transmission in I^2C bus mode • When BSYE = 0 Output enable (transmission) • When BSYE = 1 Output disable (reception)
Automatic data transmission/ reception time interval specification register (ADTI)	Not provided	Provided	
Package	• 64-pin plastic shrink DIP (750 mil) • 64-pin ceramic shrink DIP (with window) (750 mil)Note • 64-pin plastic QFP (14 × 14 mm)	• 64-pin plastic shrink DIP (750 mil) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm)	• 64-pin plastic shrink DIP (750 mil) • 64-pin ceramic shrink DIP (with window) (750 mil)Note • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm) • 64-pin ceramic WQFN (14 × 14 mm)Note
Programmer adapter	PA-78P014CW PA-78P014GC	PA-78P018CW PA-78P018GC PA-78P018GK PA-78P018KK-S	
★ Emulation board	IE-78014-R-EM IE-78014-R-EM-A IE-78018-NS-EM1	IE-78014-R-EM-A IE-78018-NS-EM1	
Access timing to external memory	Differs between μPD78014 Subseries and other Subseries. Refer to separate Data Sheet.		
Electrical characteristics, recommended soldering conditions	Refer to separate Data Sheet.		

Note PROM version only

★

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD78014H Subseries. Figure B-1 shows development tools.

Figure B-1. Development Tool Configuration (1/2)

(1) When using in-circuit emulator IE-78K0-NS

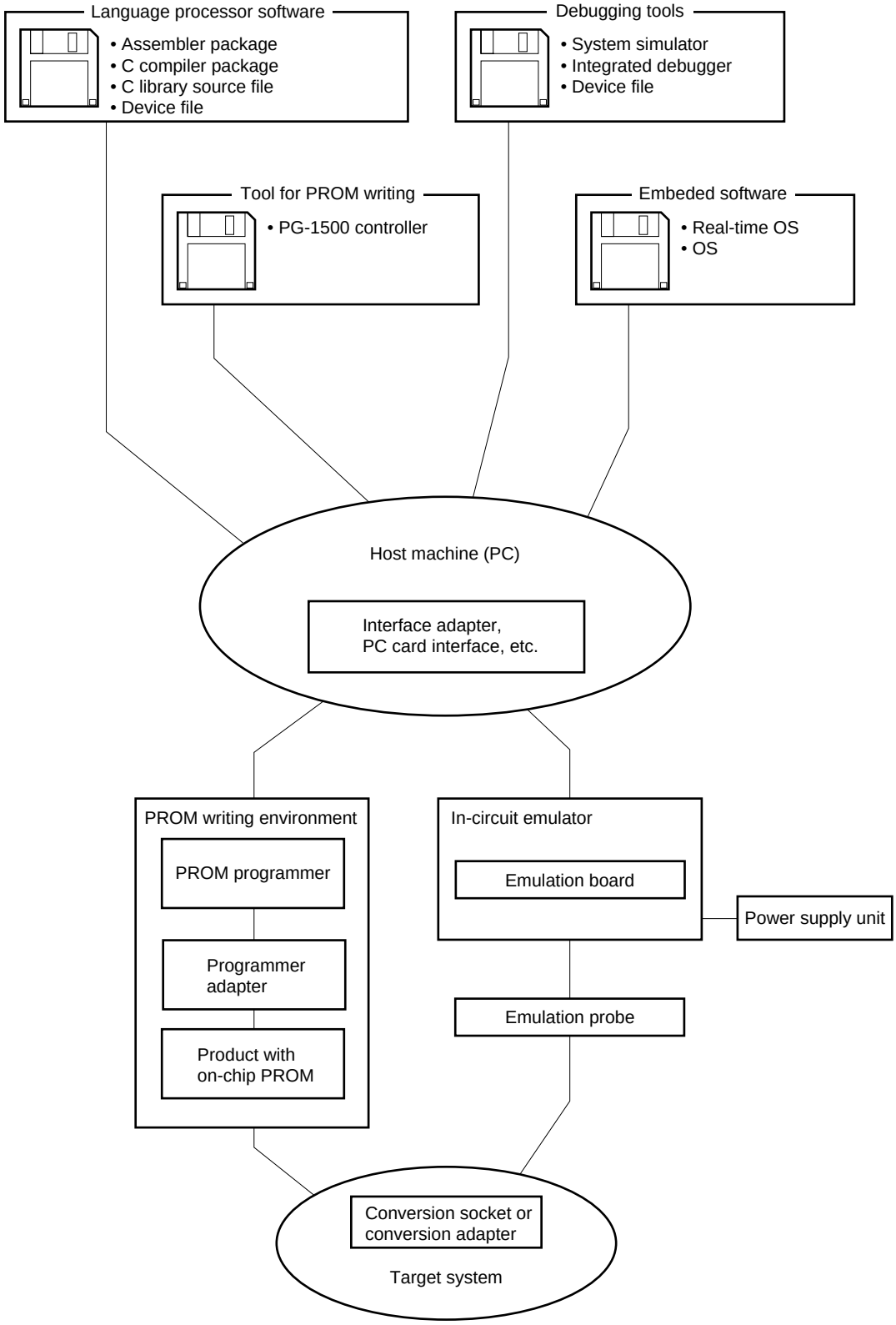
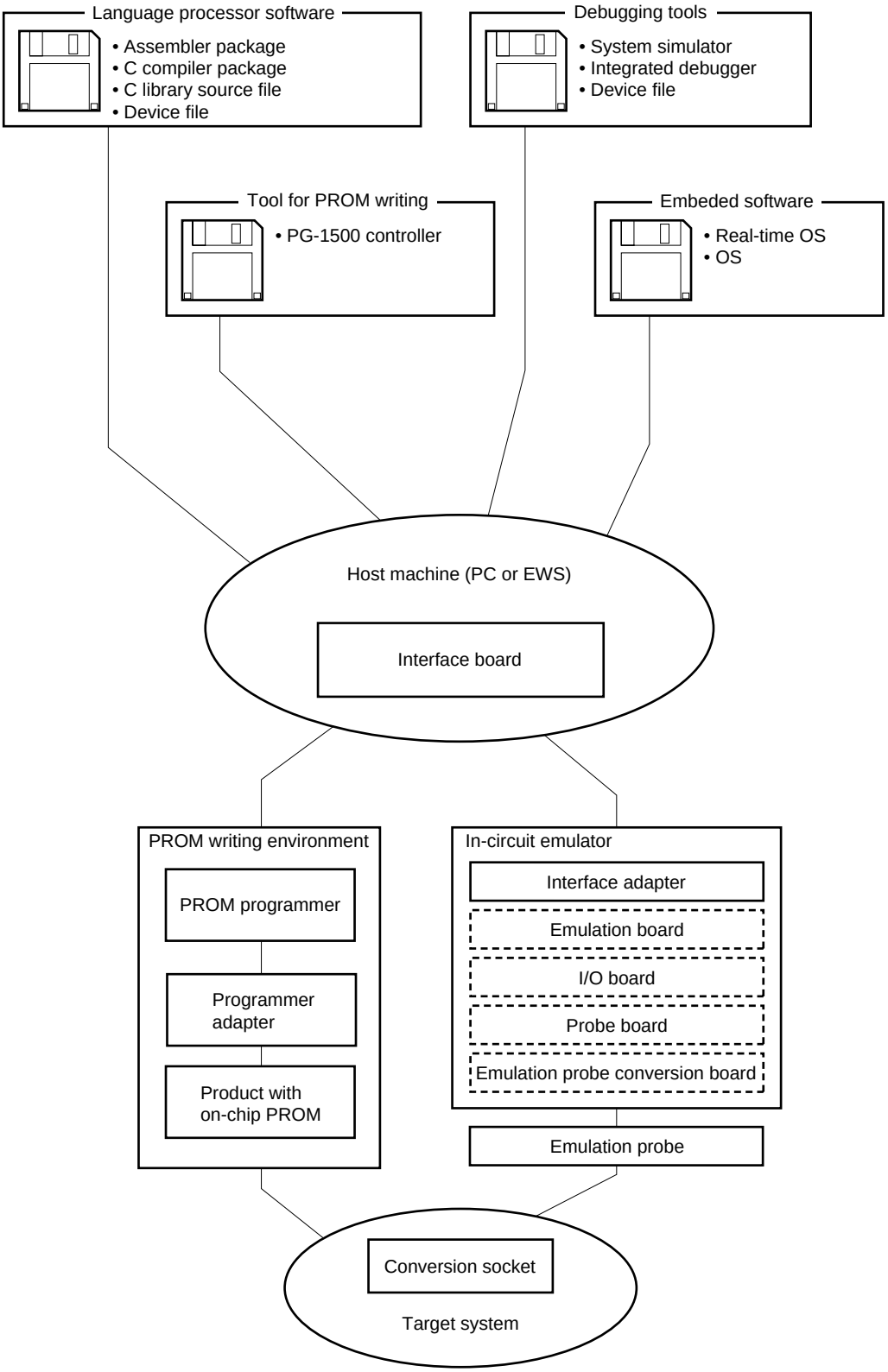


Figure B-1. Development Tool Configuration (2/2)

(2) When using in-circuit emulator IE-78001-R-A



Remark Blocks enclosed by broken line vary depending on the development environment. Refer to **B.3.1 Hardware**.

B.1 Language processing software

RA78K/0 Assembler package	Program that converts program written in mnemonic into object codes that can be executed by microcontroller. In addition, automatic functions to generate symbol table and optimize branch instructions are also provided. Used in combination with optional device file (DF78014). <Precautions when using RA78K/0 under PC environment> This assembler package is a DOS-based application. However, it can also run under Windows environment by using Project Manager (included in the assembler package) on Windows.
	Part number: μS××××RA78K0
CC78K/0 C compiler package	Program that converts program written in C language into object codes that can be executed by microcontroller. Used in combination with optional assembler package and device file . <Precautions when using CC78K/0 under PC environment> This C compiler package is a DOS-based application. However, it can also run under Windows environment by using Project Manager (included in the assembler package) on Windows.
	Part number: μS××××CC78K0
DF78014[Note] Device file	File containing information peculiar to the device. Used in combination with optional tools (RA78K/0, CC78K/0, SM78K0, ID78K0-NS, or ID78K0). Compatible OS and host machine differ depending on tools to be used.
	Part number: μS××××DF78014
CC78K/0-L C library source file	Source program of functions for generating object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Being a source file, its operating environment does not depend on OS.
	Part number: μS××××CC78K0-L

Note DF78014 is common file that can be used with RA78K/0, SM78K0, CC78K/0, ID78K0-NS, and ID78K0.

Remark ×××× in part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××DF78014
μS××××CC78K0-L

××××	Host Machine	OS	Supply Media
AA13	PC-9800 series	Windows (Japanese)[Notes 1, 2]	3.5-inch 2HD FD
AB13	IBM PC/AT™ or compatible	Windows (Japanese)[Notes 1, 2]	3.5-inch 2HC FD
BB13		Windows (English)	
3B16	HP9000 series 700™	HP-UX™ (Rel.9.05)	DAT (DDS)
3K13	SPARCstation™	SunOS™ (Rel.4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS™ (RISC)	NEWS-OS™ (Rel.6.1)	3.5-inch 2HC FD

Notes 1. Can be operated in DOS environment.
2. Does not support Windows NT™

B.2 PROM Writing Tools

B.2.1 Hardware

PG-1500 PROM programmer	PROM programmer that can program PROM-contained single-chip microcontrollers in stand-alone mode or through manipulation from host machine when connected to board supplied as accessory and optional PROM programmer adapter. Can program representative PROMs with 256K-bit to 4M-bit capacities.
PA-78P018CW PA-78P018GC PA-78P018GK PA-78P018KK-S PROM programmer adapter	PROM programmer adapter for μPD78P018F and connected to PG-1500. PA-78P018CW : for 64-pin plastic shrink DIP (750 mils) PA-78P018GC : for 64-pin plastic QFP (14 × 14 mm) PA-78P018GK : for 64-pin plastic LQFP (12 × 12 mm) PA-78P018KK-S : for 64-pin ceramic WQFN (14 × 14 mm)

B.2.2 Software

PG-1500 controller	Connects the PG-1500 to the host machine with a serial or parallel interface and controls the PG-1500 on the host machine. The PG-1500 is a DOS-based application. Therefore, run the PG-1500 in the DOS prompt under Windows.
	Part number: μS××××PG1500

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××PG1500

××××	Host Machine	OS	Supply Media
5A13	PC-9800 series	MS-DOS (Ver.3.30 to Ver.6.2[Note])	3.5-inch 2HD FD
7B13	IBM PC/AT or compatible	Refer to **B.4**.	3.5-inch 2HD FD

Note MS-DOS ver.5.0 or later has a task swap function but this function cannot be used with the above software.

B.3 Debugging Tools

B.3.1 Hardware (1/2)

(1) When using the in-circuit emulator IE-78K0-NS

IE-78K0-NS[Note] In-circuit emulator		The in-circuit emulator serves to debug hardware and software when developing application systems using a 78K/0 Series product. It corresponds to integrated debugger (ID78K0-NS). This emulator should be used in combination with power supply unit, emulation probe, and interface adapter which is required to connect thiis emulator to the host machine.
IE-70000-MC-PS-B Power supply unit		This adapter is used for supplying power from a receptacle of 100-V to 240-V AC.
IE-70000-98-IF-C[Note] Interface adapter		This adapter is required when using the PC-9800 series computer (except notebook type) as the IE-78K0-NS host machine.
IE-70000-CD-IF[Note] PC card Interface		This is PC card and interface cable requied when using the PC-9800 series notebook-type computer as the IE-78K0-NS host machine.
IE-70000-PC-IF-C[Note] Interface adapter		This adapter is required when using the IBM PC and its compatible computers as the IE-78K0-NS host machine.
IE-780034-NS-EM1[Note] Emulation board		This board emulates the operations of the peripheral hardware peculiar to a device. It should be used in combination with an in-circuit emulator.
NP-64CW Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for 64-pin plastic shrink DIP (CW type).
NP-64GC Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for 64-pin plastic QFP (GC-AB8 type).
	EV-9200GC-64 Conversion socket (Refer to **Figures B-2** and **B-3**)	This conversion socket connects the NP-64GC to the target system board designed to mount a 64-pin plastic QFP (GC-AB8 type).
NP-64GK Emulation probe		This probe is used to connect the in-circuit emulator to the target system and is designed for 64-pin plastic QFP (GK-8A8 type).
	TGK-064SBW Conversion adapter (Refer to **Figure B-4**)	This conversion adapter connects the TGK-064SBW to the target system board designed to mount a 64-pin plastic QFP (GK-8A8type).

Note Under development

Remarks 1. NP-64CW, NP-64GC, and NP-64GK are products of Naitou Densei Machidaseisakusho Co., Ltd.
Phone : (044) 822-3813 Naitou Densei Machidaseisakusho Co., Ltd.

2. TGK-064SBW is a product of TOKYO ELETECH CORPORATION.
Inquiries : Daimaru Kougyou Co., Ltd.
Phone : (03) 3820-7112 Tokyo Electronic Component Divsion
(06) 244-6672 Osaka Electronic Component Divsion

3. EV-9200GC-64 is sold in five units.

4. TGK-064SBW is sold in one unit.

B.3.1 Hardware (2/2)

(2) When using the in-circuit emulator IE-78001-R-A

IE-78000-R-A[Note] In-circuit emulator	This in-circuit emulator is used to debug hardware and software when an application system using the 78K/0 Series is developed. It supports the integrated debugger (ID78K0). This emulator is used in combination with an emulation probe and an interface adapter that connects the emulator with the host machine.
IE-70000-98-IF-B or IE-70000-98-IF-C[Note] Interface adapter	Adapter necessary when using the PC-9800 series (except the notebook type) as the host machine of the IE-78001-R-A.
IE-70000-PC-IF-B or IE-70000-PC-IF-C[Note] Interface adapter	Adapter necessary when using IBM PC/AT or compatible machines as the host machine of the IE-78001-R-A.
IE-78000-R-SV3 Interface adapter	Adapter cable necessary when using an EWS as the host machine of the IE-78001-R-A. This cable is connected to the board in the IE-78001-R-A. As Ethernet™, 10Base-5 is supported. If the other methods are used, a commercially available conversion adapter is necessary.
IE-78018-NS-EM1[Note] Emulation probe	This board emulates the peripheral hardware peculiar to a device. It is used in combination with an in-circuit emulator.
IE-78K0-R-EX1[Note] Emulation word processor conversion board	Board necessary when using IE-78018-NS-EM1 is used with IE-78001-R-A.
IE-78014-R-EM-A Emulation board	This board emulates the peripheral hardware peculiar to a device (3.0 to 6.0 V). It is used in combination with IE-78001-R-A.
EP-78240CW-R Emulation probe	This is a probe to connect an in-circuit emulator and target system. It is for a 64-pin plastic shrink DIP (CW type).
EP-78240GC-R Emulation probe	This is a probe to connect an in-circuit emulator and target system. It is for 64-pin plastic QFP (GC-AB8 type).
EV-9200GC-64 Conversion socket (Refer to **Figures B-2** and **B-3**)	This conversion socket connects the board of the target system created for mounting 64-pin plastic QFP (GC-AB8 type) and the EP-78240GC-R. Instead of connecting the EP-78240GC-R, the μPD78P018FKK-S (ceramic WQFN) can also be connected.
EP-78012GK-R Emulation probe	This probe connects an in-circuit emulator and target system. It is for a 64-pin plastic QFP (GK-8A8 type). One 64-pin conversion adapter, TGK-064SBW, which facilitates development of the target system is supplied as an accessory.
TGK-064SBW Conversion adapter (Refer to **Figure B-4**)	This conversion socket connects the board of the target system created for mounting 64-pin plastic QFP (GK-8A8 type) and the EP-78012GK-R. This product is manufactured by TOKYO ELETECH CORPORATION.
EV-9900	Jig used to remove the μPD78P018FKK-S from the EV-9200GC-64.

Note Under development

Remarks 1. TGK-064SBW is a product of TOKYO ELETECH CORPORATION.
Inquiries : Daimaru Kougyou Co., Ltd.
Phone : (03) 3820-7112 Tokyo Electronic Component Divsion
(06) 244-6672 Osaka Electronic Component Divsion

2. EV-9200GC-64 is sold in five units.

3. TGK-064SBW is sold in one unit.

B.3.2 Software (1/2)

SM78K0 System simulator	This simulator simulates the operation of the target system on the host machine and is used to debug the target system at C source level or assembler level. The SM78K0 operates in Windows. By using the SM78K0, the logic and performance of the application can be verified independently of hardware development even if an in-circuit emulator is not used, so that the development efficiency can be enhanced and software quality can be improved. This simulator is used in combination with an optional device file (DF78014).
	Part number: μS××××SM78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM78K0

××××	Host Machine	OS	Supply Media
AA13	PC-9800 series	Windows (Japanese)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT or compatible	Windows (Japanese)[Note]	3.5-inch 2HC FD
BB13		Windows (English)[Note]	

Note Does not support Windows NT

B.3.2 Software (2/2)

ID78K0-NS[Note] Integrated debugger (Supports the in-circuit emulator IE-78K0-NS)	This is a control program that is used to debug the 78K/0 Series. It uses Windows on a personal computer and OSF/Motif™ on EWS as a graphical user interface, and has the appearance and operability conforming to these interfaces. Moreover, debugging functions supporting C language are reinforced, and the trace result can be displayed in C language level by using a window integrating function that associates the source program, disassemble display, and memory display with the trace result. In addition, it can enhance the debugging efficiency of a program using a real-time OS by incorporating function expansion modules such as a task debugger and system performance analyzer. This debugger is used in combination with an optional device file (DF78014).
ID78K0 Integrated debugger (Supports the in-circuit emulator IE-78001-R-A)	
	Part number: μS××××ID78K0-NS, μS××××ID78K0

Note Under development

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID78K0-NS

××××	Host Machine	OS	Supply Media
AA13	PC-9800 series	Windows (Japanese)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT or compatible	Windows (Japanese)[Note]	3.5-inch 2HC FD
BB13		Windows (English)[Note]	

Note Does not support Windows NT

μS××××ID78K0

××××	Host Machine	OS	Supply Media
AA13	PC-9800 series	Windows (Japanese)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT or compatible	Windows (Japanese)[Note]	3.5-inch 2HC FD
BB13		Windows (English)[Note]	
3P16	HP9000 series 700	HP-UX (Rel.9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel.4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel.6.1)	3.5-inch 2HC FD

Note Does not support Windows NT

B.4 OS for IBM PC

The following OSs for the IBM PC are supported.

Table B-1. OS for IBM PC

OS	Version
PC DOS	Ver.5.02 to Ver.6.3
	J6.1/V[Note] to J6.3/V[Note]
IBM DOS™	J5.02/V[Note]
MS-DOS	Ver.5.0 to Ver.6.22
	5.0/V[Note] to 6.2/V[Note]

Note Only English mode is supported.

Caution Although Ver.5.0 or later have a task swap function, this function cannot be used with this software.

B.5 Upgrading Former In-circuit Emulators for 78K/0 Series to IE-78001-R-A

If you have a former in-circuit emulator for the 78K/0 Series (IE-78000-R or IE-78000-R-A), your in-circuit emulator can be upgraded to be equivalent to the in-circuit emulator IE-78000-R-A by only replacing the break board with the IE-78001-R-BK (under development).

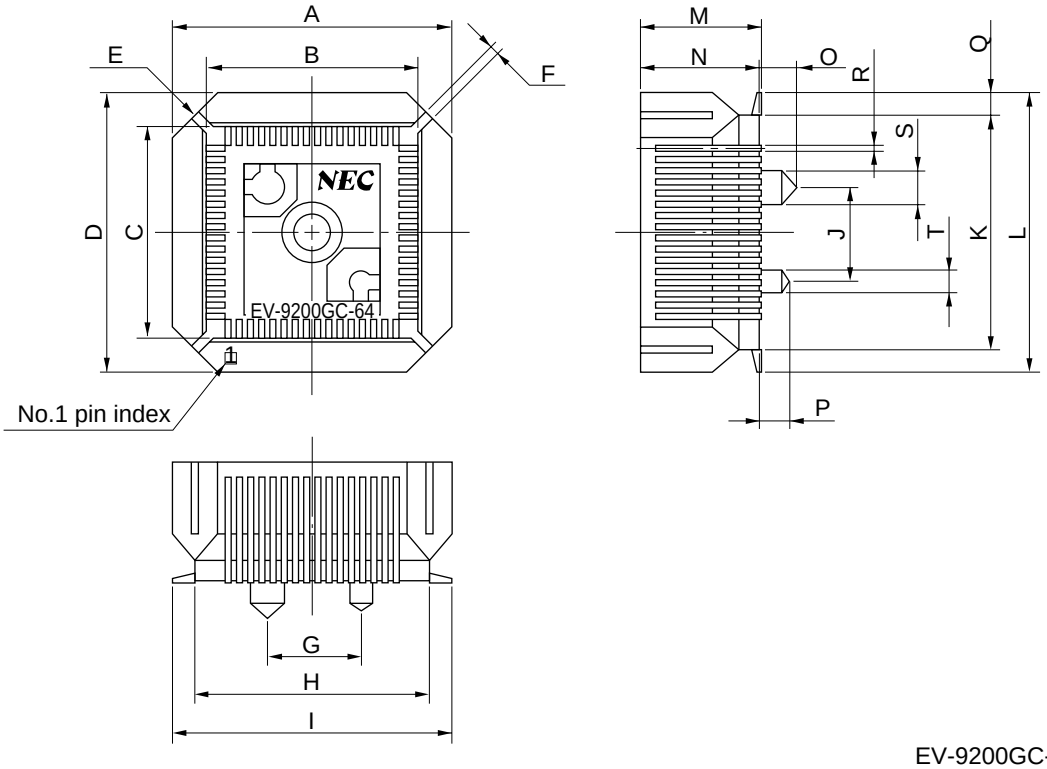
Table B-2. Upgrading Former In-circuit Emulators for 78K/0 Series to IE-78001-R-A

In-circuit Emulator	Cabinet Upgrading[Note]	Board to be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note To upgrade your cabinet, bring it to NEC.

Dimensions of Conversion Socket (EV-9200GC-64) and Recommended Footprint

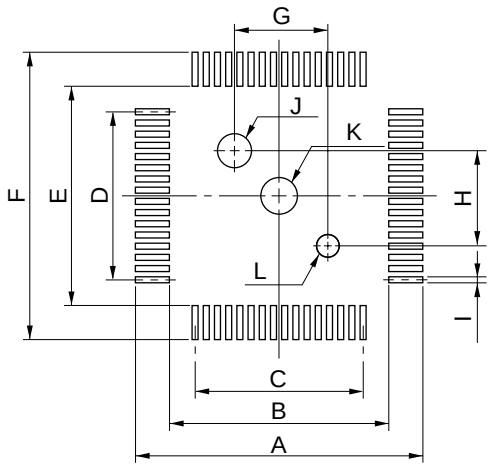
Figure B-2. EV-9200GC-64 Dimensions (For reference only)



EV-9200GC-64-G0

ITEM	MILLIMETERS	INCHES
A	18.8	0.74
B	14.1	0.555
C	14.1	0.555
D	18.8	0.74
E	4-C 3.0	4-C 0.118
F	0.8	0.031
G	6.0	0.236
H	15.8	0.622
I	18.5	0.728
J	6.0	0.236
K	15.8	0.622
L	18.5	0.728
M	8.0	0.315
N	7.8	0.307
O	2.5	0.098
P	2.0	0.079
Q	1.35	0.053
R	0.35±0.1	$0.014^{+0.004}_{-0.005}$
S	ϕ2.3	ϕ0.091
T	ϕ1.5	ϕ0.059

Figure B-3. EV-9200GC-64 Recommended Footprint (For reference only)



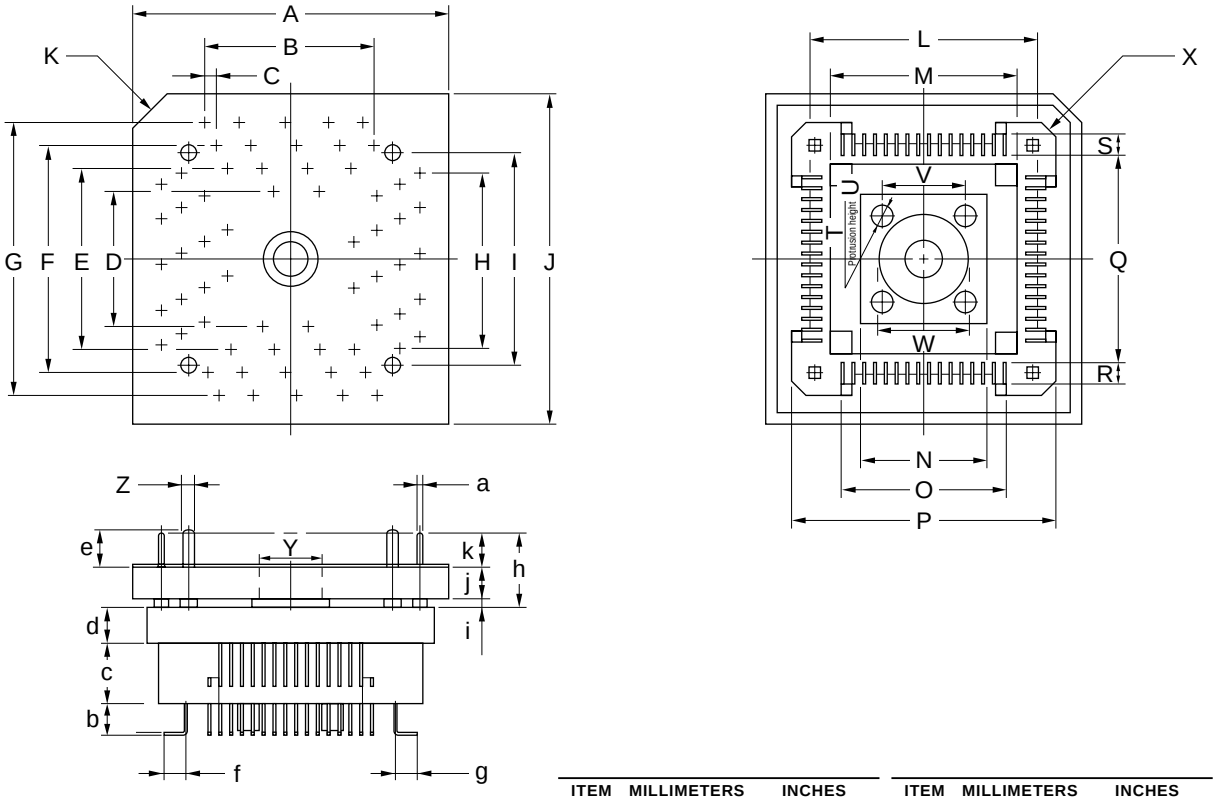
EV-9200GC-64-P1

ITEM	MILLIMETERS	INCHES
A	19.5	0.768
B	14.8	0.583
C	$0.8\pm0.02\times15=12.0\pm0.05$	$0.031^{+0.002}_{-0.001}\times0.591=0.472^{+0.003}_{-0.002}$
D	$0.8\pm0.02\times15=12.0\pm0.05$	$0.031^{+0.002}_{-0.001}\times0.591=0.472^{+0.003}_{-0.002}$
E	14.8	0.583
F	19.5	0.768
G	6.00 ± 0.08	$0.236^{+0.004}_{-0.003}$
H	6.00 ± 0.08	$0.236^{+0.004}_{-0.003}$
I	0.5 ± 0.02	$0.197^{+0.001}_{-0.002}$
J	$\phi2.36\pm0.03$	$\phi0.093^{+0.001}_{-0.002}$
K	$\phi2.2\pm0.1$	$\phi0.087^{+0.004}_{-0.005}$
L	$\phi1.57\pm0.03$	$\phi0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).

Dimensions of Conversion Socket (TGK-064SBW)

Figure B-4. TGK-064SBW Dimensions (For reference only)



ITEM	MILLIMETERS	INCHES
A	18.4	0.724
B	0.65x15=9.75	0.026x0.591=0.384
C	0.65	0.026
D	7.75	0.305
E	10.15	0.400
F	12.55	0.494
G	14.95	0.589
H	0.65x15=9.75	0.026x0.591=0.384
I	11.85	0.467
J	18.4	0.724
K	C 2.0	C 0.079
L	12.45	0.490
M	10.25	0.404
N	7.7	0.303
O	10.02	0.394
P	14.92	0.587
Q	11.1	0.437
R	1.45	0.057
S	1.45	0.057
T	4-ϕ1.3	4-ϕ0.051
U	1.8	0.071
V	5.0	0.197
W	ϕ5.3	ϕ0.209
X	4-C 1.0	4-C 0.039
Y	ϕ3.55	ϕ0.140
Z	ϕ0.9	ϕ0.035

ITEM	MILLIMETERS	INCHES
a	ϕ0.3	ϕ0.012
b	1.85	0.073
c	3.5	0.138
d	2.0	0.079
e	3.9	0.154
f	1.325	0.052
g	1.325	0.052
h	5.9	0.232
i	0.8	0.031
j	2.4	0.094
k	2.7	0.106

TGK-064SBW-G1E

Remark Product of TOKYO ELETECH CORPORATION.

[MEMO]

★

APPENDIX C EMBEDDED SOFTWARE

For efficient program development and maintenance of μPD78014H Subseries, the following embedded softwares are available.

Real-time OS (1/2)

RX78K/0 Real-time OS	RX78K/0 is real-time OS conforming to μITRON specifications. Tool (configurator) that generates nucleus of RX78K/0 and plural information tables is supplied. Used in combination with an optional assembler package (RA78K/0) and device file (DF78014). <Precautions when using RX78K/0 under PC environment> RX78K/0 is a DOS-based application. Therefore run the RX78K/0 from the DOS prompt under Windows.
	Part number: μS××××RX78013-△△△△

Caution When purchasing the RX78K/0, fill an application form and conclude the contract for use permission in advance.

Remark The part numbers ×××× and △△△△ differ depending on the host machine and OS used.

μS××××RX78013-△△△△

△△△△	Product Outline	Upper Limit of Quantity for Mass Production
001	Evaluation object	Do not use for mass-produced product
100K	Object for mass-produced product	0.1 million
001M		1 million
010M		10 million
S01	Source program	Source program for mass-produced object

××××	Host Machine	OS	Supply Media
AA13	PC-9800 series	Windows (Japanese)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT or compatible	Windows (Japanese)[Note]	3.5-inch 2HC FD
BB13		Windows (English)[Note]	
3P16	HP9000 series 700	HP-UX (Rel.9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel.4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel.6.1)	3.5-inch 2HC FD

Notes 1. Can be operated in DOS environment.

2. Does not support Windows NT.

Real-time OS (2/2)

MX78K0 OS	μITRON-specification subset OS. Nucleus of MX78K0 is supplied. This OS performs task management, event management, and time management. It controls the task execution sequence for task management and selects the task to be executed next. <Precautions when using MX78K0 under PC environment> MX78K0 is a DOS-based application. Therefore run the MX78K0 from the DOS prompt under Windows.
	Part number: μS××××MX78K0-△△△

Remark ×××× and △△△ in the part number differ depending on the host machine and OS used.

μS××××MX78K0-△△△

△△△	Product Outline	Note
001	Evaluation object	Use for trial product
××	Object for mass-produced product	Use for mass-produced product
S01	Source program	Can be purchased only when object for mass-produced product is purchased

××××	Host Machine	OS	Supply Media
AA13	PC-9800 series	Windows (Japanese)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT or compatible	Windows (Japanese)[Note]	3.5-inch 2HC FD
BB13		Windows (English)[Note]	
3P16	HP9000 series 700	HP-UX (Rel.9.05)	DAT (DDS)
3K13	SPARCstation	SunOS (Rel.4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS (RISC)	NEWS-OS (Rel.6.1)	3.5-inch 2HC FD

Notes 1. Can be operated in DOS environment.

2. Does not support Windows NT.

[MEMO]

APPENDIX D REGISTER INDEX

D.1 Register Index (By Name, in Alphabetical Order)

[S]

[T]

[W]

D.2 Register Index (By Symbol, in Alphabetical Order)

★ # APPENDIX E REVISION HISTORY

The revision history made is shown as follows. Applied chapter shows the chapter to which the revision is made in each edition.

Edition	Major Revisions from Previous Edition	Applied Chapter
Second	The following block diagrams were changed. • **Figure 4-6 Block Diagram of P20, P21, and P23 to P26** • **Figure 4-7 Block Diagram of P22 and P27** • **Figure 4-8 Block Diagram of P30 to P37**	CHAPTER 4 PORT FUNCTIONS
	Table 5-2 Relationship between CPU Clock and Minimum Instruction Execution Time was added.	CHAPTER 5 CLOCK GENERATION CIRCUIT
	Figure 7-10 and **7-13 Square Wave Output Operation Timings** were added.	CHAPTER 7 8-BIT TIMER/ EVENT COUNTER
	Condition to release (output ready signal) busy mode of serial interface channel 0 was changed.	CHAPTER 13 SERIAL INTERFACE CHANNEL 0
	Caution in wiring was added to **13.4.3 (2) (a) Bus release signal (REL)**, **(b) Command signal (CMD)**.	
	Entirely revised: Compatible with in-circuit emulator IE-78K0-NS and IE-78001-R-A.	APPENDIX B DEVELOPMENT TOOLS
	Entirely revised: Fussy Interface Development Support System was deleted.	APPENDIX C EMBEDDED SOFTWARE

[MEMO]

Phase-out/Discontinued

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑