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

Phase-out/Discontinued

μPD78018F, 78018FY Subseries

8-bit Single-chip Microcontroller

μPD78011F
μPD78012F
μPD78013F
μPD78014F
μPD78015F
μPD78016F
μPD78018F
μPD78P018F

μPD78011FY
μPD78012FY
μPD78013FY
μPD78014FY
μPD78015FY
μPD78016FY
μPD78018FY
μPD78P018FY

μPD78011F(A)
μPD78012F(A)
μPD78013F(A)
μPD78014F(A)
μPD78015F(A)
μPD78016F(A)
μPD78018F(A)
μPD78P018F(A)
μPD78012F(A2)

Document No. U10659EJ5V0UM00 (5th edition)
Date Published January 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.

FIP, EEPROM, IEBus, and QTOP are trademarks of NEC Corp.
MS-DOS, Windows, and WindowsNT are trademarks of Microsoft Corp.
IBM DOS, PC/AT, and PC DOS are trademarks of IBM Corp.
HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Co.
SPARCstation is a trademark of SPARC International, Inc.
SunOS is a trademark of Sun Microsystems, Inc.
Ethernet is a trademark of Xerox Corp.
NEWS, NEWS-OS are trademarks of Sony Corp.
OSF/Motif is a trademark of Open Software Foundation, Inc.
TRON is an abbreviation of the Realtime Operating system Nucleus.
ITRON is an abbreviation of Industrial TRON.

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, 78P018FYDW, 78P018FYKK-S
The customer must judge the need for license	:	μPD78011FCW-XXX, 78011FGC-XXX-AB8, 78011FGK-XXX-8A8
		μPD78012FCW-XXX, 78012FGC-XXX-AB8, 78012FGK-XXX-8A8
		μPD78013FCW-XXX, 78013FGC-XXX-AB8, 78013FGK-XXX-8A8
		μPD78014FCW-XXX, 78014FGC-XXX-AB8, 78014FGK-XXX-8A8
		μPD78015FCW-XXX, 78015FGC-XXX-AB8, 78015FGK-XXX-8A8
		μPD78016FCW-XXX, 78016FGC-XXX-AB8, 78016FGK-XXX-8A8
		μPD78018FCW-XXX, 78018FGC-XXX-AB8, 78018FGK-XXX-8A8
		μPD78P018FCW, 78P018FGC-AB8, 78P018FGK-8A8
		μPD78011FCW(A)-XXX, 78011FGC(A)-XXX-AB8
		μPD78012FCW(A)-XXX, 78012FGC(A)-XXX-AB8, 78012FGC(A2)-XXX-AB8
		μPD78013FCW(A)-XXX, 78013FGC(A)-XXX-AB8
		μPD78014FCW(A)-XXX, 78014FGC(A)-XXX-AB8
		μPD78015FCW(A)-XXX, 78015FGC(A)-XXX-AB8
		μPD78016FCW(A)-XXX, 78016FGC(A)-XXX-AB8
		μPD78018FCW(A)-XXX, 78018FGC(A)-XXX-AB8
		μPD78P018FCW(A), 78P018FGC(A)-AB8
		μPD78011FYCW-XXX, 78011FYGC-XXX-AB8, 78011FYGK-XXX-8A8
		μPD78012FYCW-XXX, 78012FYGC-XXX-AB8, 78012FYGK-XXX-8A8
		μPD78013FYCW-XXX, 78013FYGC-XXX-AB8
		μPD78014FYCW-XXX, 78014FYGC-XXX-AB8, 78014FYGK-XXX-8A8
		μPD78015FYCW-XXX, 78015FYGC-XXX-AB8
		μPD78016FYCW-XXX, 78016FYGC-XXX-AB8
		μPD78018FYCW-XXX, 78018FYGC-XXX-AB8
		μPD78P018FYCW, 78P018FYGC-AB8, 78P018FYGK-8A8

Phase-out/Discontinued

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

The information in this document is subject to change without notice.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated “quality assurance program“ for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

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

Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

Anti-radioactive design is not implemented in this product.

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

Throughout	Description
p.46, 61	**1.6, 2.6 Product Development of 78K/0 Series:** Addition of the following models • μPD780018AY subseries • μPD780988 subseries • μPD78098B subseries • μPD780973 subseries Deletion of the following models • μPD78075BY subseries • μPD780018 subseries • μPD780018Y subseries
p.140-143	Change of **Figures 6-6, 6-8 Block Diagram of P20, P21, and P23-P26**, **Figures 6-7, 6-9 Block Diagram of P22 and P27**
p.173	Addition of **8.1 Outline of Timers in μPD780018F, 780018FY Subseries**
p.214, 218	Addition of Figures **9-10, 9-13 Square Wave Output Operation**
p.247	Addition of Caution to **14.2 (6) ANI0-ANI7 pins**
p.248	Addition of Caution to **14.2 (7) AV$_{\text{REF}}$ pin**
p.268	Change of Note on BSYE flag in **Figure 15-4 Format of Serial Bus Interface Control Register**
p.280, 302	Addition of Caution to **15.4.3 (2) (a) Bus release signal (REL)**, **(b) Command signal (CMD)**, and **(11) Notes on SBI mode**
p.495	**APPENDIX B DEVELOPMENT TOOLS** Throughout: Support for in-circuit emulator IE-78K0-NS
p.509	**APPENDIX C EMBEDDED SOFTWARE** Throughout: Deletion of fuzzy inference development support system

The mark ★ shows major revised points.

[MEMO]

INTRODUCTION

Readers

This manual is intended for user engineers who understand the functions of the μPD78018F and μPD78018FY subseries and wish to design and develop its application systems and programs.
Target products are as follows:

• μPD78018F subseries:	μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F, 78P018F μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), 78P018F(A), 78012F(A2)
• μPD78018FY subseries:	μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, 78018FY, 78P018FY

Caution Of the μPD78018F and 78018FY subseries, the μPD78P018FDW, 78P018FKK-S, 78P018FYDW, and 78P018FYKK-S do not have a reliability intended for mass production of your systems. Use these models for experiment or function evaluation only.

Purpose

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

Organization

Two manuals are available for the μPD78018F and μPD78018FY subseries: this manual and Instruction Manual (common to the 78K/0 series).

μPD78018F, 78018FY 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	

How to Read This Manual It is assumed that the readers of this manual have general knowledge on electric engineering, logic circuits, and microcomputers.

□ **If this manual is used as the manual of the μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), or 78P018F(A)**

→ Unless otherwise specified, the μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F, and 78P018F are treated as the representative models in this manual. If you use the μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), or 78P018F(A), take the μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F, and 78P018F as the μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), and 78P018F(A), respectively.

□ **If this manual is used as the manual of the μPD78012F(A2)**

→ Unless otherwise specified, the μPD78012F is treated as the representative model of the μPD78012F(A) and μPD78012F(A2). If you use the μPD78012F(A2), take the μPD78012F as the μPD78012F(A2).

□ To understand the overall functions of the μPD78018F and μPD78018FY 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 D REGISTER INDEX**.

□ To learn the differences with the μPD78014 and 78014H subseries

→ Refer to **APPENDIX A DIFFERENCES BETWEEN μPD78014, 78014H, AND 78018F SUBSERIES.**

□ To learn the details of the instruction functions of the μPD78018F and μPD78018FY subseries

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

□ To check the electrical characteristics of the μPD78018F and μPD78018FY subseries,

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

□ For the application examples of the respective functions of the μPD78018F and μPD78018FY subseries,

→ Refer to the **Application Notes** separately available.

Caution The examples in this manual are for the "standard" quality grade of general-purpose electronic systems. If you use an example in this manual for applications where "special" quality grade is required, evaluate the quality of the parts and circuits actually used.

Chapter Organization

The functions that differ between the μPD78018F subseries and μPD78018FY subseries are explained in separate chapters in this manual. The chapters corresponding to each subseries are shown below.
Refer to the chapter marked ○ in the following table.

Chapter	μPD78018F Subseries	μPD78018FY Subseries
CHAPTER 1 GENERAL (μPD78018F SUBSERIES)	○	–
CHAPTER 2 GENERAL (μPD78018FY SUBSERIES)	–	○
CHAPTER 3 PIN FUNCTIONS (μPD78018F SUBSERIES)	○	–
CHAPTER 4 PIN FUNCTIONS (μPD78018FY SUBSERIES)	–	○
CHAPTER 5 CPU ARCHITECTURE	○	○
CHAPTER 6 PORT FUNCTIONS	○	○
CHAPTER 7 CLOCK GENERATION CIRCUIT	○	○
CHAPTER 8 16-BIT TIMER/EVENT COUNTER	○	○
CHAPTER 9 8-BIT TIMER/EVENT COUNTER	○	○
CHAPTER 10 WATCH TIMER	○	○
CHAPTER 11 WATCHDOG TIMER	○	○
CHAPTER 12 CLOCK OUTPUT CONTROL CIRCUIT	○	○
CHAPTER 13 BUZZER OUTPUT CONTROL CIRCUIT	○	○
CHAPTER 14 A/D CONVERTER	○	○
CHAPTER 15 SERIAL INTERFACE CHANNEL 0 (μPD78018F SUBSERIES)	○	–
CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD78018FY SUBSERIES)	–	○
CHAPTER 17 SERIAL INTERFACE CHANNEL 1	○	○
CHAPTER 18 INTERRUPT FUNCTIONS AND TEST FUNCTIONS	○	○
CHAPTER 19 EXTERNAL DEVICE EXTENSION FUNCTION	○	○
CHAPTER 20 STANDBY FUNCTION	○	○
CHAPTER 21 RESET FUNCTION	○	○
CHAPTER 22 ROM CORRECTION	○	○
CHAPTER 23 μPD78P018F, 78P018FY	○	○
CHAPTER 24 INSTRUCTION SET	○	○

Differences Between μPD78018F Subseries and μPD78018FY Subseries

The μPD78018F subseries and μPD78018FY subseries differ from each other in some parts of serial interface channel 0.

Modes of Serial Interface Channel 0	μPD78018F Subseries	μPD78018FY Subseries
3-wire serial I/O mode	○	○
2-wire serial I/O mode	○	○
SBI (serial bus interface) mode	○	–
I^{2}C (Inter IC) bus mode	–	○

○: Supported

– : Not supported

Legend			
	Data significance	:	Left: higher digit, right: lower digit
	Active low	:	$\overline{\text{XXX}}$ (top bar over pin or signal name)
	Note	:	Explanation of the text marked **Note**
	Caution	:	Important information
	Remark	:	Supplement
	Numerical representation	:	Binary ... XXXX or XXXXB
			Decimal ... XXXX
			Hexadecimal ... XXXXH

★ **Related Documents**

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

- **Device-related documents**

Document Name		Document Number	
		Japanese	English
μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F Data Sheet		U10280J	U10280E
μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, 78018FY Data Sheet		U10281J	U10281E
μPD78P018F Data Sheet		U10955J	U10955E
μPD78P018FY Data Sheet		U10989J	U10989E
μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A) Data Sheet		U11921J	U11921E
μPD78P018F(A) Data Sheet		U12132J	U12132E
μPD78018F, 78018FY Subseries User's Manual		U10659J	This manual
78K/0 Series User's Manual - Instruction		U12326J	U12326E
78K/0 Series Instruction Table		U10903J	–
78K/0 Series Instruction Set		U10904J	–
μPD78018F Subseries Special Function Register Table		IEM-5594	–
μPD78018FY Subseries Special Function Register Table		U10287J	–
78K/0 Series Application Note	Basic (I)	U12704J	U12704E
	Floating-point Program	IEA-718	IEA-1289

Caution The contents of the above related documents are subject to change without notice. Be sure to use the latest edition in designing your system.

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

Document Name		Document Number	
		Japanese	English
RA78K0 Assembler Package	Operation	U11802J	U11802E
	Assembly Language	U11801J	U11801E
	Structured Assembly Language	U11789E	U11789E
RA78K Series Structured Assembler Preprocessor		U12323J	EEU-1402
CC78K0 C Compiler	Operation	U11517J	U11517E
	Language	U11518J	U11518E
CC78K/0 C Compiler Application Note	Programming Know-how	U13034J	EEA-1208
CC78K Series Library Source File		U12322J	–
PG-1500 PROM Programmer		U11940J	EEU-1335
PG-1500 Controller PC-9800 Series (MS-DOS™) Base		EEU-704	EEU-1291
PG-1500 Controller IBM PC Series (PC DOS™) Base		EEU-5008	U10540E
IE-78K0-NS		Planned	Planned
IE-78001-R-A		Planned	Planned
IE-78K0-R-EX1		Planned	Planned
IE-78018-NS-EM1		Planned	Planned
IE-78014-R-EM-A		EEU-962	EEU-1487
EP-78240		EEU-986	U10332E
EP-78012GK-R		EEU-5012	EEU-1538
SM78K0 System Simulator Windows™ Base	Reference	U10181J	U10181E
SM78K Series System Simulator	External Part User Open Interface Specifications	U10092J	U10092E
ID78K0-NS Integrated Debugger	Reference	U12900J	Planned
ID78K/0 Integrated Debugger EWS Base	Reference	U11151J	–
ID78K0 Integrated Debugger PC Base	Reference	U11539J	U11539E
ID78K0 Integrated Debugger Windows Base	Guide	U11649J	U11649E

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

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

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

- **Other related documents**

Document Name	Document Number	
	Japanese	English
IC Package Manual	C10943X	
Semiconductor Device Mounting Technology Manual	C10535J	C10535E
Quality Grades on NEC Semiconductor Devices	C11531J	C11531E
Reliability and Quality Control of NEC Semiconductor Devices	C10983J	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892J	C11892E
Semiconductor Quality/Reliability Handbook	C11893J	MEI-1202
Microcomputer Related Product Guide - Other Manufactures	U11416J	–

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

[MEMO]

TABLE OF CONTENTS

LIST OF FIGURES (1/7)

LIST OF FIGURES (2/7)

LIST OF FIGURES (3/7)

LIST OF FIGURES (4/7)

Figure No. Title Page

LIST OF FIGURES (5/7)

Figure No. Title Page

LIST OF FIGURES (6/7)

LIST OF FIGURES (7/7)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

Phase-out/Discontinued

[MEMO]

CHAPTER 1 GENERAL (μPD78018F SUBSERIES)

1.1 Features

- High-capacity ROM and RAM

Item Part Number	Program Memory (ROM)	Data Memory		
		Internal High-Speed RAM	Internal Extension RAM	Internal Buffer RAM
μPD78011F	8 KB	512 B	—	32 B
μPD78012F	16 KB			
μPD78013F	24 KB	1024 B		
μPD78014F	32 KB			
μPD78015F	40 KB		512 B	
μPD78016F	48 KB			
μPD78018F	60 KB		1024 B	
μPD78P018F	60 KB[Note 1]	1024 B[Note 2]	1024 B[Note 3]	

Notes **1.** 8, 16, 24, 32, 40, 48, or 60 KB is selectable by using memory size select register (IMS).

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

3. 0, 512, or 1024 B is selectable by using internal extension RAM size select register (IXS).

- External memory extension space: 64 KB
- 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 mode : 1 channel
  - 3-wire serial I/O mode (with automatic transmit/receive 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

- **μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F, 78P018F**
  Telephones, VCRs, audio sets, cameras, home appliances, etc.

- **μPD78011F(A), 78012F(A), 78012F(A2), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), 78P018F(A)**
  Control unit of automotive appliances, gas leak breaker, safety devices, etc.

1.3 Ordering Information

(1) Standard grade products (including not-applicable products)

Part Number	Package	Internal ROM
μPD78011FCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78011FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78011FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78012FCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78012FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78012FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78013FCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78013FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78013FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78014FCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78014FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78014FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78015FCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78015FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78015FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78016FCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78016FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78016FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78018FCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78018FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78018FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78P018FCW	64-pin plastic shrink DIP (750 mil)	One-time PROM
μPD78P018FDW	64-pin ceramic shrink DIP (with window) (750 mil)	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

Remark XXX indicates ROM code suffix.

(2) Special grade products

Part Number	Package	Internal ROM
μPD78011FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78011FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78012FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78012FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78013FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78013FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78014FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78014FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78015FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78015FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78016FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78016FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78018FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78018FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78P018FCW(A)	64-pin plastic shrink DIP (750 mil)	One-time PROM
μPD78P018FGC(A)-AB8	64-pin plastic QFP (14 × 14 mm)	One-time PROM
μPD78P012FGC(A2)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM

Remark XXX indicates ROM code suffix.

1.4 Quality Grade

(1) Standard grade products (including not-applicable products)

Part Number	Package	Quality Grade
μPD78011FCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78011FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78011FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78012FCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78012FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78012FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78013FCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78013FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78013FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78014FCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78014FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78014FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78015FCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78015FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78015FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78016FCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78016FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78016FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78018FCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78018FGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78018FGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78P018FCW	64-pin plastic shrink DIP (750 mil)	Standard
μPD78P018FDW	64-pin ceramic shrink DIP (with window) (750 mil)	Not applicable (for function evaluation)
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 (for function evaluation)

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 XXX indicates ROM code suffix.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

(2) Standard grade products

Part Number	Package	Quality Grade
μPD78011FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Special
μPD78011FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78012FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Special
μPD78012FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78013FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Special
μPD78013FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78014FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Special
μPD78014FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78015FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Special
μPD78015FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78016FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Special
μPD78016FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78018FCW(A)-XXX	64-pin plastic shrink DIP (750 mil)	Special
μPD78018FGC(A)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78P018FCW(A)	64-pin plastic shrink DIP (750 mil)	Special
μPD78P018FGC(A)-AB8	64-pin plastic QFP (14 × 14 mm)	Special
μPD78012FGC(A2)-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Special

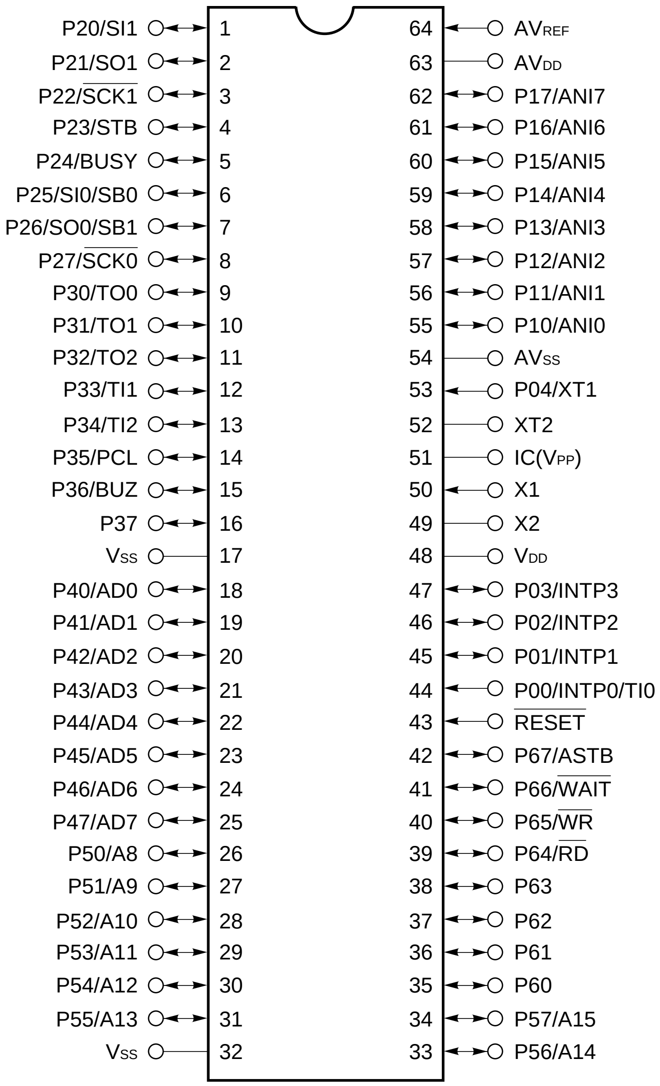
Remark XXX indicates ROM code suffix.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. 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)

(1) Normal operation mode

- **64-pin plastic shrink DIP (750 mil)**
  μPD78011FCW-XXX, 78012FCW-XXX, 78013FCW-XXX,
  μPD78014FCW-XXX, 78015FCW-XXX, 78016FCW-XXX,
  μPD78018FCW-XXX, 78P018FCW
  μPD78011FCW(A)-XXX, 78012FCW(A)-XXX, 78013FCW(A)-XXX,
  μPD78014FCW(A)-XXX, 78015FCW(A)-XXX, 78016FCW(A)-XXX,
  μPD78018FCW(A)-XXX, 78P018FCW(A)
- **64-pin ceramic shrink DIP (with window) (750 mil)**
  μPD78P018FDW



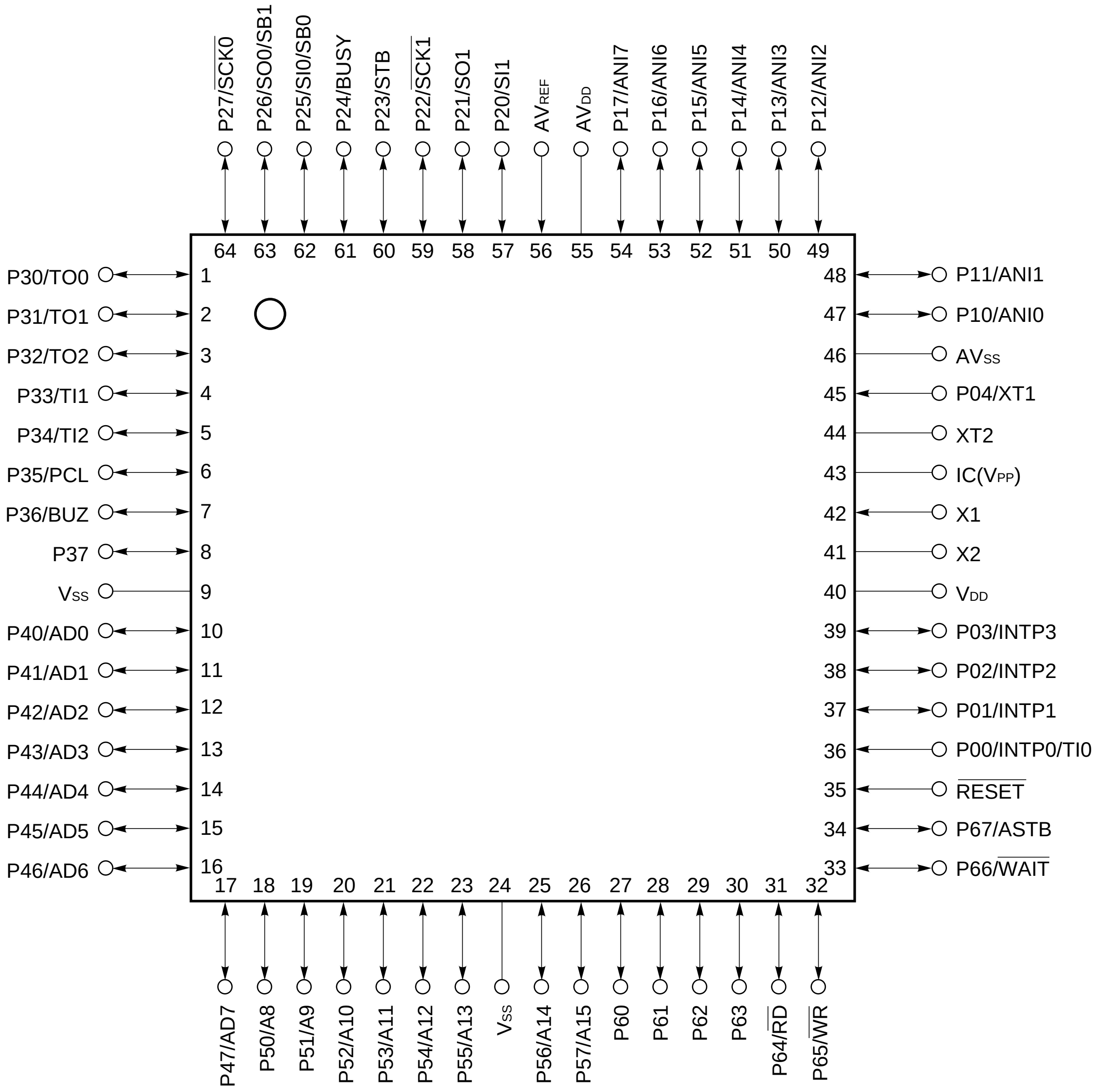
Cautions 1. Directly connect the IC (Internally Connected) pins to V_{SS}.

2. Connect the AV_{DD} pin to V_{DD}.

3. Connect the AV_{SS} pin to V_{SS}.

Remark ( ): μPD78P018F

- **64-pin plastic QFP (14 × 14 mm)**
  μPD78011FGC-XXX-AB8, 78012FGC-XXX-AB8, 78013FGC-XXX-AB8,
  μPD78014FGC-XXX-AB8, 78015FGC-XXX-AB8, 78016FGC-XXX-AB8,
  μPD78018FGC-XXX-AB8, 78P018FGC-AB8
  μPD78011FGC(A)-XXX-AB8, 78012FGC(A)-XXX-AB8, 78013FGC(A)-XXX-AB8,
  μPD78014FGC(A)-XXX-AB8, 78015FGC(A)-XXX-AB8, 78016FGC(A)-XXX-AB8,
  μPD78018FGC(A)-XXX-AB8, 78P018FGC(A)-AB8, 78012FGC(A2)-XXX-AB8
- **64-pin plastic LQFP (12 × 12 mm)**
  μPD78011FGC-XXX-8A8, 78012FGK-XXX-8A8, 78013FGK-XXX-8A8,
  μPD78014FGK-XXX-8A8, 78015FGK-XXX-8A8, 78016FGK-XXX-8A8,
  μPD78018FGK-XXX-8A8, 78P018FGK-8A8
- **64-pin ceramic WQFN (14 × 14 mm)**
  μPD78P018FKK-S



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

2. Connect the AV_{DD} pin to V_{DD}.

3. Connect the AV_{SS} pin to V_{SS}.

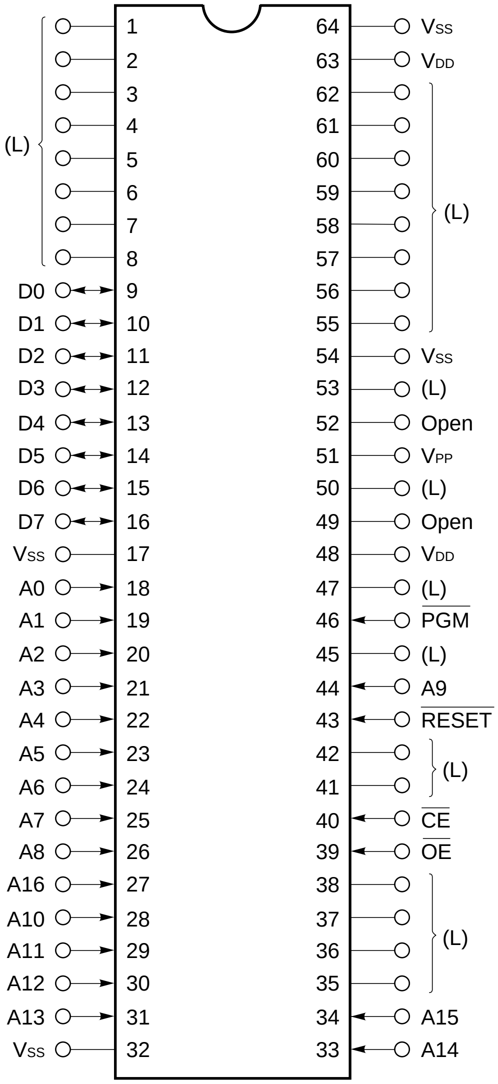
Remark ( ): μPD78P018F

A8-A15	:	Address Bus	PCL	:	Programmable Clock
AD0-AD7	:	Address/Data Bus	$\overline{RD}$	:	Read Strobe
ASTB	:	Address Strobe	$\overline{RESET}$	:	Reset
ANI0-ANI7	:	Analog Input	SB0, SB1	:	Serial Bus
AV_{DD}	:	Analog Power Supply	$\overline{SCK0}$, $\overline{SCK1}$	:	Serial Clock
AV_{REF}	:	Analog Reference Voltage	SI0, SI1	:	Serial Input
AV_{SS}	:	Analog Ground	SO0, SO1	:	Serial Output
BUSY	:	Busy	STB	:	Strobe
BUZ	:	Buzzer Clock	TI0-TI2	:	Timer Input
IC	:	Internally Connected	TO0-TO2	:	Timer Output
INTP0-INTP3	:	Interrupt from Peripherals	$\overline{WAIT}$	:	Wait
P00-P04	:	Port 0	$\overline{WR}$	:	Write Strobe
P10-P17	:	Port 1	X1, X2	:	Crystal (Main System Clock)
P20-P27	:	Port 2	XT1, XT2	:	Crystal (Subsystem Clolck)
P30-P37	:	Port 3	V_{DD}	:	Power Supply
P40-P47	:	Port 4	V_{PP}	:	Programming Power Supply
P50-P57	:	Port 5	V_{SS}	:	Ground
P60-P67	:	Port 6			

Remark V_{PP} is for the μPD78P018F. It is replaced by an IC pin for the mask ROM model.

(2) PROM programming mode

- **64-pin plastic shrink DIP (750 mil)**
  μPD78P018FCW, 78P018FCW(A)
- **64-pin ceramic shrink DIP (with window) (750 mil)**
  μPD78P018FDW



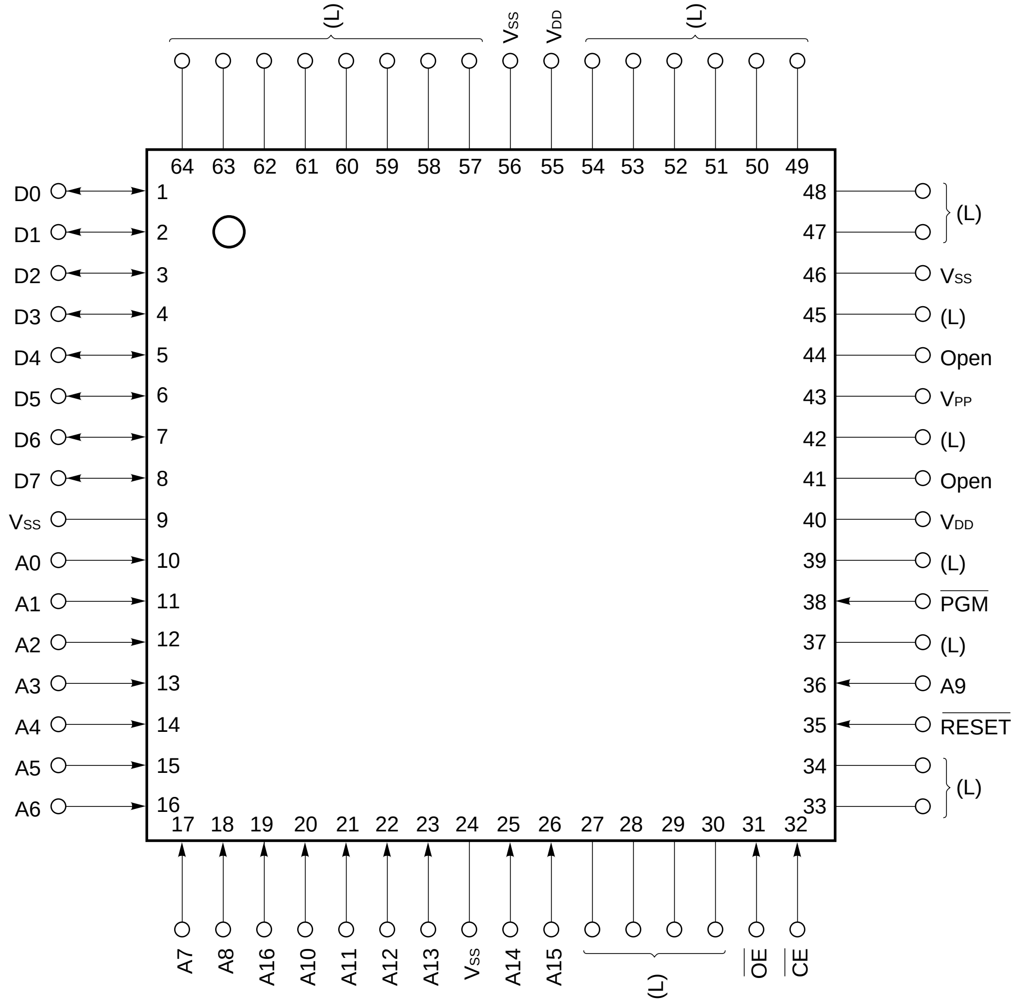
Cautions 1. (L) : Individually connect this pin to V_{SS} via a pull-down resistor (10 kΩ).

2. V_{SS} : Connect this pin to ground.

3. $\overline{\text{RESET}}$: Fix this pin to the low level.

4. Open : Leave this pin unconnected.

- **64-pin plastic QFP (14 × 14 mm)**
  μPD78P018FGC-AB8, 78P018FGC(A)-AB8
- **64-pin plastic LQFP (12 × 12 mm)**
  μPD78P018FGK-8A8
- **64-pin ceramic WQFN (14 × 14 mm)**
  μPD78P018FKK-S



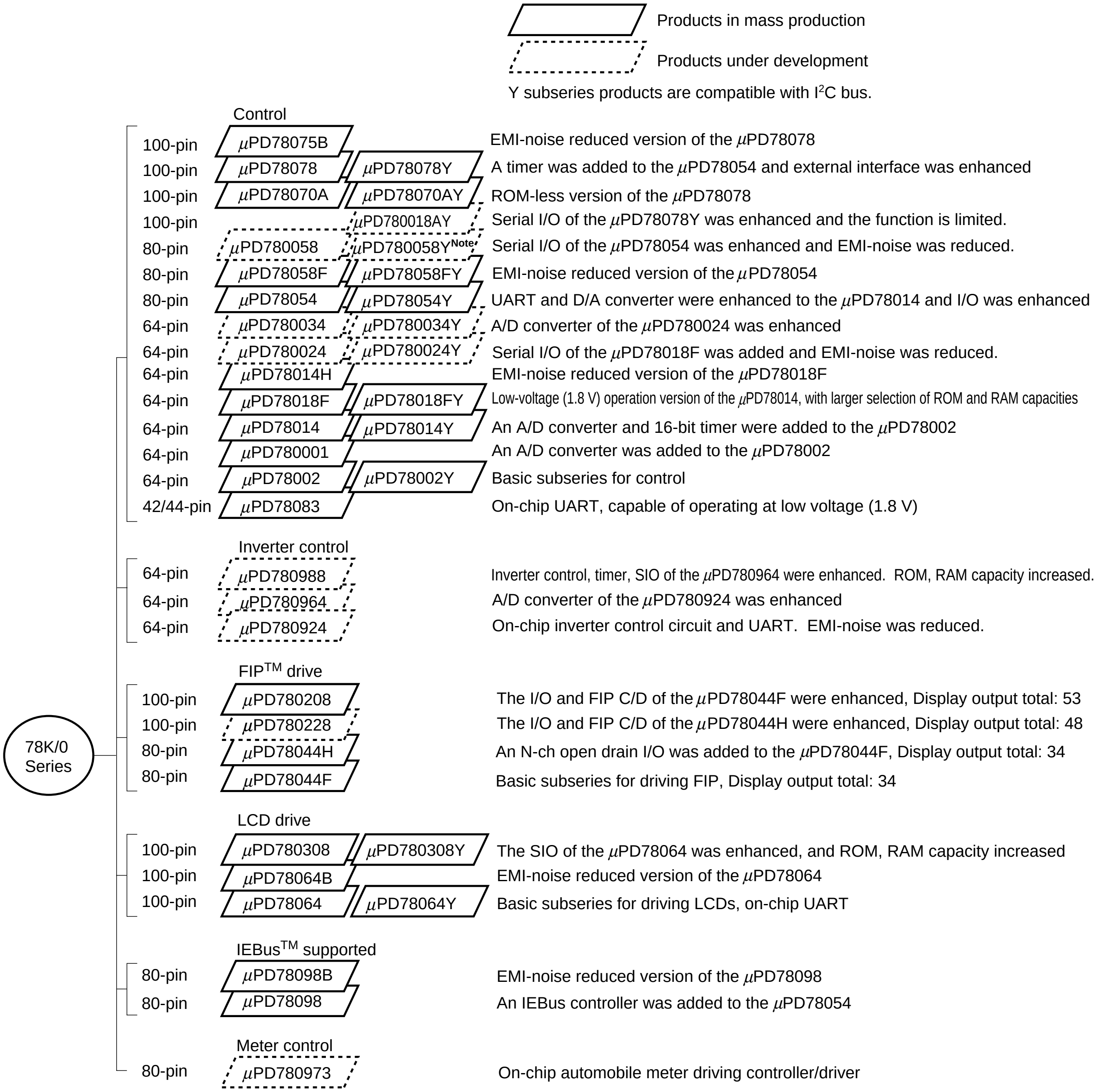
Cautions 1. (L) : Individually connect this pin to V_{SS} via a pull-down resistor (10 kΩ).
2. V_{SS} : Connect this pin to ground.
3. $\overline{\text{RESET}}$: Fix this pin to the low level.
4. Open : Leave this pin unconnected.

A0-A16	: Address Bus	$\overline{\text{RESET}}$	: Reset
$\overline{\text{CE}}$	: Chip Enable	V_{DD}	: Power Supply
D0-D7	: Data Bus	V_{PP}	: Programming Power Supply
$\overline{\text{OE}}$	: Output Enable	V_{SS}	: Ground
$\overline{\text{PGM}}$	: Program		

Phase-out/Discontinued

★ 1.6 Product Development of 78K/0 Series

The products in the 78K/0 series are listed below. The names enclosed in boxes are subseries names.


Products in mass production
Products under development
Y subseries products are compatible with I^{2}C bus.
78K/0 Series

Control

100-pin	µPD78075B		EMI-noise reduced version of the µPD78078
100-pin	µPD78078	µPD78078Y	A timer was added to the µPD78054 and external interface was enhanced
100-pin	µPD78070A	µPD78070AY	ROM-less version of the µPD78078
100-pin		µPD780018AY	Serial I/O of the µPD78078Y was enhanced and the function is limited.
80-pin	µPD780058	µPD780058Y[Note]	Serial I/O of the µPD78054 was enhanced and EMI-noise was reduced.
80-pin	µPD78058F	µPD78058FY	EMI-noise reduced version of the µPD78054
80-pin	µPD78054	µPD78054Y	UART and D/A converter were enhanced to the µPD78014 and I/O was enhanced
64-pin	µPD780034	µPD780034Y	A/D converter of the µPD780024 was enhanced
64-pin	µPD780024	µPD780024Y	Serial I/O of the µPD78018F was added and EMI-noise was reduced.
64-pin	µPD78014H		EMI-noise reduced version of the µPD78018F
64-pin	µPD78018F	µPD78018FY	Low-voltage (1.8 V) operation version of the µPD78014, with larger selection of ROM and RAM capacities
64-pin	µPD78014	µPD78014Y	An A/D converter and 16-bit timer were added to the µPD78002
64-pin	µPD780001		An A/D converter was added to the µPD78002
64-pin	µPD78002	µPD78002Y	Basic subseries for control
42/44-pin	µPD78083		On-chip UART, capable of operating at low voltage (1.8 V)

Inverter control

64-pin	µPD780988	Inverter control, timer, SIO of the µPD780964 were enhanced. ROM, RAM capacity increased.
64-pin	µPD780964	A/D converter of the µPD780924 was enhanced
64-pin	µPD780924	On-chip inverter control circuit and UART. EMI-noise was reduced.

FIP^{TM} drive

100-pin	µPD780208	The I/O and FIP C/D of the µPD78044F were enhanced, Display output total: 53
100-pin	µPD780228	The I/O and FIP C/D of the µPD78044H were enhanced, Display output total: 48
80-pin	µPD78044H	An N-ch open drain I/O was added to the µPD78044F, Display output total: 34
80-pin	µPD78044F	Basic subseries for driving FIP, Display output total: 34

LCD drive

100-pin	µPD780308	µPD780308Y	The SIO of the µPD78064 was enhanced, and ROM, RAM capacity increased
100-pin	µPD78064B		EMI-noise reduced version of the µPD78064
100-pin	µPD78064	µPD78064Y	Basic subseries for driving LCDs, on-chip UART

IEBus^{TM} supported

80-pin	µPD78098B	EMI-noise reduced version of the µPD78098
80-pin	µPD78098	An IEBus controller was added to the µPD78054

Meter control

80-pin	µPD780973	On-chip automobile meter driving controller/driver

Note Under planning

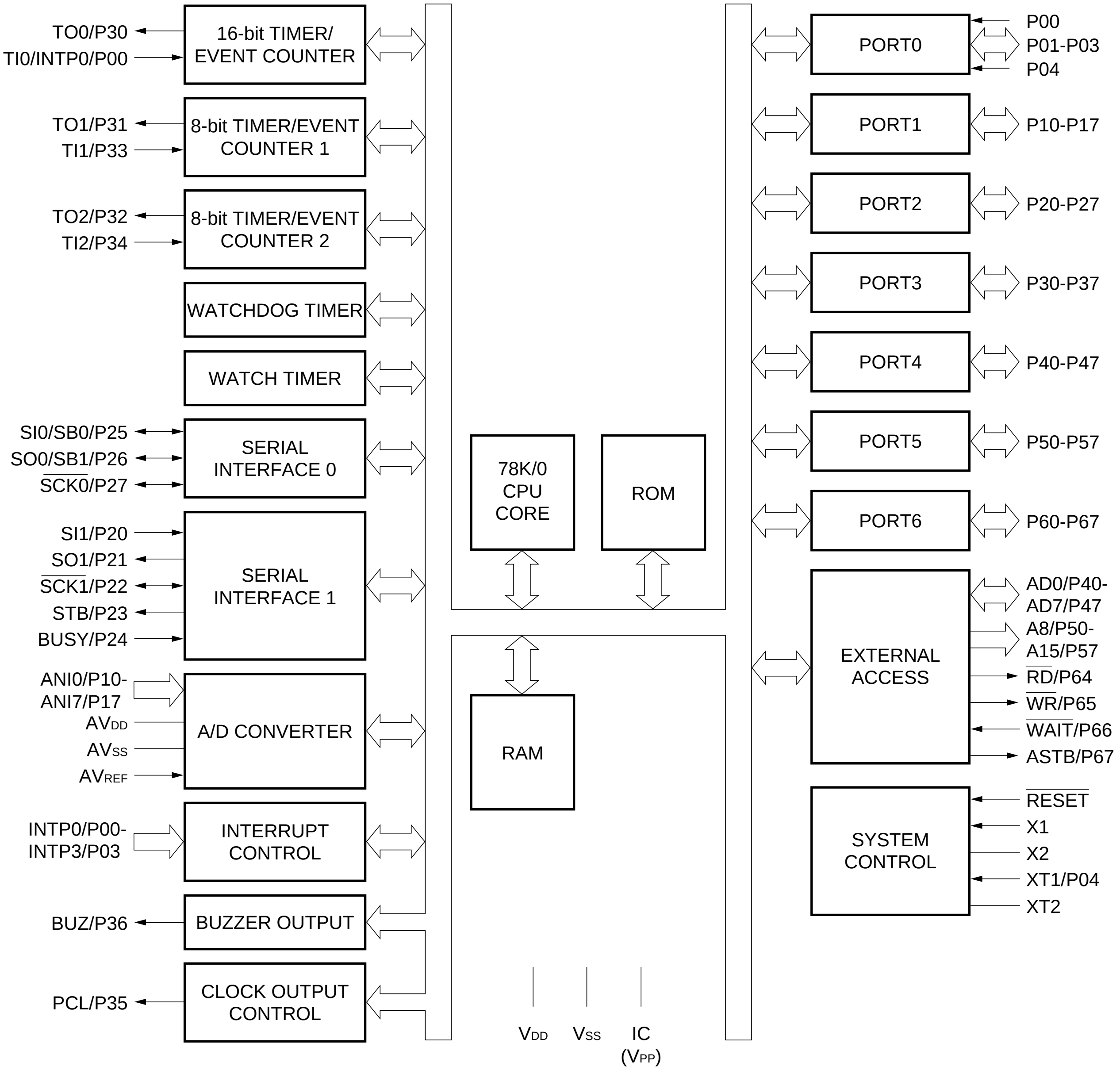
The following lists the main functional differences between subseries products.

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



Remarks 1. The internal ROM and RAM capacities differ depending on the product.

2. ( ): µPD78P018F

1.8 Functional Outline

Item / Part Number		μPD78011F	μPD78012F	μPD78013F	μPD78014F	μPD78015F	μPD78016F	μPD78018F	μPD78P018F
Internal memory	ROM	Mask ROM							PROM
		8 KB	16 KB	24 KB	32 KB	40 KB	48 KB	60 KB	60 KB[Note 1]
	High-speed RAM	512 B		1024 B					1024 B[Note 1]
	Extension RAM	–				512 B		1024 B	1024 B[Note 2]
	Buffer RAM	32 B							
Memory space		64 KB							
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 model 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 transmit/receive function of up to 32 B): 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 extension RAM can be changed by using internal extension RAM size select register (IXS).

Part Number Item		μPD78011F	μPD78012F	μPD78013F	μPD78014F	μPD78015F	μPD78016F	μPD78018F	μPD78P018F
Vectored interrupt sources	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 mil) • 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							

1.9 Differences from Standard Quality Models and (A) Models

Table 1-1 shows the differences between the standard quality models (μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, 78018F, and 78P018F) and (A) models.

Table 1-1. Differences between Special Quality Models and (A) Models

Classification Item	Standard Quality Models	(A) Models
Quality grade	Standard	Special
Package	• 64-pin plastic shrink DIP (750 mil) • 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 mil) • 64-pin plastic QFP (14 × 14 mm)

1.10 Differences between (A) Model and (A2) Model

Table 1-2 shows the differences between the (A) model (μPD78012F(A)) and (A2) model (μPD78012F(A2)).

Table 1-2. Differences between (A) Model and (A2) Model

Item \ Classification	(A) Model	(A2) Model
Operating ambient temperature	–40 to +85 °C	–40 to +125 °C
DC characteristics	Analog pin input leakage current, power supply current, and data retention current differ	
AC characteristics	Bus timing differs	

1.11 Mask Option

The mask ROM models (μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, and 78018F) have a mask option. By specifying the mask option when placing your order, the pull-up resistors shown in Table 1-3 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 1-3 shows the mask option for the μPD78018F subseries.

Table 1-3. Mask Option for Mask ROM Model

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

Phase-out/Discontinued

[MEMO]

CHAPTER 2 GENERAL (µPD78018FY SUBSERIES)

2.1 Features

- High-capacity ROM and RAM

Item / Part Number	Program Memory (ROM)	Data Memory		
		Internal High-Speed RAM	Internal Extension RAM	Internal Buffer RAM
µPD78011FY	8 KB	512 B	—	32 B
µPD78012FY	16 KB			
µPD78013FY	24 KB	1024 B		
µPD78014FY	32 KB			
µPD78015FY	40 KB		512 B	
µPD78016FY	48 KB			
µPD78018FY	60 KB		1024 B	
µPD78P018FY	60 KB[Note 1]	1024 B[Note 2]	1024 B[Note 3]	

Notes **1.** 8, 16, 24, 32, 40, 48, or 60 KB is selectable by using memory size select register (IMS).
2. 512 or 1024 B is selectable by using IMS.
3. 0, 512, or 1024 B is selectable by using internal extension RAM size select register (IXS).

- External memory extension space: 64 KB
- 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 / 2-wire serial I/O / I^{2}C bus mode	:	1 channel
· 3-wire serial I/O mode (with automatic transmit/receive 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 lines
- 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

2.2 Application Field

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

2.3 Ordering Information

Part Number	Package	Internal ROM
μPD78011FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78011FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78011FYGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78012FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78012FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78012FYGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78013FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78013FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78014FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78014FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78014FYGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Mask ROM
μPD78015FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78015FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78016FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78016FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78018FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Mask ROM
μPD78018FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Mask ROM
μPD78P018FYCW	64-pin plastic shrink DIP (750 mil)	One-time PROM
μPD78P018FYDW	64-pin ceramic shrink DIP (with window) (750 mil)	EPROM
μPD78P018FYGC-AB8	64-pin plastic QFP (14 × 14 mm)	One-time PROM
μPD78P018FYGK-8A8	64-pin plastic LQFP (12 × 12 mm)	One-time PROM
μPD78P018FYKK-S	64-pin ceramic WQFN (14 × 14 mm)	EPROM

Remark XXX indicates ROM code suffix.

2.4 Quality Grade

Part Number	Package	Quality Grade
μPD78011FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78011FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78011FYGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78012FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78012FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78012FYGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78013FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78013FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78014FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78014FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78014FYGK-XXX-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78015FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78015FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78016FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78016FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78018FYCW-XXX	64-pin plastic shrink DIP (750 mil)	Standard
μPD78018FYGC-XXX-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78P018FYCW	64-pin plastic shrink DIP (750 mil)	Standard
μPD78P018FYDW	64-pin ceramic shrink DIP (with window) (750 mil)	Not applicable (for function evaluation)
μPD78P018FYGC-AB8	64-pin plastic QFP (14 × 14 mm)	Standard
μPD78018FYGK-8A8	64-pin plastic LQFP (12 × 12 mm)	Standard
μPD78P018FYKK-S	64-pin ceramic WQFN (14 × 14 mm)	Not applicable (for function evaluation)

Caution Of the μPD78P018FYDW, and 78P018FYKK-S do not have a reliability intended for mass production of your systems. Use these models for experiment or function evaluation only.

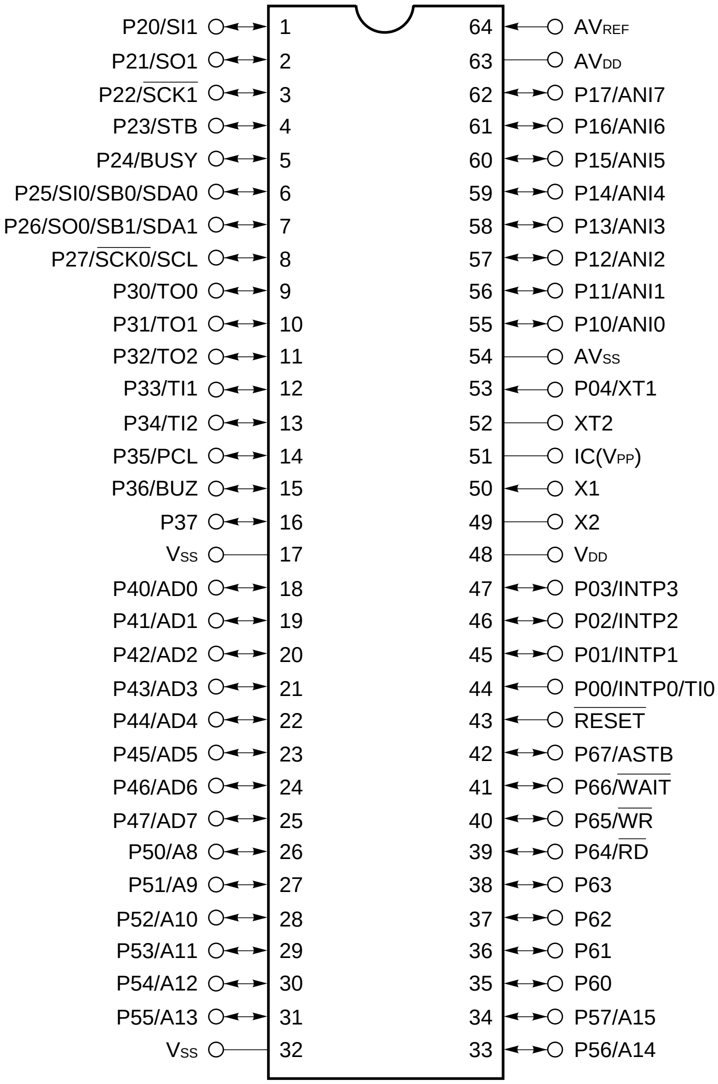
Remark XXX indicates ROM code suffix.

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

2.5 Pin Configuration (Top View)

(1) Normal operation mode

- **64-pin plastic shrink DIP (750 mil)**
  μPD78011FYCW-XXX, 78012FYCW-XXX, 78013FYCW-XXX,
  μPD78014FYCW-XXX, 78015FYCW-XXX, 78016FYCW-XXX,
  μPD78018FYCW-XXX, 78P018FYCW
- **64-pin ceramic shrink DIP (with window) (750 mil)**
  μPD78P018FYDW



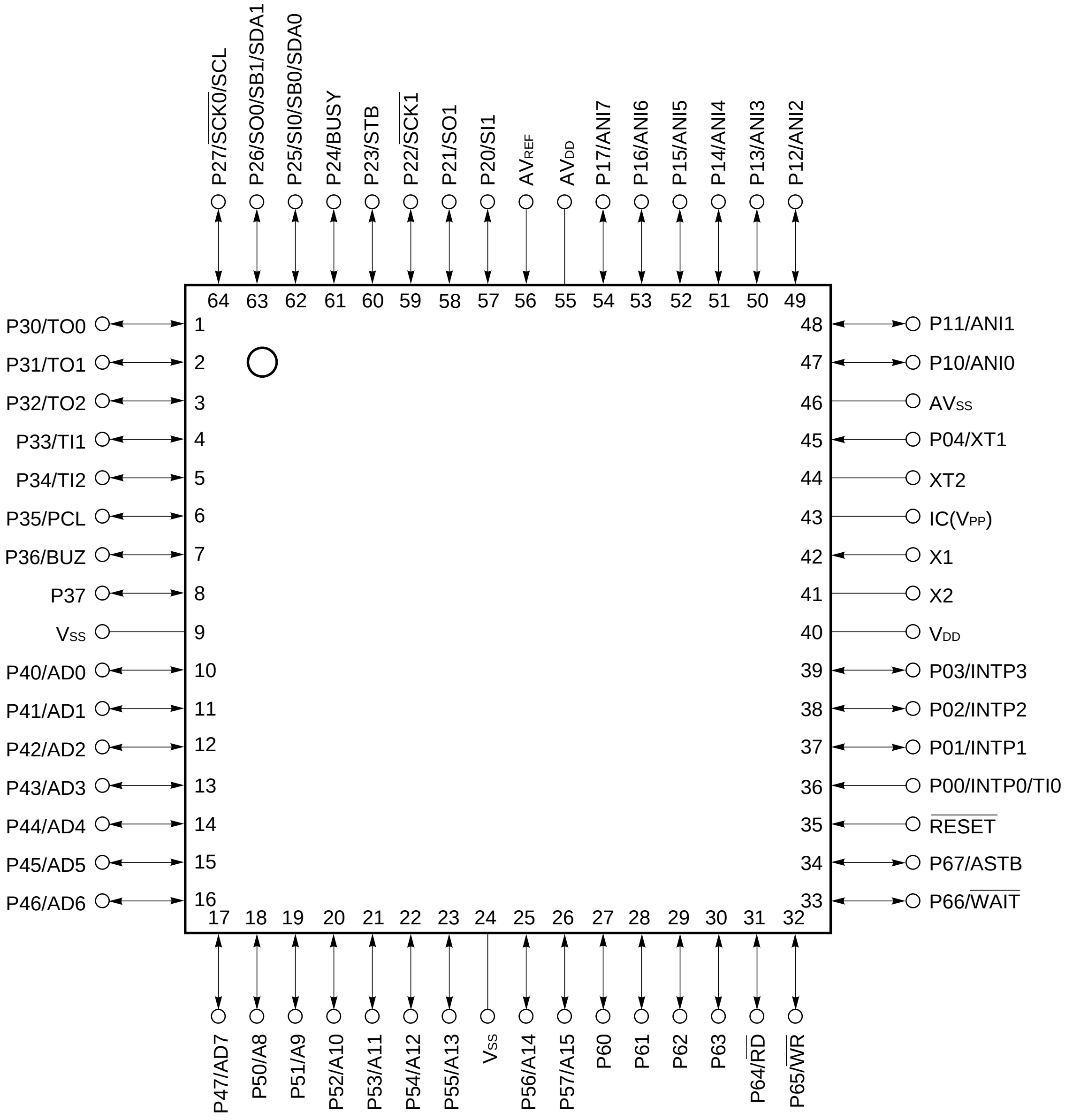
Cautions 1. Directly connect the IC (Internally Connected) pins to V_{SS}.

2. Connect the AV_{DD} pin to V_{DD}.

3. Connect the AV_{SS} pin to V_{SS}.

Remark ( ): μPD78P018FY

- **64-pin plastic QFP (14 × 14 mm)**
  μPD78011FYGC-XXX-AB8, 78012FYGC-XXX-AB8, 78013FYGC-XXX-AB8,
  μPD78014FYGC-XXX-AB8, 78015FYGC-XXX-AB8, 78016FYGC-XXX-AB8,
  μPD78018FYGC-XXX-AB8, 78P018FYGC-AB8
- **64-pin plastic LQFP (12 × 12 mm)**
  μPD78011FYGK-XXX-8A8, 78012FYGK-XXX-8A8,
  μPD78014FYGK-XXX-8A8, 78P018FYGK-8A8
- **64-pin ceramic WQFN (14 × 14 mm)**
  μPD78P018FYKK-S



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

2. Connect the AV$_{DD}$ pin to V$_{DD}$.

3. Connect the AV$_{SS}$ pin to V$_{SS}$.

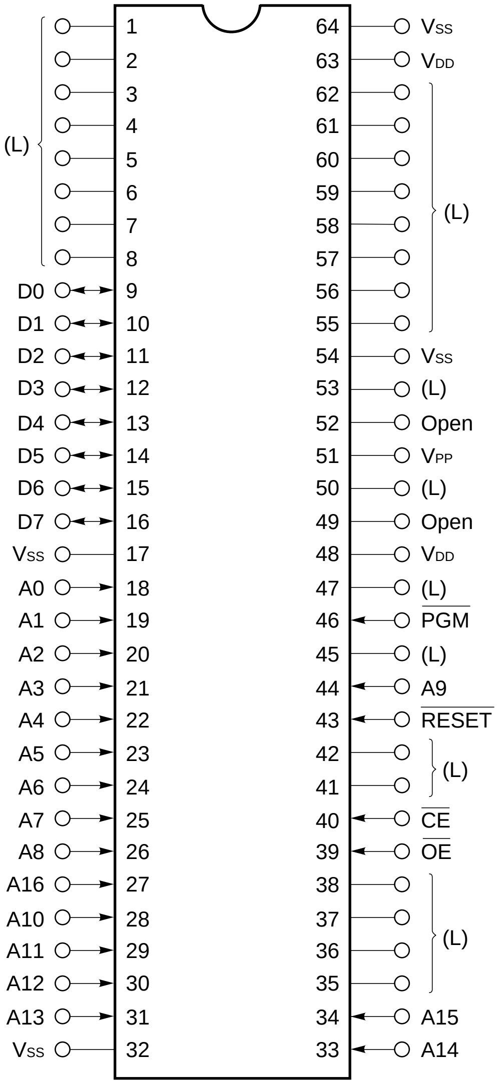
Remark ( ): μPD78P018FY

A8-A15	: Address Bus	$\overline{RD}$	: Read Strobe
AD0-AD7	: Address/Data Bus	$\overline{RESET}$	: Reset
ASTB	: Address Strobe	SB0, SB1	: Serial Bus
ANI0-ANI7	: Analog Input	$\overline{SCK0}$, $\overline{SCK1}$	: Serial Clock
AV_{DD}	: Analog Power Supply	SCL	: Serial Clock
AV_{REF}	: Analog Reference Voltage	SDA0, SDA1	: Serial Data
AV_{SS}	: Analog Ground	SI0, SI1	: Serial Input
BUSY	: Busy	SO0, SO1	: Serial Output
BUZ	: Buzzer Clock	STB	: Strobe
IC	: Internally Connected	TI0-TI2	: Timer Input
INTP0-INTP3	: Interrupt from Peripherals	TO0-TO2	: Timer Output
P00-P04	: Port 0	$\overline{WAIT}$	: Wait
P10-P17	: Port 1	$\overline{WR}$	: Write Strobe
P20-P27	: Port 2	X1, X2	: Crystal (Main System Clock)
P30-P37	: Port 3	XT1, XT2	: Crystal (Subsystem Clolck)
P40-P47	: Port 4	V_{DD}	: Power Supply
P50-P57	: Port 5	V_{PP}	: Programming Power Supply
P60-P67	: Port 6	V_{SS}	: Ground
PCL	: Programmable Clock		

Remark V_{PP} is for the μPD78P018FY. It is replaced by an IC pin for the mask ROM model.

(2) PROM programming mode

- **64-pin plastic shrink DIP (750 mil)**
  μPD78P018FYCW
- **64-pin ceramic shrink DIP (with window) (750 mil)**
  μPD78P018FYDW



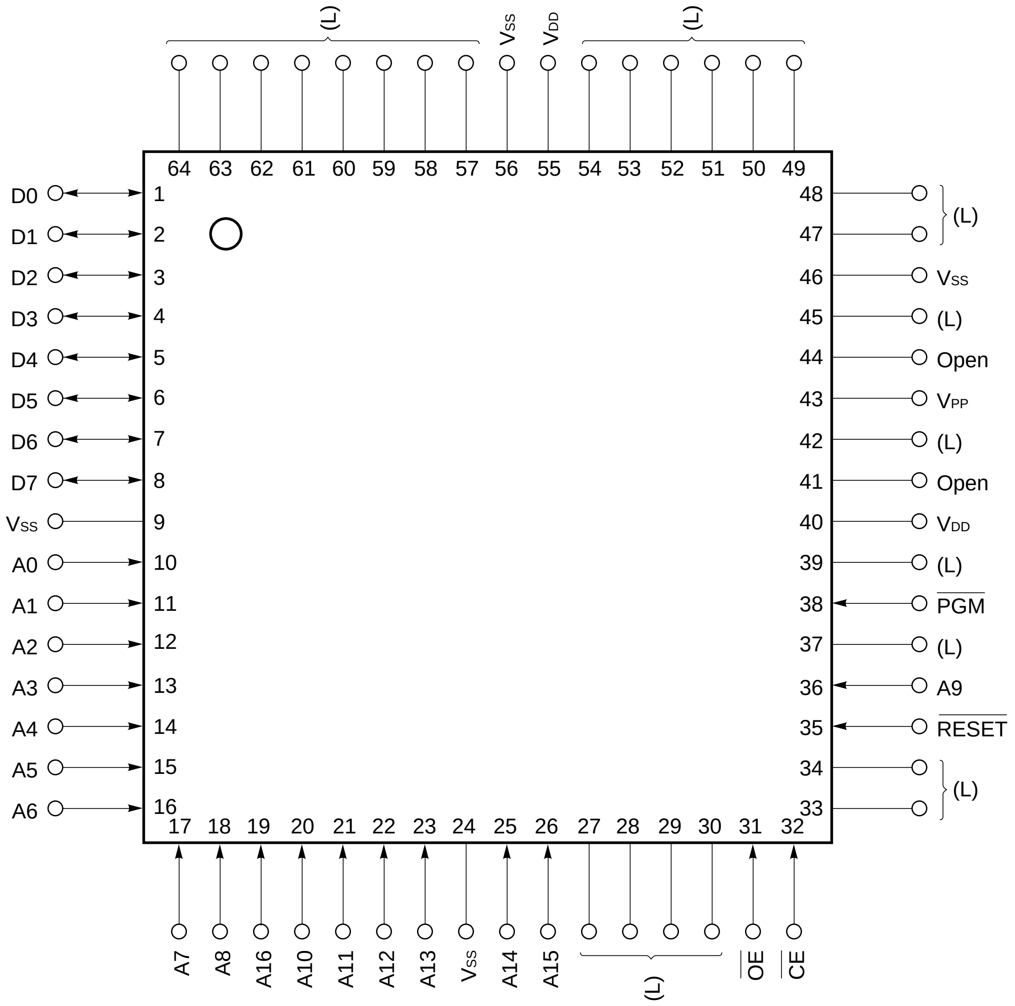
Cautions 1. (L) : Individually connect this pin to V_{SS} via a pull-down resistor (10 kΩ).

2. V_{SS} : Connect this pin to ground.

3. $\overline{\text{RESET}}$: Fix this pin to the low level.

4. Open : Leave this pin unconnected.

- **64-pin plastic QFP (14 × 14 mm)**
  μPD78P018FYGC-AB8
- **64-pin plastic LQFP (12 × 12 mm)**
  μPD78P018FYGK-8A8
- **64-pin ceramic WQFN (14 × 14 mm)**
  μPD78P018FYKK-S



Cautions 1. (L) : Individually connect this pin to V_{SS} via a pull-down resistor (10 kΩ).

2. V_{SS} : Connect this pin to ground.

3. $\overline{\text{RESET}}$: Fix this pin to the low level.

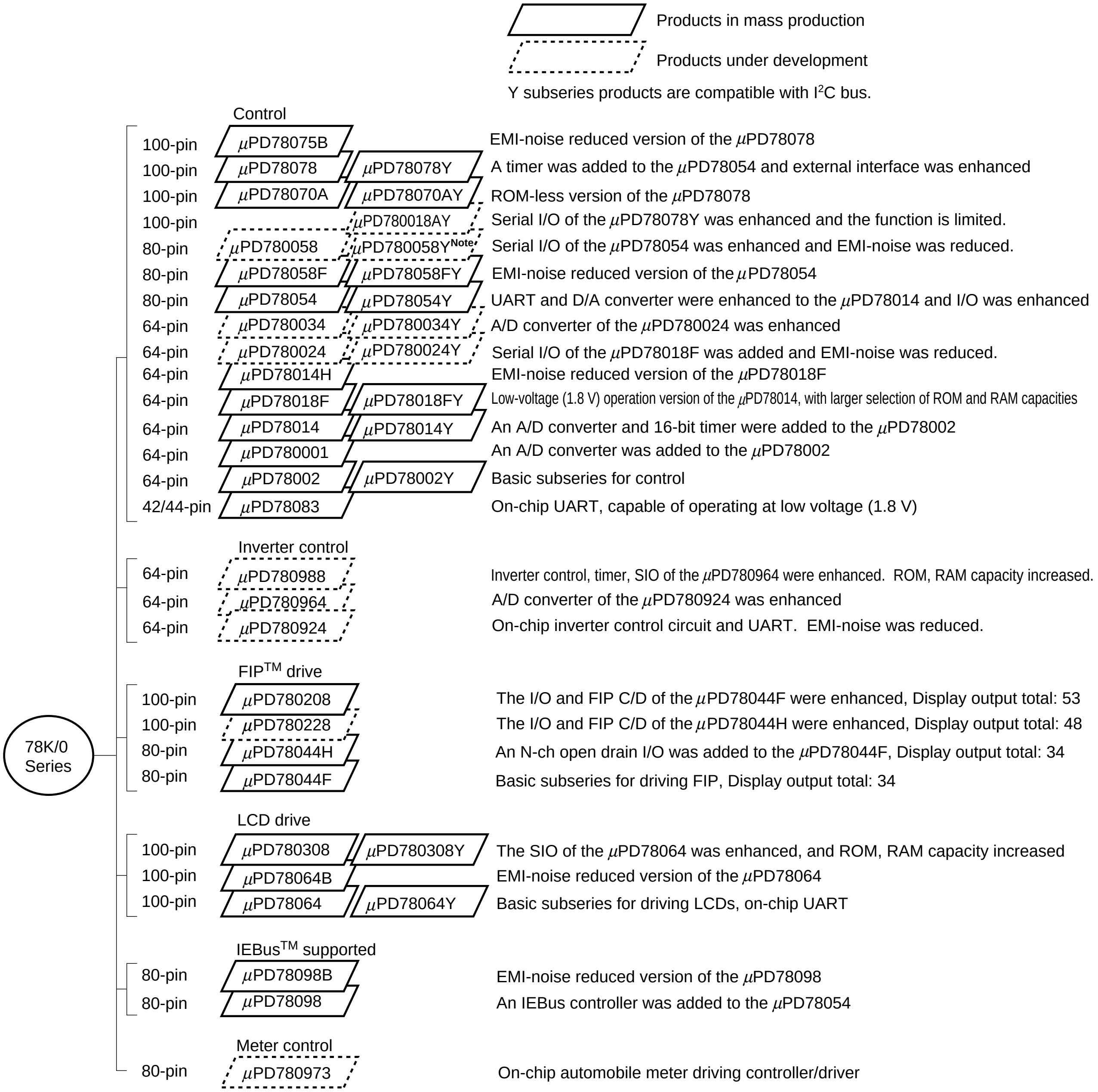
4. Open : Leave this pin unconnected.

A0-A16	:	Address Bus	$\overline{\text{RESET}}$	:	Reset
$\overline{\text{CE}}$	:	Chip Enable	V_{DD}	:	Power Supply
D0-D7	:	Data Bus	V_{PP}	:	Programming Power Supply
$\overline{\text{OE}}$	:	Output Enable	V_{SS}	:	Ground
$\overline{\text{PGM}}$	:	Program			

Phase-out/Discontinued

2.6 Product Development of 78K/0 Series

The products in the 78K/0 series are listed below. The names enclosed in boxes are subseries names.



Products in mass production

Products under development

Y subseries products are compatible with I²C bus.

78K/0 Series

Control

100-pin	μPD78075B		EMI-noise reduced version of the μPD78078
100-pin	μPD78078	μPD78078Y	A timer was added to the μPD78054 and external interface was enhanced
100-pin	μPD78070A	μPD78070AY	ROM-less version of the μPD78078
100-pin		μPD780018AY	Serial I/O of the μPD78078Y was enhanced and the function is limited.
80-pin	μPD780058	μPD780058Y^Note	Serial I/O of the μPD78054 was enhanced and EMI-noise was reduced.
80-pin	μPD78058F	μPD78058FY	EMI-noise reduced version of the μPD78054
80-pin	μPD78054	μPD78054Y	UART and D/A converter were enhanced to the μPD78014 and I/O was enhanced
64-pin	μPD780034	μPD780034Y	A/D converter of the μPD780024 was enhanced
64-pin	μPD780024	μPD780024Y	Serial I/O of the μPD78018F was added and EMI-noise was reduced.
64-pin	μPD78014H		EMI-noise reduced version of the μPD78018F
64-pin	μPD78018F	μPD78018FY	Low-voltage (1.8 V) operation version of the μPD78014, with larger selection of ROM and RAM capacities
64-pin	μPD78014	μPD78014Y	An A/D converter and 16-bit timer were added to the μPD78002
64-pin	μPD780001		An A/D converter was added to the μPD78002
64-pin	μPD78002	μPD78002Y	Basic subseries for control
42/44-pin	μPD78083		On-chip UART, capable of operating at low voltage (1.8 V)

Inverter control

64-pin	μPD780988	Inverter control, timer, SIO of the μPD780964 were enhanced. ROM, RAM capacity increased.
64-pin	μPD780964	A/D converter of the μPD780924 was enhanced
64-pin	μPD780924	On-chip inverter control circuit and UART. EMI-noise was reduced.

FIP™ drive

100-pin	μPD780208	The I/O and FIP C/D of the μPD78044F were enhanced, Display output total: 53
100-pin	μPD780228	The I/O and FIP C/D of the μPD78044H were enhanced, Display output total: 48
80-pin	μPD78044H	An N-ch open drain I/O was added to the μPD78044F, Display output total: 34
80-pin	μPD78044F	Basic subseries for driving FIP, Display output total: 34

LCD drive

100-pin	μPD780308	μPD780308Y	The SIO of the μPD78064 was enhanced, and ROM, RAM capacity increased
100-pin	μPD78064B		EMI-noise reduced version of the μPD78064
100-pin	μPD78064	μPD78064Y	Basic subseries for driving LCDs, on-chip UART

IEBus™ supported

80-pin	μPD78098B	EMI-noise reduced version of the μPD78098
80-pin	μPD78098	An IEBus controller was added to the μPD78054

Meter control

80-pin	μPD780973	On-chip automobile meter driving controller/driver

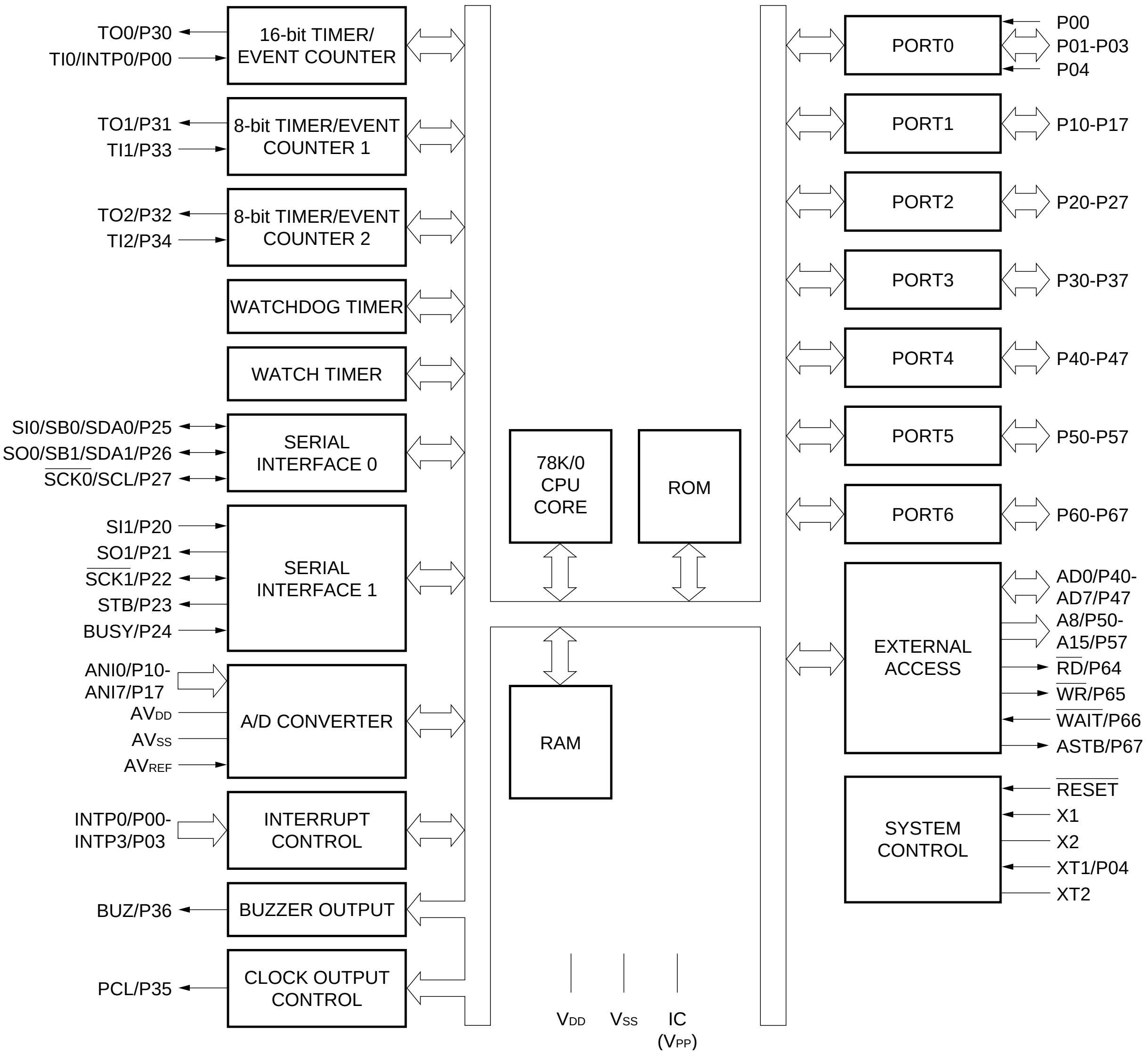
Note Under planning

The following lists the main functional differences between subseries products.

Function		ROM	Serial Interface	I/O	V_{DD}
Subseries Name		Capacity			MIN. Value
Control	µPD78078Y	48 K-60 K	3-wire/2-wire/I^{2}C :1 ch 3-wire with automatic transmit/receive function :1 ch 3-wire/UART :1 ch	88	1.8 V
	µPD78070AY	—		61	2.7 V
	µPD780018AY	48 K-60 K	3-wire with automatic transmit/receive function :1 ch Time division 3-wire :1 ch I^{2}C bus (multi-master compatible) :1 ch	88	
	µPD780058Y	24 K-60 K	3-wire/2-wire/I^{2}C :1 ch 3-wire with automatic transmit/receive function :1 ch 3-wire/time division UART :1 ch	68	1.8 V
	µPD78058FY	48 K-60 K	3-wire/2-wire/I^{2}C :1 ch 3-wire with automatic transmit/receive function :1 ch 3-wire/UART :1 ch	69	2.7 V
	µPD78054Y	16 K-60 K			2.0 V
	µPD780034Y	8 K-32 K	UART :1 ch 3-wire :1 ch I^{2}C bus (multi-master compatible) :1 ch	51	1.8 V
	µPD780024Y				
	µPD78018FY	8 K-60 K	3-wire/2-wire/I^{2}C :1 ch 3-wire with automatic transmit/receive function :1 ch	53	
	µPD78014Y	8 K-32 K	3-wire/2-wire/SBI/I^{2}C :1 ch 3-wire with automatic transmit/receive function :1 ch		2.7 V
	µPD78002Y	8 K-16 K	3-wire/2-wire/SBI/I^{2}C :1 ch		
LCD drive	µPD780308Y	48 K-60 K	3-wire/2-wire/I^{2}C :1 ch 3-wire/time division UART :1 ch 3-wire :1 ch	57	2.0 V
	µPD78064Y	16 K-32 K	3-wire/2-wire/I^{2}C :1 ch 3-wire/UART :1 ch		

Remark The functions other than serial interface are common to the µPD78018F subseries.

2.7 Block Diagram



Remarks 1. The internal ROM and RAM capacities differ depending on the product.

2. ( ): μPD78P018FY

2.8 Functional Outline

Part Number Item		μPD78011FY	μPD78012FY	μPD78013FY	μPD78014FY	μPD78015FY	μPD78016FY	μPD78018FY	μPD78P018FY
Internal memory	ROM	Mask ROM							PROM
		8 KB	16 KB	24 KB	32 KB	40 KB	48 KB	60 KB	60 KB[Note 1]
	High-speed RAM	512 B		1024 B					1024 B[Note 1]
	Extension RAM	–				512 B		1024 B	1024 B[Note 2]
	Buffer RAM	32 B							
Memory space		64 KB							
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 model 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/2-wire serial I/O/I^2C bus mode selectable : 1 channel • 3-wire serial I/O mode (with automatic transmit/receive function of up to 32 B) : 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 extension RAM can be changed by using internal extension RAM size select register (IXS).

Item		μPD78011FY	μPD78012FY	μPD78013FY	μPD78014FY	μPD78015FY	μPD78016FY	μPD78018FY	μPD78P018FY
Vectored interrupt sources	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 mil) • 64-pin plastic QFP (14 × 14 mm) • 64-pin plastic LQFP (12 × 12 mm)[Note] • 64-pin ceramic shrink DIP (with window) (750 mil): μPD78P018FY only • 64-pin ceramic WQFN (14 × 14 mm): μPD78P018FY only							

Note μPD78011FY, 78012FY, 78014FY, and 78018FY only

2.9 Mask Option

The mask ROM models (μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, and 78018FY) have a mask option. By specifying the mask option when placing your order, the pull-up resistors shown in Table 2-1 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 2-1 shows the mask option for the μPD78018FY subseries.

Table 2-1. Mask Option for Mask ROM Model

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

[MEMO]

CHAPTER 3 PIN FUNCTIONS (μPD78018F SUBSERIES)

3.1 List of Pin Functions

3.1.1 Pins in normal operation mode

(1) Port pins (1/2)

Pin Name	I/O	Function		On Reset	Shared by:
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-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-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					$\overline{\text{SCK1}}$
P23					STB
P24					BUSY
P25					SI0/SB0
P26					SO0/SB1
P27					$\overline{\text{SCK0}}$

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 input mode. The internal pull-up resistors are automatically disconnected.

(1) Port pins (2/2)

Pin Name	I/O	Function		On Reset	Shared by:
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-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-AD7
P50-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-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 with mask ROM model only. 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

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

Pin Name	I/O	Function	On Reset	Shared by:
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
$\overline{\text{SCK0}}$	I/O	Serial clock I/O of serial interface	Input	P27
$\overline{\text{SCK1}}$				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-AD7	I/O	Low-order address/data bus when memory is externally extended	Input	P40-P47
A8-A15	Output	High-order address bus when memory is externally extended	Input	P50-P57
$\overline{\text{RD}}$	Output	Strobe signal output for external memory read	Input	P64
$\overline{\text{WR}}$		Strobe signal output for external memory write		P65
$\overline{\text{WAIT}}$	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	On Reset	Shared by:
ANI0-ANI7	Input	Analog input to A/D converter	Input	P10-P17
AV_{REF}	Input	Reference voltage input to A/D converter	—	—
AV_{DD}	—	Analog power to A/D converter. Connect to V_{DD}.	—	—
AV_{SS}	—	Ground of A/D converter. Connect to V_{SS}.	—	—
$\overline{\text{RESET}}$	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	—	—
V_{PP}	—	High-voltage application for program write/verify, Directly connect this pin to V_{SS} in normal operation mode.	—	—
V_{SS}	—	Ground	—	—
IC	—	Internally connected. Directly connect to V_{SS}	—	—

3.1.2 Pins in PROM programming mode (μPD78P018F only)

Pin Name	I/O	Function
$\overline{\text{RESET}}$	Input	Sets PROM programming mode. When +5 V or +12.5 V is applied to V_{PP} pin and low level is input to $\overline{\text{RESET}}$ pin, PROM programming mode is set.
V_{PP}	Input	Applies high voltage when setting PROM programming mode and when writing/verifying program
A0-A16	Input	Address bus
D0-D7	I/O	Data bus
$\overline{\text{CE}}$	Input	PROM enable input/program pulse input
$\overline{\text{OE}}$	Input	Read strobe input to PROM
$\overline{\text{PGM}}$	Input	Program/program inhibit input in PROM programming mode
V_{DD}	—	Positive power supply
V_{SS}	—	Ground

3.2 Description of Pin Functions

3.2.1 P00-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-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 request input pin.

(b) TI0

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

(c) XT1

Subsystem clock oscillation crystal connecting pin

3.2.2 P10-P17 (Port1)

These pins form an 8-bit I/O port, port 1. 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-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.

3.2.3 P20-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 transmission/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 transmission/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 transmit/receive function of the serial interface.

(e) STB

This pin outputs a strobe signal for the automatic transmit/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 15-3. Format of Serial Operation Mode Register 0 and Figure 17-3. Format of Serial Operation Mode Register 1.

3.2.4 P30-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-TO2

Timer output pins.

(c) PCL

Clock output pin.

(d) BUZ

Buzzer output pin.

3.2.5 P40-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 low-order address/data bus pins (AD0-AD7) in the external memory extension mode. The pins used as address/data bus pins are automatically disconnected from the internal pull-up resistor.

3.2.6 P50-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 high-order address bus pins (A8-A15) in the external memory extension mode. The pins used as address bus pins are automatically disconnected from the internal pull-up resistor.

3.2.7 P60-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 of the mask ROM model can be connected to an internal pull-up resistor 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 ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, 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.

3.2.8 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.

3.2.9 AV_{DD}

This is the analog power supply pin of the A/D converter.

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

3.2.10 AV_{SS}

This is the ground pin of the A/D converter.

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

3.2.11 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

3.2.12 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.

3.2.13 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.

3.2.14 V_{DD}

Positive power supply pin

3.2.15 V_{SS}

Ground pin

3.2.16 V_{PP} (μPD78P018F only)

A high voltage should be applied to this pin when the PROM programming mode is set and when the program is written or verified.

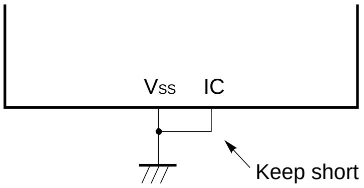
Directly connect this pin to V_{SS} in the normal operation mode.

3.2.17 IC (mask ROM model only)

The IC (Internally Connected) pin is used to set the μPD78011F, 78012F, 78013F, 78014F, 78015F, 78016F, and 78018F 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.



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

Table 3-1 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 3-1.

Table 3-1. 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-A	I/O	Individually connect to V_{SS} via resistor
P02/INTP2			
P03/INTP3			
P04/XT1	16	Input	Connect to V_{DD}
P10/ANI0-P17/ANI7	11	I/O	Individually connect to V_{DD} or V_{SS}
P20/SI1	8-A		
P21/SO1	5-A		
P22/$\overline{\text{SCK1}}$	8-A		
P23/STB	5-A		
P24/BUSY	8-A		
P25/SI0/SB0	10-A		
P26/SO0/SB1			
P27/$\overline{\text{SCK0}}$			
P30/TO0	5-A		
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			
P37			
P40/AD0-P47/AD7	5-E	I/O	Individually connect to V_{DD} via resistor
P50/A8-P57/A15	5-A	I/O	Individually connect to V_{DD} or V_{SS} via resistor
P60-P63 (Mask ROM model)	13-B	I/O	Individually connect to V_{DD} via resistor
P60-P63 (PROM model)	13-D		
P64/$\overline{\text{RD}}$	5-A		Individually connect to V_{DD} or V_{SS} via resistor
P65/$\overline{\text{WR}}$			
P66/$\overline{\text{WAIT}}$			
P67/ASTB			
$\overline{\text{RESET}}$	2	Input	—
XT2	16	—	Open
AV_{REF}	—		Connect to V_{SS}
AV_{DD}			Connect to V_{DD}
AV_{SS}			Connect to V_{SS}
IC (Mask ROM model)			Directly connect to V_{SS}
V_{PP} (PROM model)			

Figure 3-1. I/O Circuits of Pins (1/2)

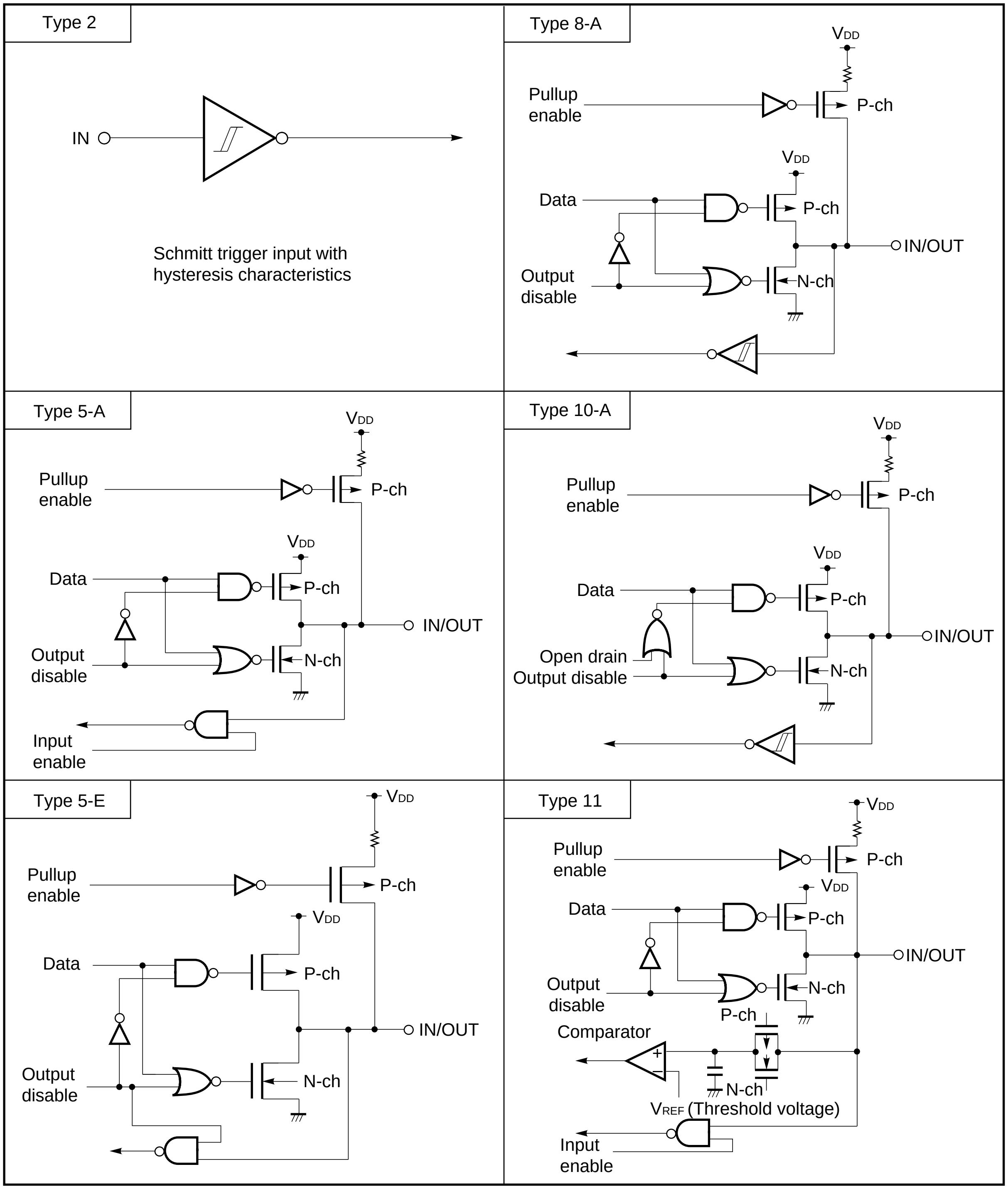
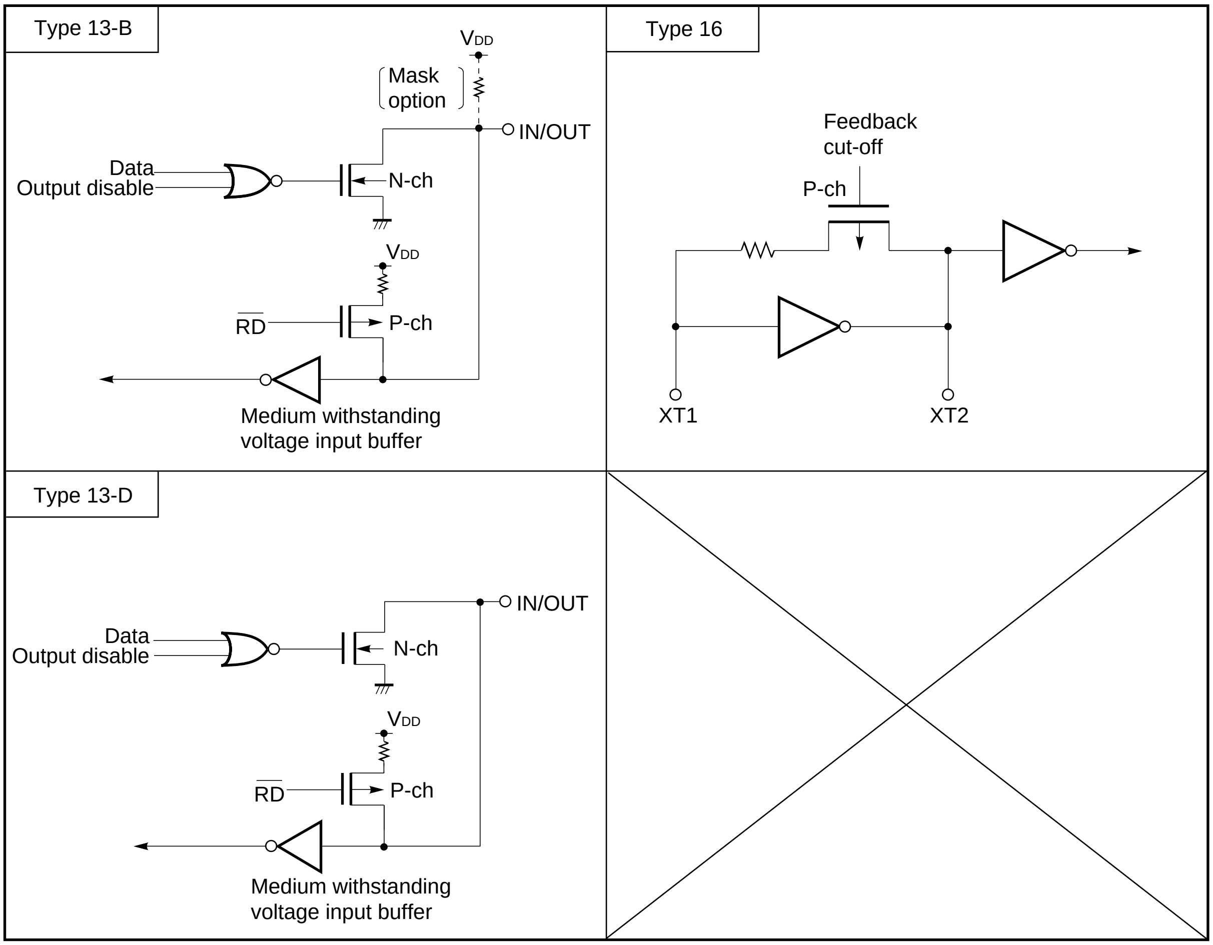


Figure 3-1. I/O Circuits of Pins (2/2)



4.1 List of Pin Functions

4.1.1 Pins in normal operation mode

(1) Port pins (1/2)

Pin Name	I/O	Function		On Reset	Shared by:
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-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-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					$\overline{\text{SCK1}}$
P23					STB
P24					BUSY
P25					SI0/SB0/SDA0
P26					SO0/SB1/SDA1
P27					$\overline{\text{SCK0}}$/SCL

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 input mode. The internal pull-up resistors are automatically disconnected.

(1) Port pins (2/2)

<table>
<tr><th>Pin Name</th><th>I/O</th><th colspan="2">Function</th><th>On Reset</th><th>Shared by:</th></tr>
<tr><td>P30</td><td rowspan="8">I/O</td><td colspan="2" rowspan="8">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.</td><td rowspan="8">Input</td><td>TO0</td></tr>
<tr><td>P31</td><td>TO1</td></tr>
<tr><td>P32</td><td>TO2</td></tr>
<tr><td>P33</td><td>TI1</td></tr>
<tr><td>P34</td><td>TI2</td></tr>
<tr><td>P35</td><td>PCL</td></tr>
<tr><td>P36</td><td>BUZ</td></tr>
<tr><td>P37</td><td>—</td></tr>
<tr><td>P40-P47</td><td>I/O</td><td colspan="2">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.</td><td>Input</td><td>AD0-AD7</td></tr>
<tr><td>P50-P57</td><td>I/O</td><td colspan="2">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.</td><td>Input</td><td>A8-A15</td></tr>
<tr><td>P60</td><td rowspan="8">I/O</td><td rowspan="8">Port 6.
8-bit I/O port.
Can be specified for input/output bitwise.</td><td rowspan="4">N-ch open-drain I/O port.
Connection of internal pull-up resistor can be specified by mask option with mask ROM model only.
Can directly drive LED.</td><td rowspan="8">Input</td><td rowspan="4">—</td></tr>
<tr><td>P61</td></tr>
<tr><td>P62</td></tr>
<tr><td>P63</td></tr>
<tr><td>P64</td><td rowspan="4">When used as input port, internal pull-up resistor can be connected by software.</td><td>$\overline{RD}$</td></tr>
<tr><td>P65</td><td>$\overline{WR}$</td></tr>
<tr><td>P66</td><td>$\overline{WAIT}$</td></tr>
<tr><td>P67</td><td>ASTB</td></tr>
</table>

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

Pin Name	I/O	Function	On Reset	Shared by:
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/SDA0
SI1				P20
SO0	Output	Serial data output of serial interface	Input	P26/SB1/SDA1
SO1				P21
SB0	I/O	Serial data I/O of serial interface	Input	P25/SI0/SDA0
SB1				P26/SO0/SDA1
SDA0				P25/SI0/SB0
SDA1				P26/SO0/SB1
$\overline{\text{SCK0}}$	I/O	Serial clock I/O of serial interface	Input	P27/SCL
$\overline{\text{SCK1}}$				P22
SCL				P27/$\overline{\text{SCK0}}$
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-AD7	I/O	Low-order address/data bus when memory is externally extended	Input	P40-P47
A8-A15	Output	High-order address bus when memory is externally extended	Input	P50-P57
$\overline{\text{RD}}$	Output	Strobe signal output for external memory read	Input	P64
$\overline{\text{WR}}$		Strobe signal output for external memory write		P65
$\overline{\text{WAIT}}$	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	On Reset	Shared by:
ANI0-ANI7	Input	Analog input to A/D converter	Input	P10-P17
AV_{REF}	Input	Reference voltage input to A/D converter	—	—
AV_{DD}	—	Analog power to A/D converter. Connect to V_{DD}.	—	—
AV_{SS}	—	Ground of A/D converter. Connect to V_{SS}.	—	—
$\overline{\text{RESET}}$	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	—	—
V_{PP}	—	High-voltage application for program write/verify, Directly connect this pin to V_{SS} in normal operation mode.	—	—
V_{SS}	—	Ground	—	—
IC	—	Internally connected. Directly connect to V_{SS}	—	—

4.1.2 Pins in PROM programming mode (μPD78P018FY only)

Pin Name	I/O	Function
$\overline{\text{RESET}}$	Input	Sets PROM programming mode. When +5 V or +12.5 V is applied to V_{PP} pin and low level is input to $\overline{\text{RESET}}$ pin, PROM programming mode is set.
V_{PP}	Input	Applies high voltage when setting PROM programming mode and when writing/verifying program
A0-A16	Input	Address bus
D0-D7	I/O	Data bus
$\overline{\text{CE}}$	Input	PROM enable input/program pulse input
$\overline{\text{OE}}$	Input	Read strobe input to PROM
$\overline{\text{PGM}}$	Input	Program/program inhibit input in PROM programming mode
V_{DD}	—	Positive power supply
V_{SS}	—	Ground

4.2 Description of Pin Functions

4.2.1 P00-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-INTP3

INTP0-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 request input pin.

(b) TI0

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

(c) XT1

Subsystem clock oscillation crystal connecting pin

4.2.2 P10-P17 (Port1)

These pins form an 8-bit I/O port, port 1. 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-ANI7) of the A/D converter. When these pins are specified to serve as analog input pins, the pull-up resistor is automatically disconnected.

4.2.3 P20-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 transmission/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 transmission/reception, and output a strobe signal.

(a) SI0, SI1, SO0, SO1, SB0, SB1, SDA0, SDA1

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

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

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

(c) BUSY

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

(d) STB

This pin outputs a strobe signal for the automatic transmit/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 16-3. Format of Serial Operation Mode Register 0 and Figure 17-3. Format of Serial Operation Mode Register 1.**

4.2.4 P30-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-TO2

Timer output pins.

(c) PCL

Clock output pin.

(d) BUZ

Buzzer output pin.

4.2.5 P40-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 low-order address/data bus pins (AD0-AD7) in the external memory extension mode. The pins used as address/data bus pins are automatically disconnected from the internal pull-up resistor.

4.2.6 P50-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 high-order address bus pins (A8-A15) in the external memory extension mode. The pins used as address bus pins are automatically disconnected from the internal pull-up resistor.

4.2.7 P60-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 of the mask ROM model can be connected to an internal pull-up resistor 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 ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, 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.

4.2.8 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.

4.2.9 AV_{DD}

This is the analog power supply pin of the A/D converter.

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

4.2.10 AV_{SS}

This is the ground pin of the A/D converter.

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

4.2.11 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

4.2.12 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.

4.2.13 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.

4.2.14 V_{DD}

Positive power supply pin

4.2.15 V_{SS}

Ground pin

4.2.16 V_{PP} (μPD78P018FY only)

A high voltage should be applied to this pin when the PROM programming mode is set and when the program is written or verified.

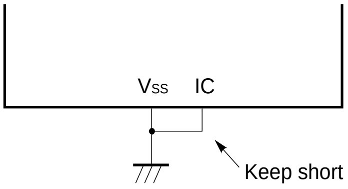
Directly connect this pin to V_{SS} in the normal operation mode.

4.2.17 IC (mask ROM model only)

The IC (Internally Connected) pin is used to set the μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, and 78018FY 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.**



Phase-out/Discontinued

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

Table 4-1 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 4-1.

Table 4-1. 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-A	I/O	Individually connect to V_{SS} via resistor
P02/INTP2			
P03/INTP3			
P04/XT1	16	Input	Connect to V_{DD}
P10/ANI0-P17/ANI7	11	I/O	Individually connect to V_{DD} or V_{SS}
P20/SI1	8-A		
P21/SO1	5-A		
P22/$\overline{\text{SCK1}}$	8-A		
P23/STB	5-A		
P24/BUSY	8-A		
P25/SI0/SB0/SDA0	10-A		
P26/SO0/SB1/SDA1			
P27/$\overline{\text{SCK0}}$/SCL			
P30/TO0	5-A		
P31/TO1			
P32/TO2			
P33/TI1	8-A		
P34/TI2			
P35/PCL	5-A		
P36/BUZ			
P37			
P40/AD0-P47/AD7	5-E	I/O	Individually connect to V_{DD} via resistor
P50/A8-P57/A15	5-A	I/O	Individually connect to V_{DD} or V_{SS} via resistor
P60-P63 (Mask ROM model)	13-B	I/O	Individually connect to V_{DD} via resistor
P60-P63 (PROM model)	13-D		
P64/$\overline{\text{RD}}$	5-A		Individually connect to V_{DD} or V_{SS} via resistor
P65/$\overline{\text{WR}}$			
P66/$\overline{\text{WAIT}}$			
P67/ASTB			
$\overline{\text{RESET}}$	2	Input	—
XT2	16	—	Open
AV_{REF}	—		Connect to V_{SS}
AV_{DD}			Connect to V_{DD}
AV_{SS}			Connect to V_{SS}
IC (Mask ROM model)			Directly connect to V_{SS}
V_{PP} (PROM model)			

Figure 4-1. I/O Circuits of Pins (1/2)

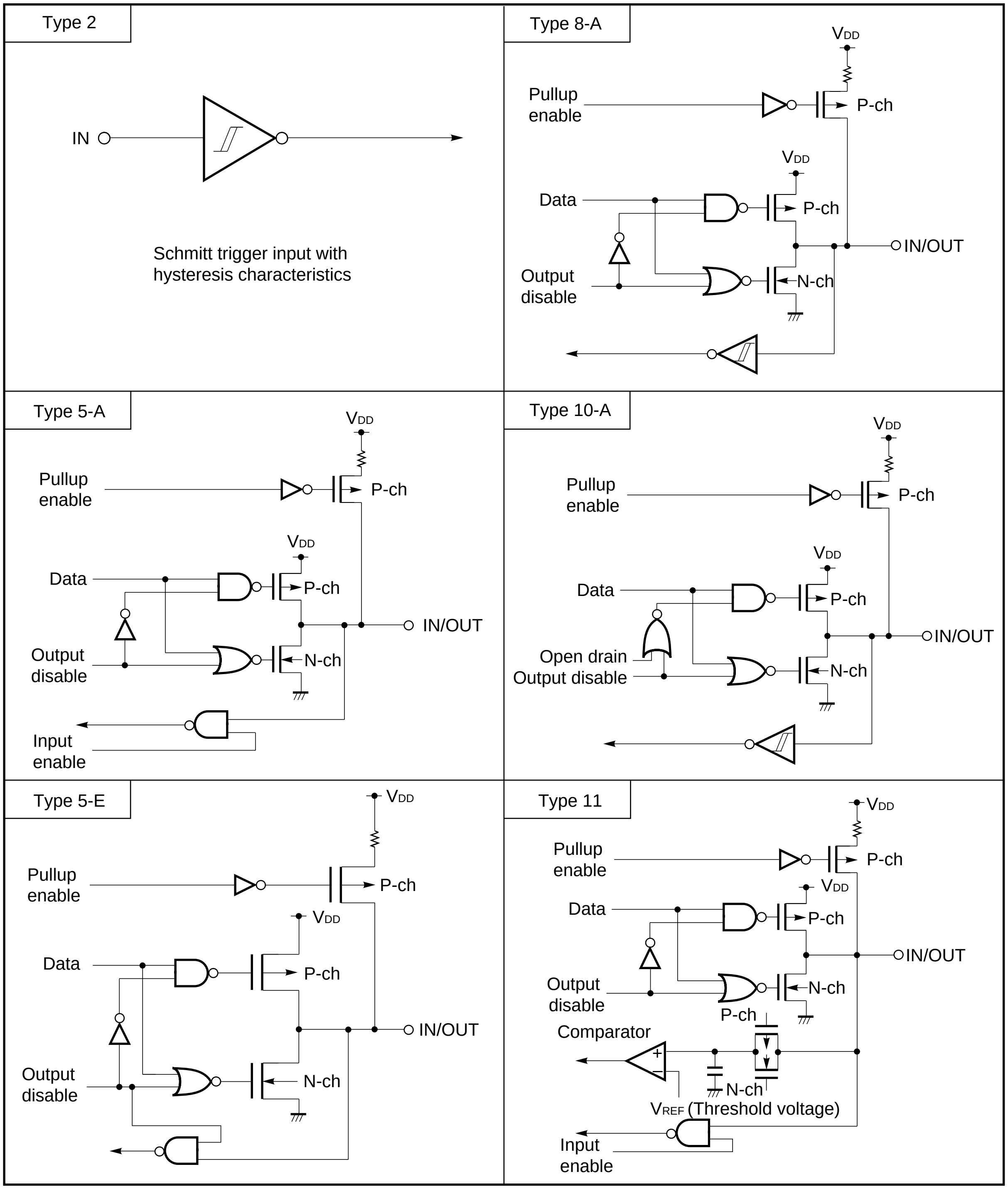
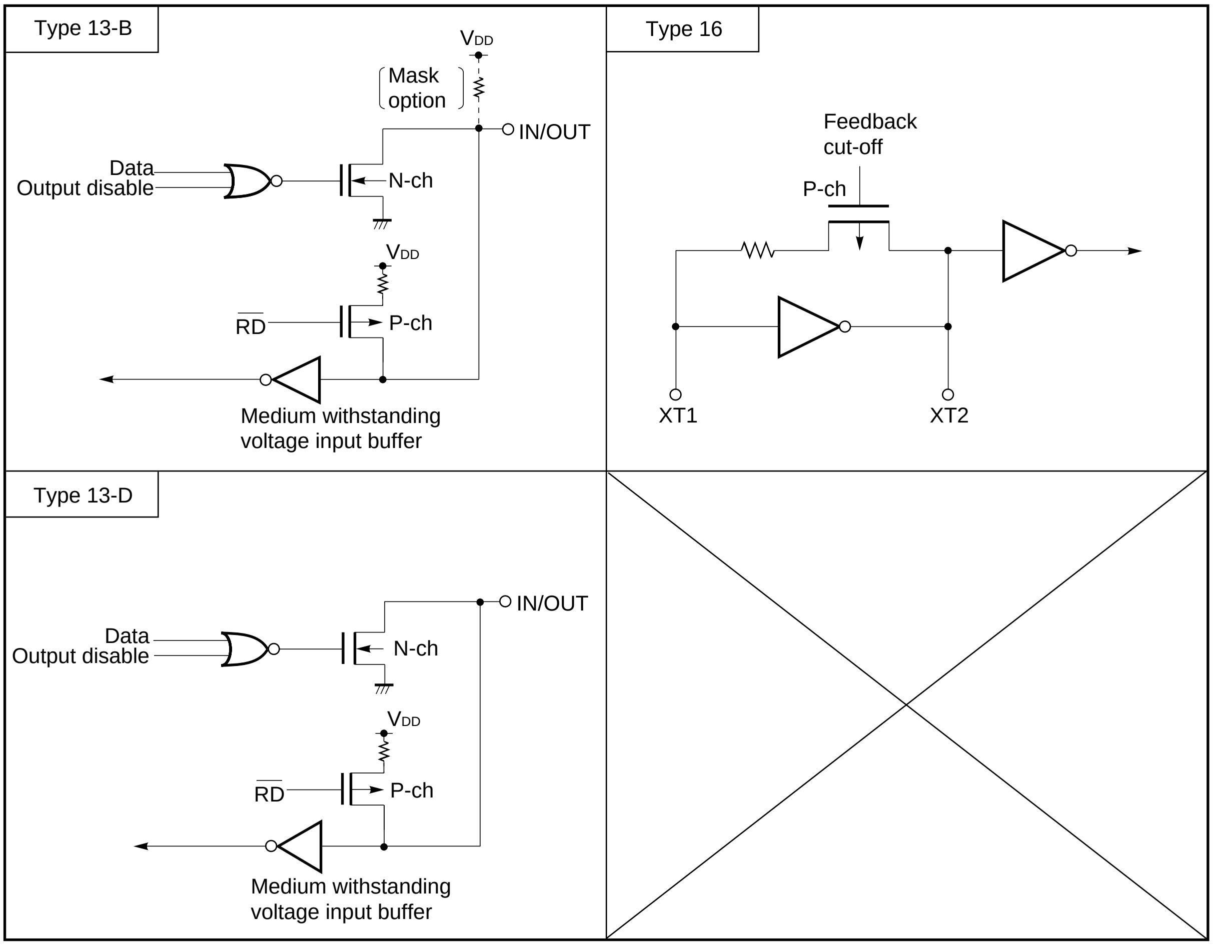


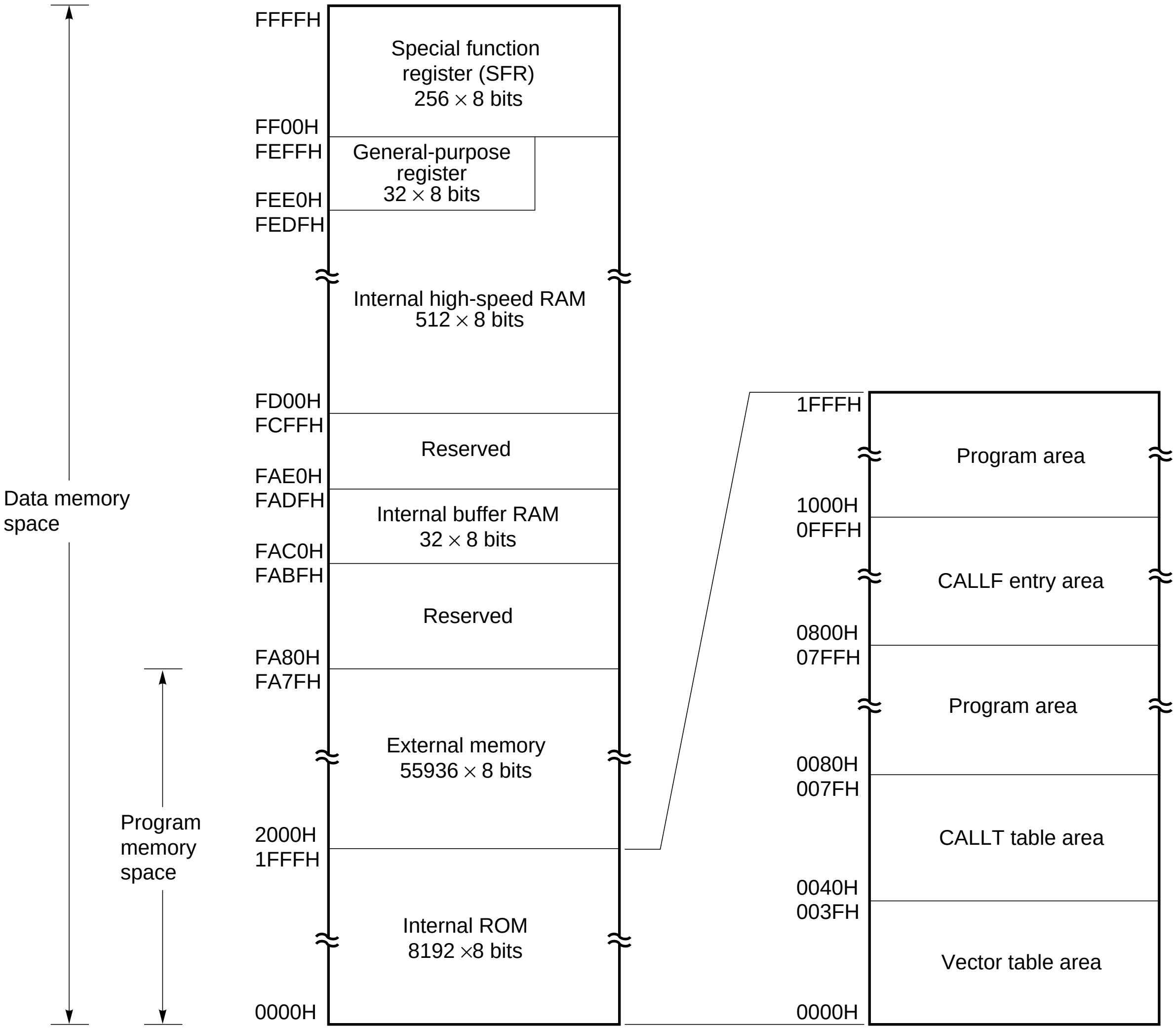
Figure 4-1. I/O Circuits of Pins (2/2)



5.1 Memory Space

Each model in the μPD78018F, 78018FY subseries can access a memory space of 64 KB. Figures 5-1 through 5-8 show memory maps of the respective models.

Figure 5-1. Memory Map (μPD78011F, 78011FY)



Phase-out/Discontinued

Figure 5-2. Memory Map (μPD78012F, 78012FY)

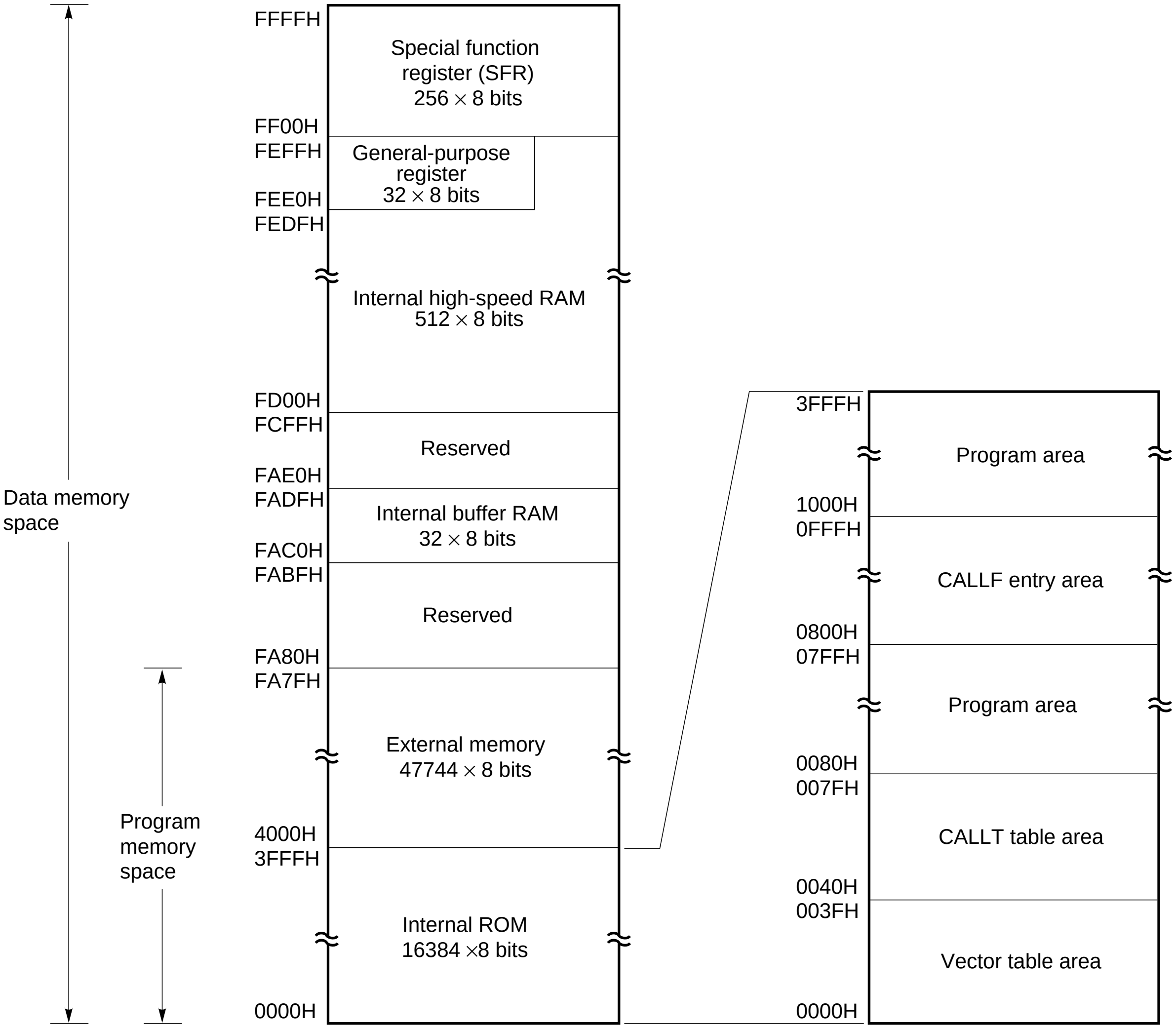
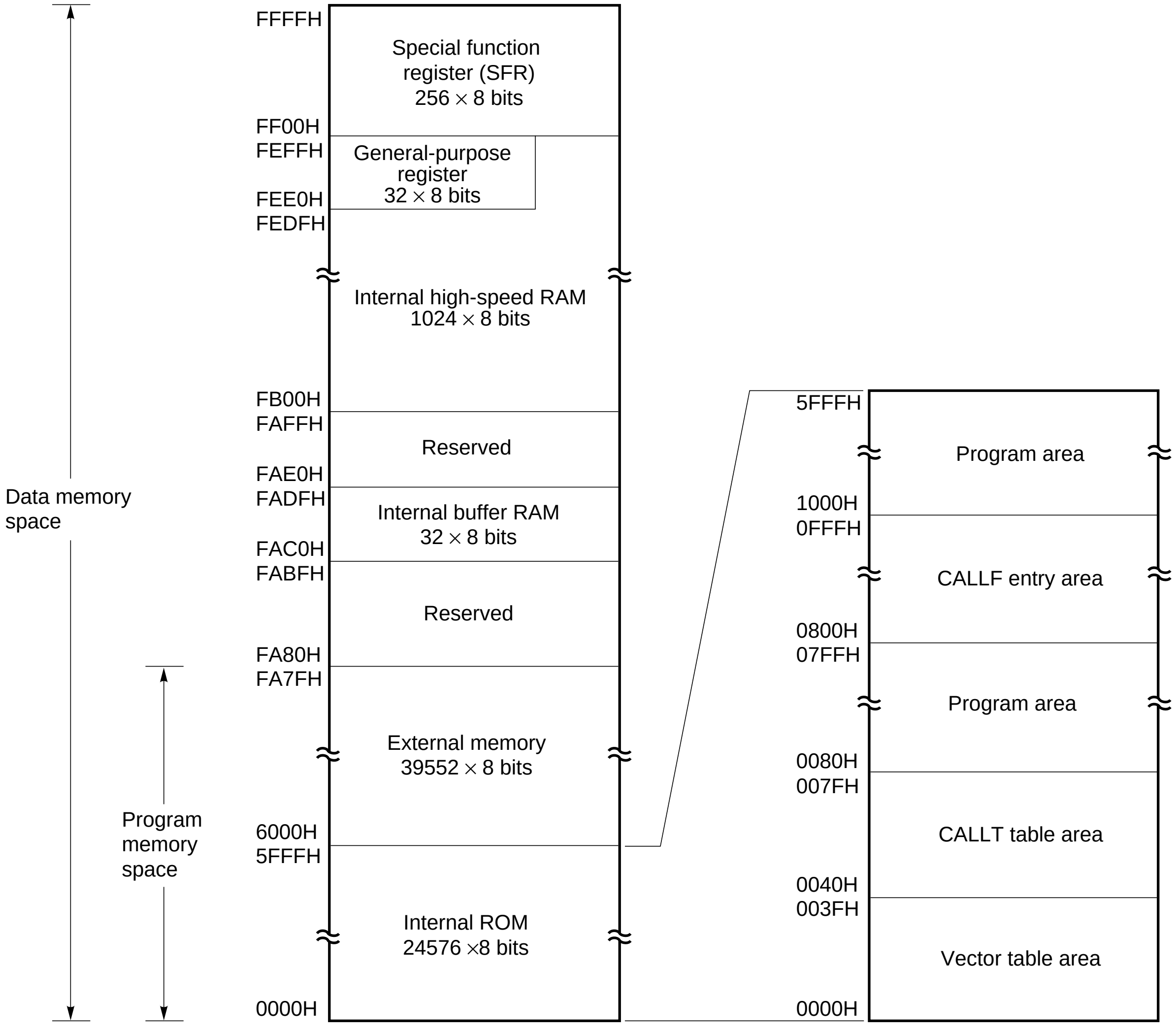
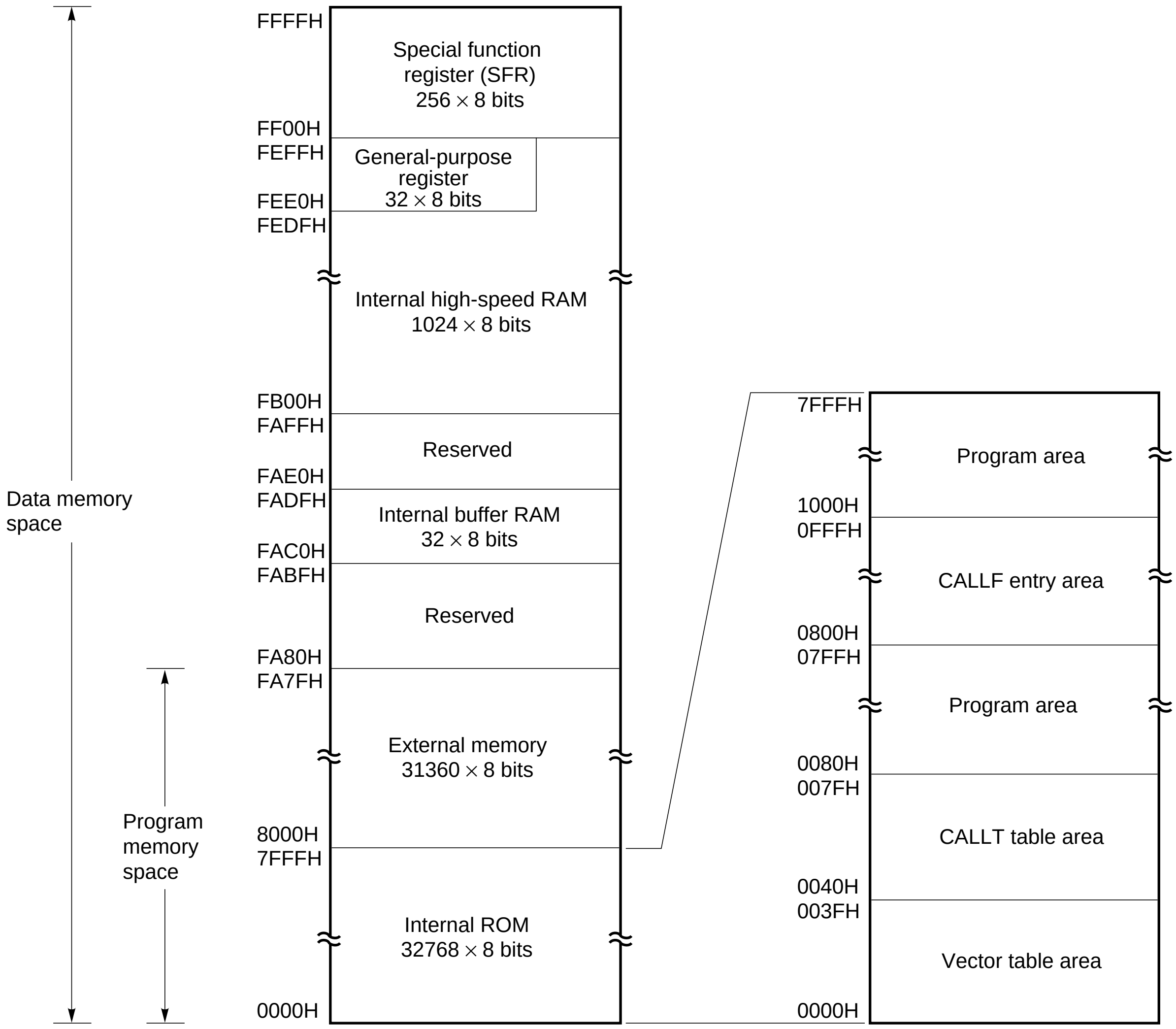


Figure 5-3. Memory Map (μPD78013F, 78013FY)



Phase-out/Discontinued

Figure 5-4. Memory Map (μPD78014F, 78014FY)



Phase-out/Discontinued

Figure 5-5. Memory Map (μPD78015F, 78015FY)

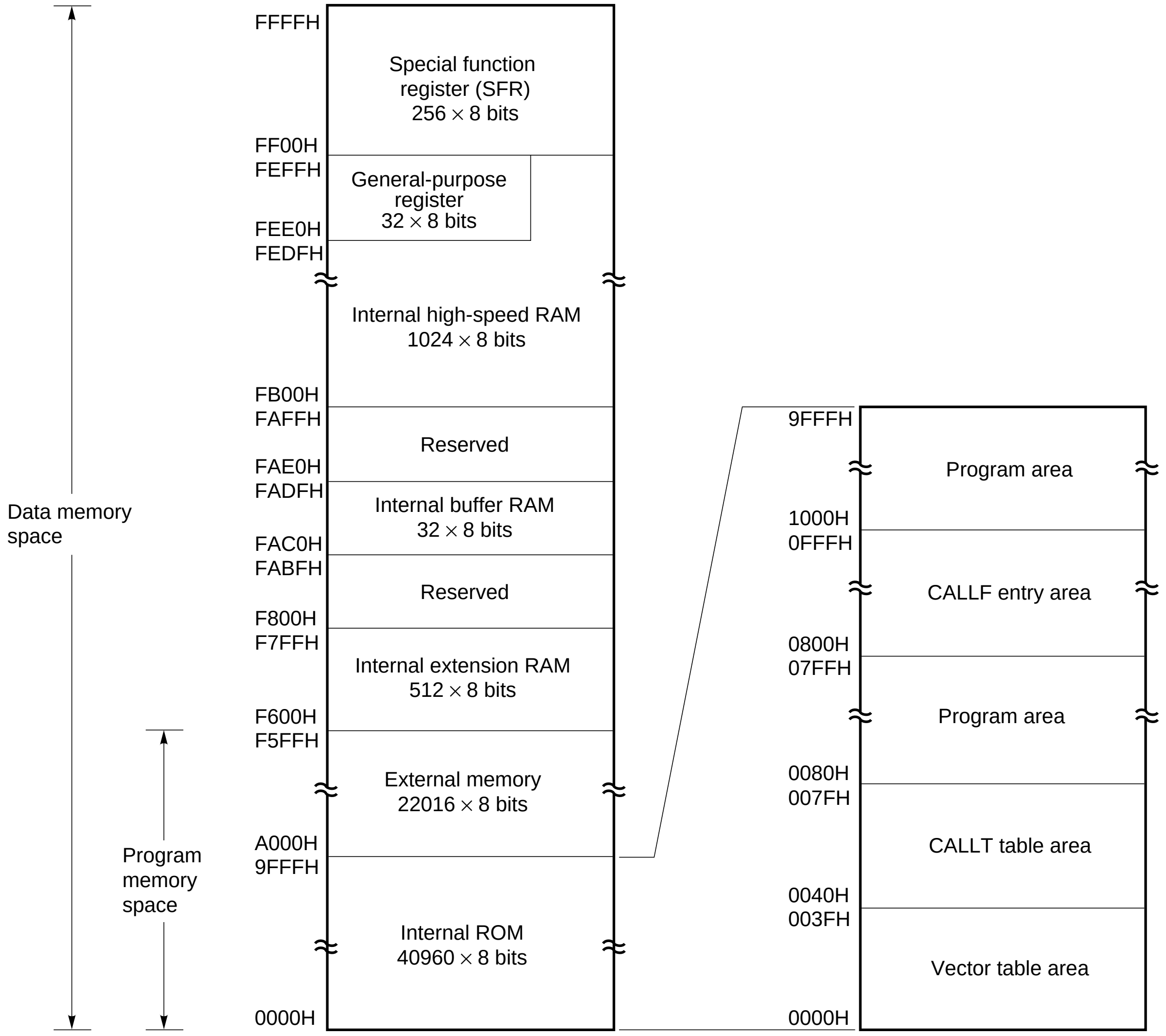


Figure 5-6. Memory Map (μPD78016F, 78016FY)

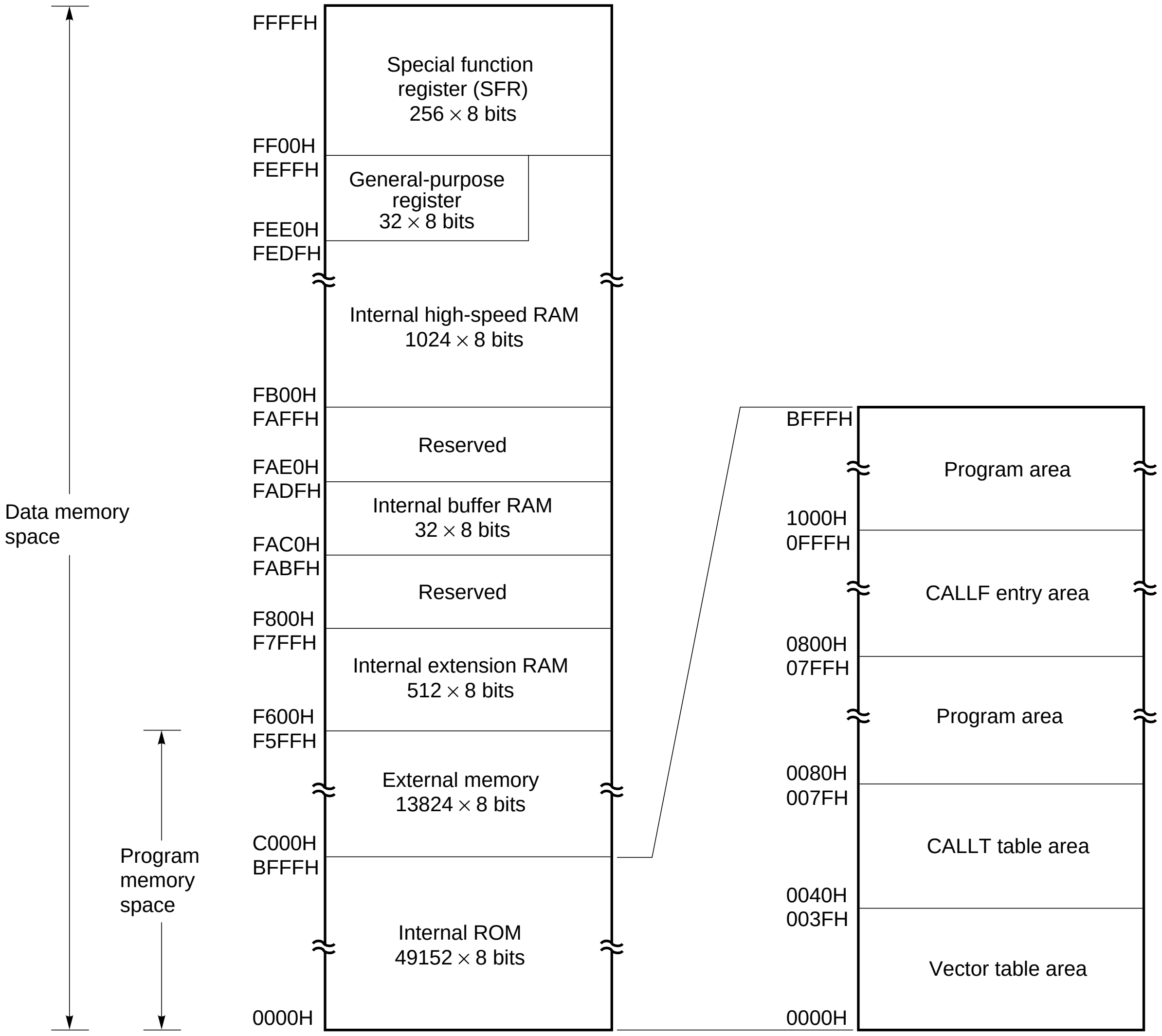
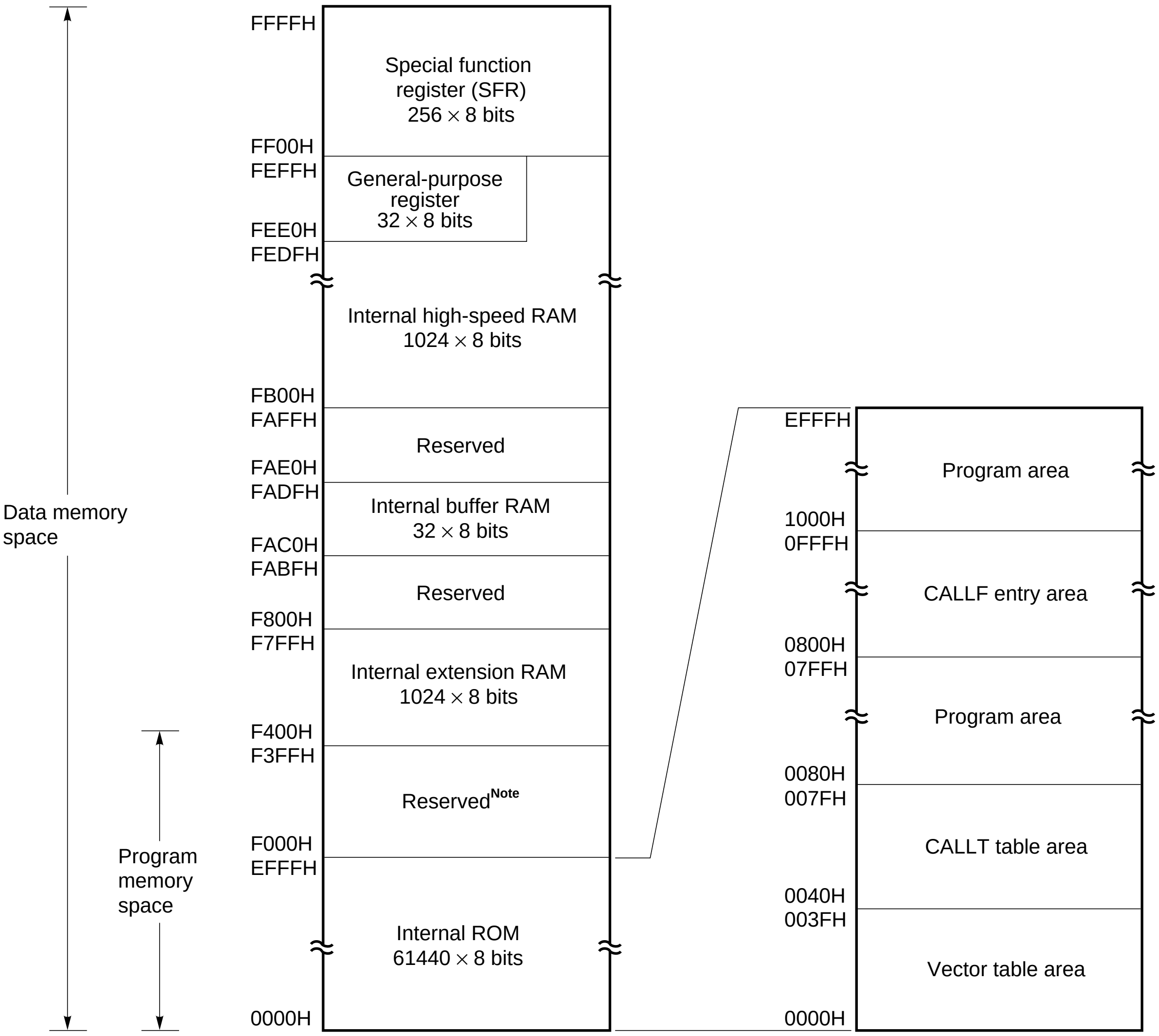
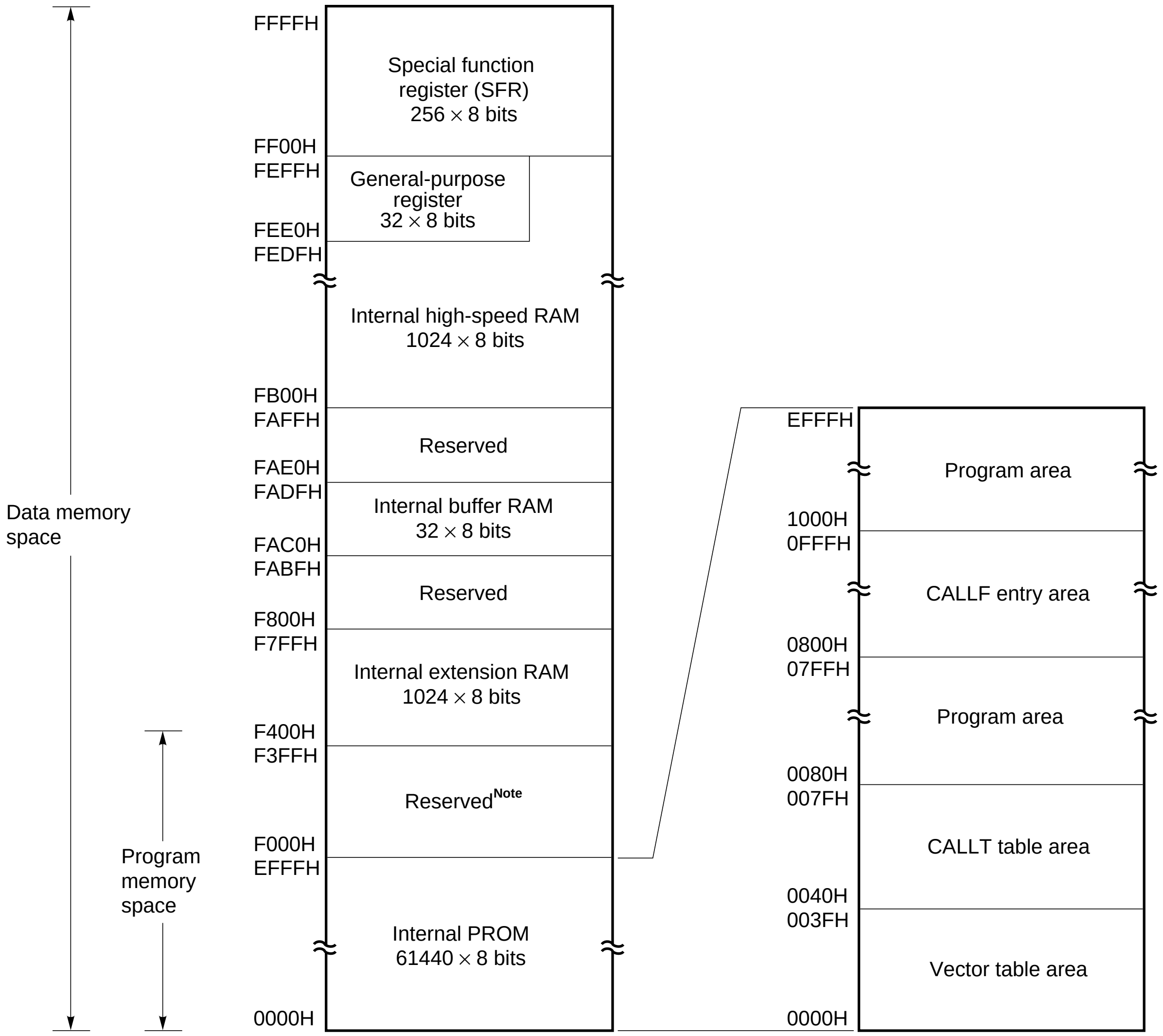


Figure 5-7. Memory Map (μPD78018F, 78018FY)



Note When the internal ROM capacity is 60 KB, the area of F000H-F3FFH cannot be used.
The area of F000H-F3FFH can be used as an external memory by setting the internal ROM capacity to 56 KB or less with the memory size select register (IMS).

Figure 5-8. Memory Map (μPD78P018F, 78P018FY)



Note When the internal PROM capacity is 60 KB, the area of F000H-F3FFH cannot be used. The area of F000H-F3FFH can be used as an external memory by setting the internal PROM capacity to 56 KB or less with the memory size select register (IMS).

5.1.1 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 μPD78018F, 78018FY subseries is provided with the following internal ROM (or PROM):

Table 5-1. Internal ROM Capacity

Part Number	Structure	Capacity
μPD78011F, 78011FY	Mask ROM	8192 × 8 bits (0000H-1FFFH)
μPD78012F, 78012FY		16384 × 8 bits (0000H-3FFFH)
μPD78013F, 78013FY		24576 × 8 bits (0000H-5FFFH)
μPD78014F, 78014FY		32768 × 8 bits (0000H-7FFFH)
μPD78015F, 78015FY		40960 × 8 bits (0000H-9FFFH)
μPD78016F, 78016FY		49152 × 8 bits (0000H-BFFFH)
μPD78018F, 78018FY		61440 × 8 bits (0000H-EFFFH)
μPD78P018F, 78P018FY	PROM	61440 × 8 bits (0000H-EFFFH)

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

(1) Vector table area

A 64-byte area of addresses 0000H-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 low-order 8 bits are stored in an even address, and the high-order 8 bits are stored in an odd address.

Table 5-2. 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-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-0FFFH, a subroutine can be directly called by using a 2-byte call instruction (CALLF).

5.1.2 Internal data memory space

The μPD78018F, 78018FY subseries have the following RAMs:

(1) Internal high-speed RAM

The μPD78018F and 78018FY subseries are provided with the following internal high-speed RAM.

Table 5-3. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD78011F, 78011FY	512 × 8 bits (FD00H-FEFFH)
μPD78012F, 78012FY	
μPD78013F, 78013FY	1024 × 8 bits (FB00H-FEFFH)
μPD78014F, 78014FY	
μPD78015F, 78015FY	
μPD78016F, 78016FY	
μPD78018F, 78018FY	
μPD78P018F, 78P018FY	

A 32-byte area of addresses FEE0H-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.

(2) Internal extension RAM

Only the following products of the μPD78018F, 78018FY subseries are provided with the following internal extension RAM:

Table 5-4. Internal Extension RAM Capacity

Part Number	Internal Extension RAM
μPD78015F, 78015FY	512 × 8 bits (F600H-F7FFH)
μPD78016F, 78016FY	
μPD78018F, 78018FY	1024 × 8 bits (F400H-F7FFH)
μPD78P018F, 78P018FY	

(3) Internal buffer RAM

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

5.1.3 Special function register (SFR) area

Special function registers (SFRs) of on-chip peripheral hardware are allocated to an area of FF00H-FFFFH (refer to **5.2.3 Special function registers (SFRs) Table 5-7 Special Function Register List**).

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

5.1.4 External memory space

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

5.2 Processor Registers

The μPD78018F, 78018FY subseries are provided with the following processor registers:

5.2.1 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 5-9. Program Counter Configuration

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

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of 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 5-10. Program Status Word Configuration

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

When IE = 0, all the interrupts are disabled except the non-maskable interrupt.

When IE = 1, the interrupts are enabled. At this time, accepting an interrupt is controlled by the in-service priority flag (ISP), interrupt mask flag corresponding to each interrupt, and interrupt priority specification flag.

These flags are reset to 0 when the DI instruction is executed or when an interrupt request is accepted, 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 ISP = 0, the vectored interrupt which is assigned a low priority by the priority specification flag registers (PR0L and PR0H) (refer to **18.3 (3) Priority specification flag registers (PR0L, PR0H)**)should not be accepted. Whether the interrupt is actually accepted is controlled by the status of the interrupt enable flag (IE).

(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. The internal high-speed RAM area of each product is as follows:

Table 5-5. Internal High-Speed RAM Area

Part Number	Internal High-Speed RAM Area
μPD78011F, 78011FY, 78012F, 78012FY	FD00H-FEFFH
μPD78013F, 78013FY, 78014F, 78014FY μPD78015F, 78015FY, 78016F, 78016FY μPD78018F, 78018FY, 78P018F, 78P018FY	FB00H-FEFFH

Figure 5-11. Stack Pointer Configuration

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

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 5-12 and 5-13.

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 5-12. Data Saved to Stack Memory

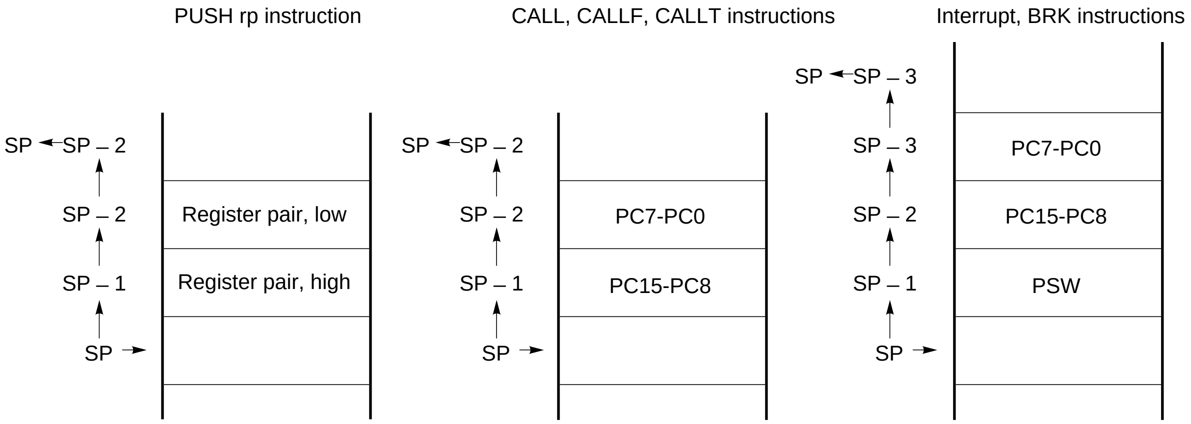
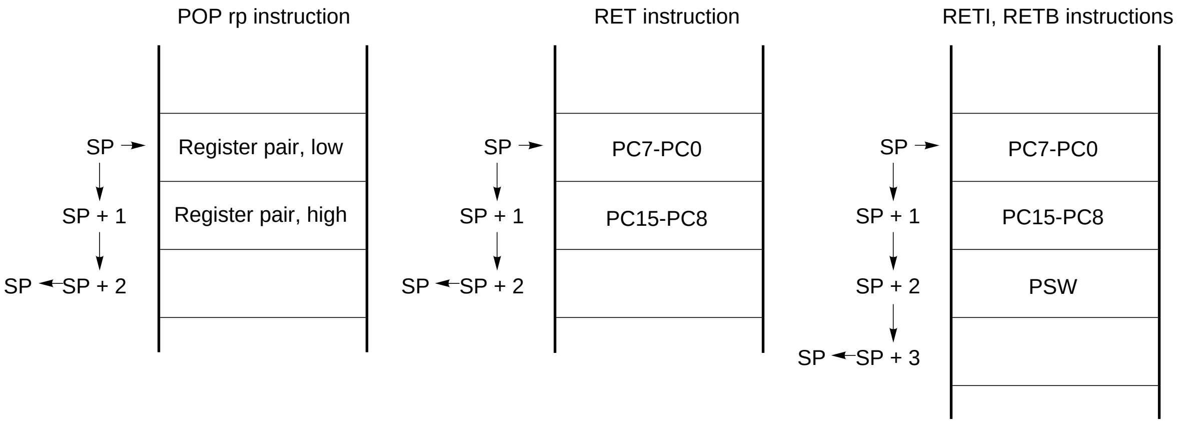


Figure 5-13. Data Restored from Stack Memory



5.2.2 General-purpose registers

General-purpose registers are mapped to the specific addresses of the data memory (FEE0H-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).

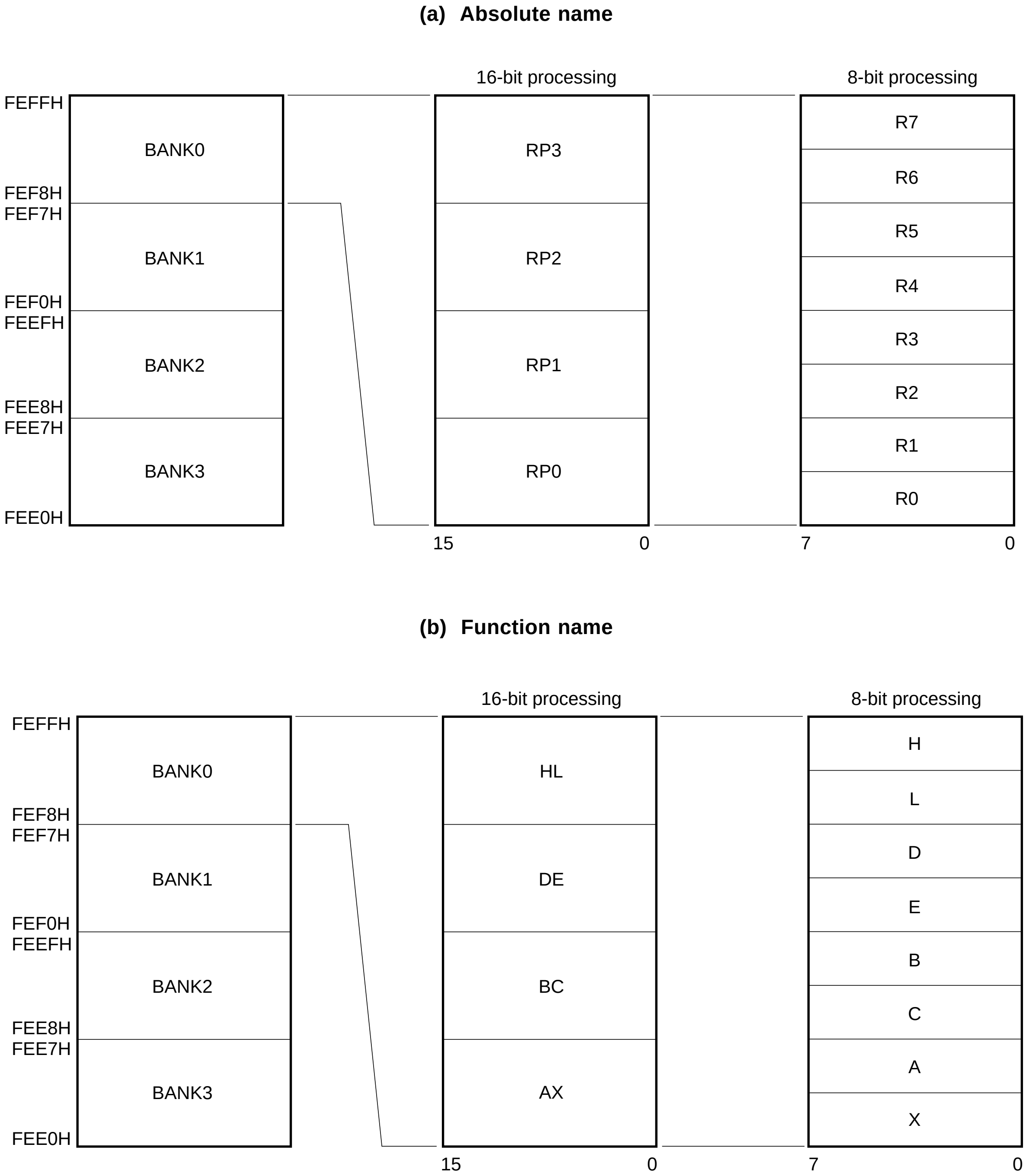
Each register can be written not only in function name (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) but also in absolute name (R0-R7, RP0-RP3).

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 processing.

Table 5-6. 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 5-14. General-Purpose Register Configuration



(a) Absolute name

(b) Function name

5.2.3 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-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 written 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 written 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 written as the operand (sfrp) of a 16-bit manipulation instruction. To specify address, write an even address.

Table 5-7 lists the special function register. The meanings of the symbols in this table are as follows:

- **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 written as the operands of instructions when the RA78K/0, ID78K0-NS, ID78K0, 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**
  ○ indicates the bit units (1, 8, 16) in which the special function register in question can be manipulated. – indicates the bit units that cannot be manipulated.
- **At reset**
  Indicates the status of the special function register when the $\overline{\text{RESET}}$ signal is input.

Phase-out/Discontinued

Table 5-7. Special Function Register List (1/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Units for Manipulation			At 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		○	○	–	
FF38H FF39H	Correction address register 0 Note	CORAD0		–	–	○	0000H
FF3AH FF3BH	Correction address register 1 Note	CORAD1		–	–	○	
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		○	○	–	

Note Only μPD78015F, 78015FY, 78016F, 78016FY, 78018F, 78018FY, 78P018F, and 78P018FY have this register.

Table 5-7. Special Function Register List (2/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Units for Manipulation			At 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 transmit/receive control register	ADTC			○	○	–	
FF6AH	Automatic data transmit/receive address pointer	ADTP			–	○	–	
FF6BH	Automatic transmit/receive interval specification register	ADTI			○	○	–	
FF80H	A/D converter mode register	ADM			○	○	–	01H
FF84H	A/D converter input select register	ADIS			–	○	–	00H
FF8AH	Correction control register[Note 1]	CORCN			○	○	–	
FFD0H | FFDFH	External access area[Note 2]				○	○	–	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 3
FFF4H	Internal extension RAM size select register	IXS			–	○	–	
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. Provided to μPD78015F, 78015FY, 78016F, 78016FY, 78018F, 78018FY, 78P018F and 78P018FY only.

2. The external access area cannot be accessed by SFR addressing. Access this area by direct addressing.

3. The value on reset for the memory size selected register (IMS), internal extension RAM size select register (IXS) differs depending on the product, as follows:

	μPD78011F μPD78011FY	μPD78012F μPD78012FY	μPD78013F μPD78013FY	μPD78014F μPD78014FY	μPD78015F μPD78015FY	μPD78016F μPD78016FY	μPD78018F μPD78018FY	μPD78P018F μPD78P018FY
IMS	42H	44H	C6H	C8H	CAH	CCH	CFH	
IXS	0CH				0BH		0AH	

To use a mask ROM version, do not set to IMS and IXS a value other than that on reset.

Except, however, when the external device extension function is used with the μPD78018F or 78018FY.

5.3 Addressing Instruction Address

An instruction address is determined by the contents of the program counter (PC). The contents of the PC are 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)**).

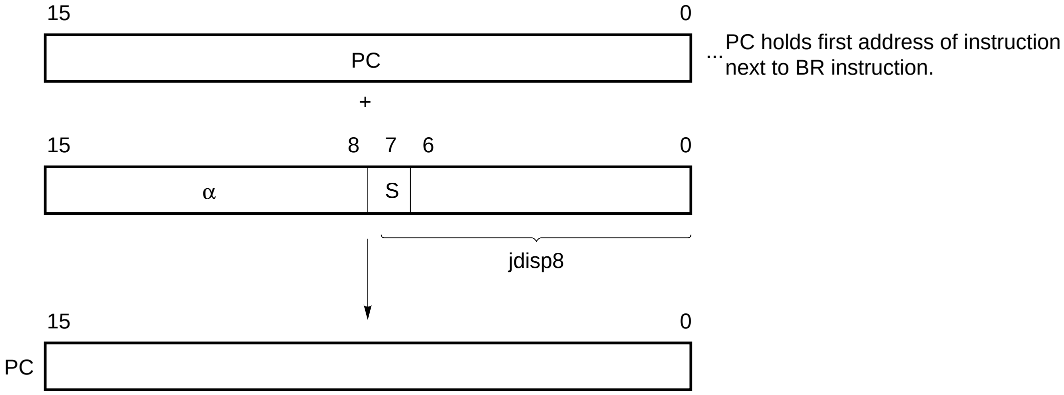
5.3.1 Relative addressing

[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. That is, using relative addressing, the program branches in the range –128 to +127 relative to the first address of the next instruction.

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

[Operation]


15
0
PC
...PC holds first address of instruction next to BR instruction.
+
15
8
7
6
0
α
S
jdisp8
15
0
PC

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

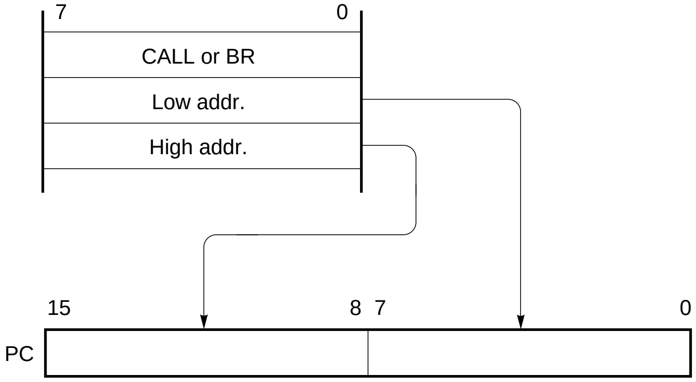
5.3.2 Immediate addressing

[Function]

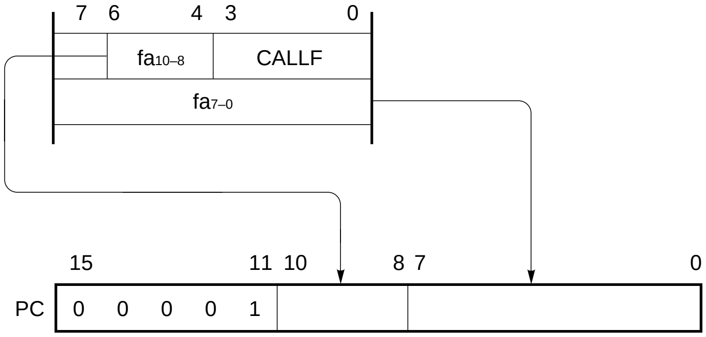
The immediate data in an instruction word is transferred to the program counter (PC), and execution branches. This addressing is used when the “CALL !addr16”, “BR !addr16”, or “CALLF !addr11” instruction is executed. The CALL !addr16, BR !addr16 instructions can be used to branch to any location in the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

[Operation]

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



When “CALLF !addr11“ instruction is executed

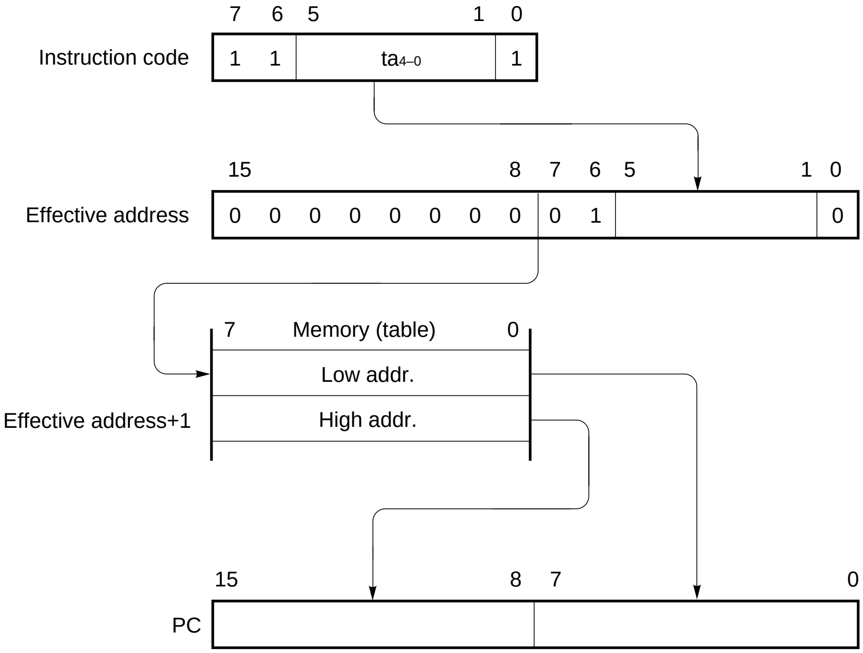


5.3.3 Table indirect addressing

[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. This instruction references an address stored in the memory table between 40H through 7FH, and can be used to branch to any location in the memory.

[Operation]



Phase-out/Discontinued

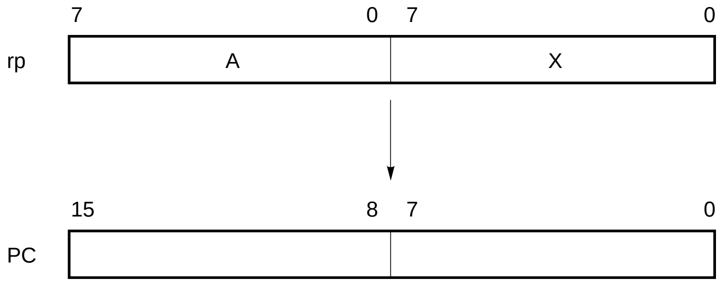
5.3.4 Register addressing

[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]



5.4 Addressing of Operand Address

5.4.1 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 **5.3 Addressing Instruction Address**).

To address the memory to be manipulated when an instruction is executed, the μPD78018F and 78018FY subseries have many addressing modes to improve the operability. In particular, the area assigned as the data memory can be addressed using special addressing modes in accordance with the functions of the area, such as special function registers (SFRs) and general-purpose registers. Figures 5-15 through 5-22 illustrate how the data memory is addressed.

Figure 5-15. Data Memory Addressing (μPD78011F, 78011FY)

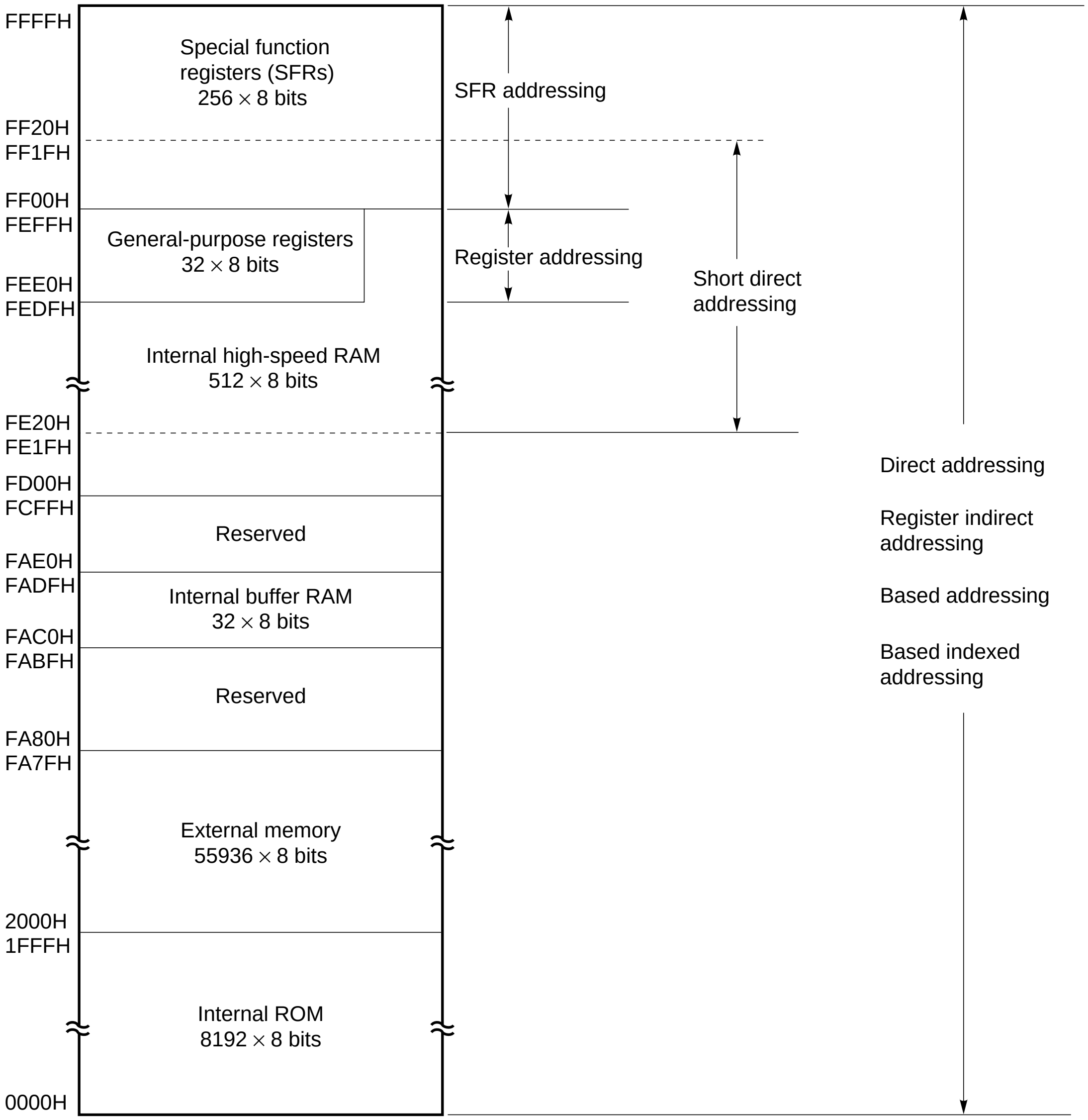

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 512 × 8 bits
FE20H
FE1FH
FD00H
FCFFH
Reserved
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
FAC0H
FABFH
Reserved
FA80H
FA7FH
External memory 55936 × 8 bits
2000H
1FFFH
Internal ROM 8192 × 8 bits
0000H
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing

Figure 5-16. Data Memory Addressing (μPD78012F, 78012FY)

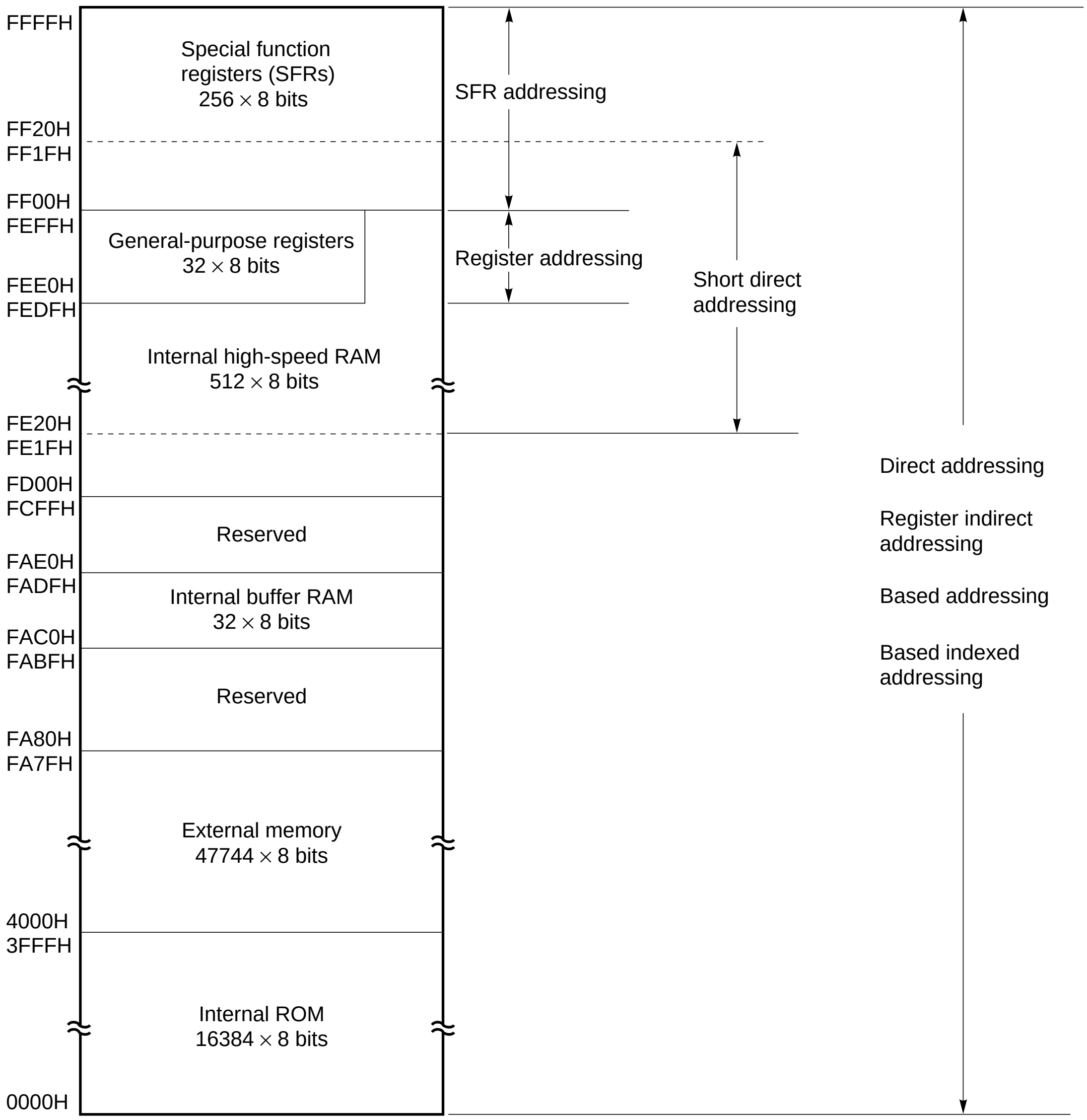
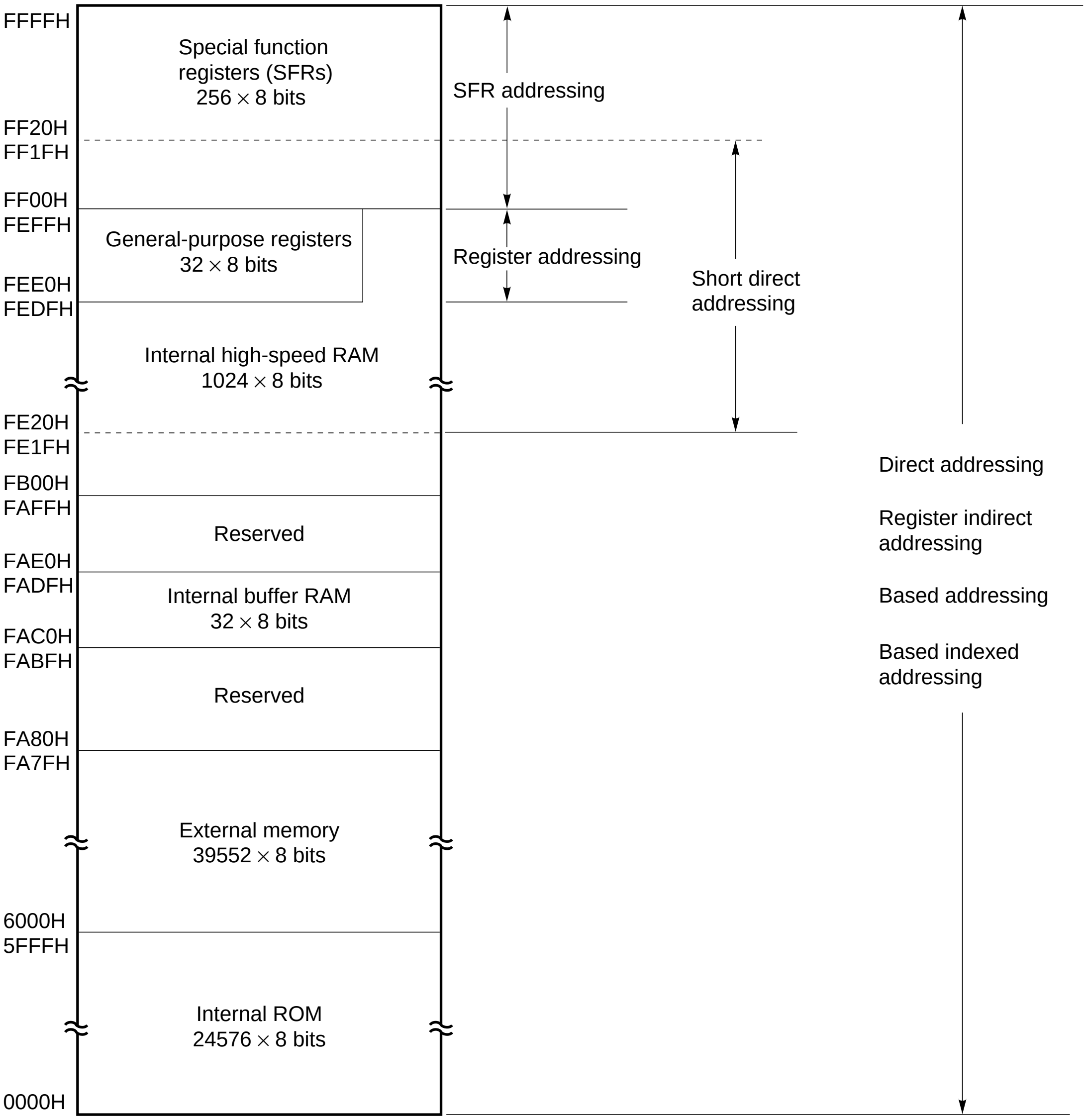

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 512 × 8 bits
FE20H
FE1FH
FD00H
FCFFH
Reserved
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
FAC0H
FABFH
Reserved
FA80H
FA7FH
External memory 47744 × 8 bits
4000H
3FFFH
Internal ROM 16384 × 8 bits
0000H
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing

Figure 5-17. Data Memory Addressing (μPD78013F, 78013FY)



FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
FE20H
FE1FH
FB00H
FAFFH
Reserved
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
FAC0H
FABFH
Reserved
FA80H
FA7FH
External memory 39552 × 8 bits
6000H
5FFFH
Internal ROM 24576 × 8 bits
0000H
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing

Figure 5-18. Data Memory Addressing (μPD78014F, 78014FY)

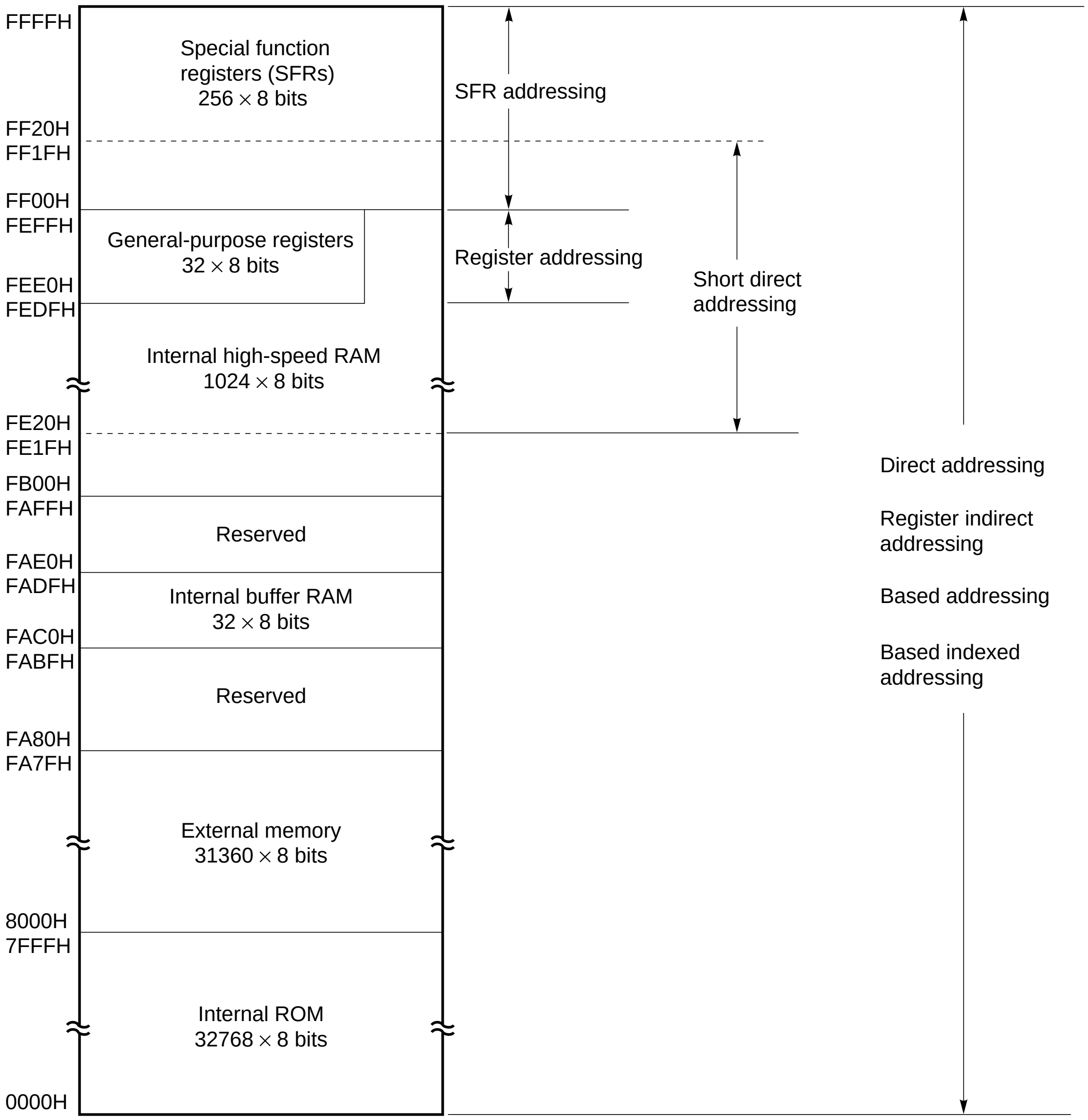

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
FE20H
FE1FH
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing
FB00H
FAFFH
Reserved
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
FAC0H
FABFH
Reserved
FA80H
FA7FH
External memory 31360 × 8 bits
8000H
7FFFH
Internal ROM 32768 × 8 bits
0000H

Figure 5-19. Data Memory Addressing (μPD78015F, 78015FY)

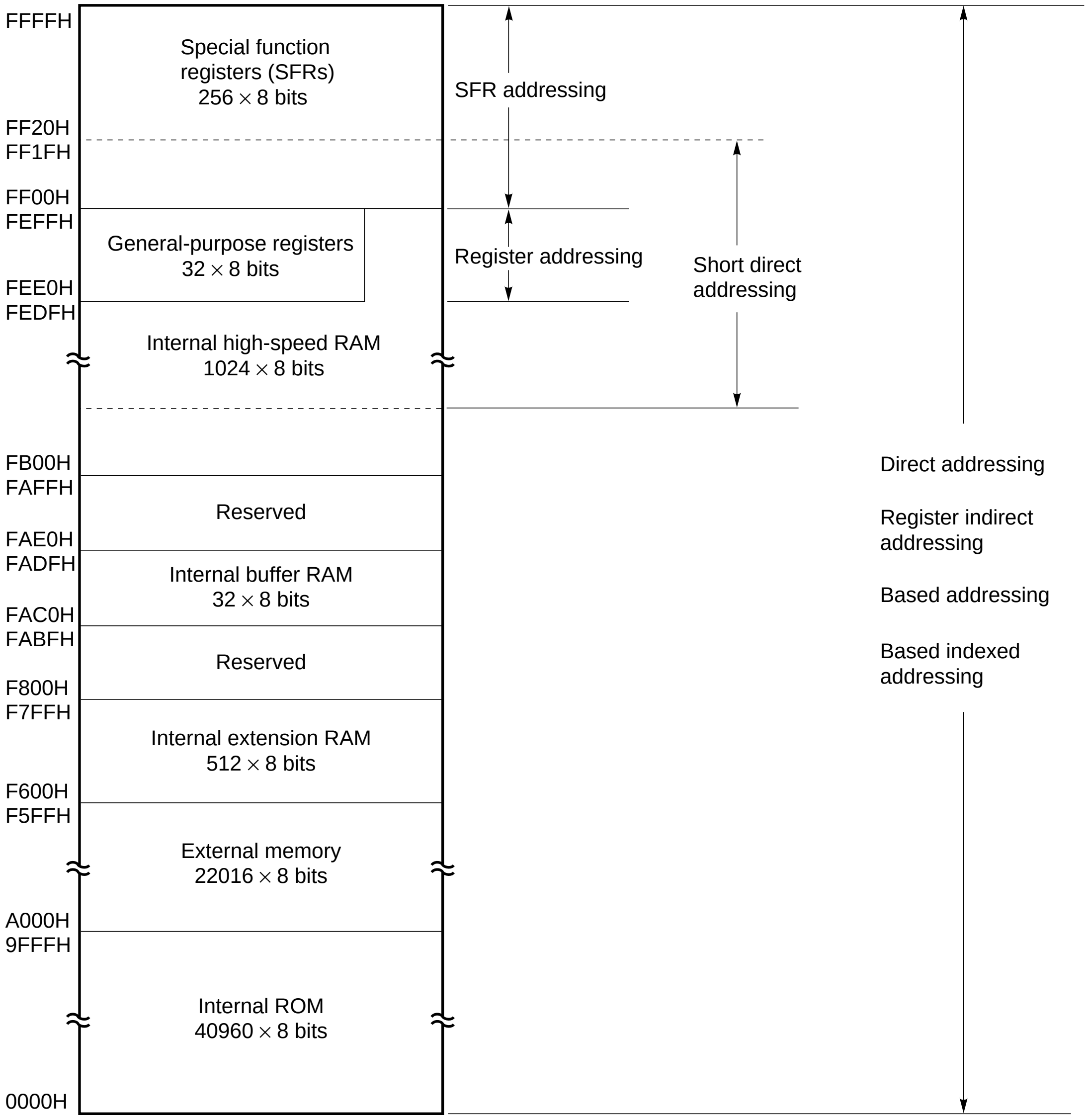

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
FB00H
FAFFH
Reserved
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
FAC0H
FABFH
Reserved
F800H
F7FFH
Internal extension RAM 512 × 8 bits
F600H
F5FFH
External memory 22016 × 8 bits
A000H
9FFFH
Internal ROM 40960 × 8 bits
0000H
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing

Figure 5-20. Data Memory Addressing (μPD78016F, 78016FY)

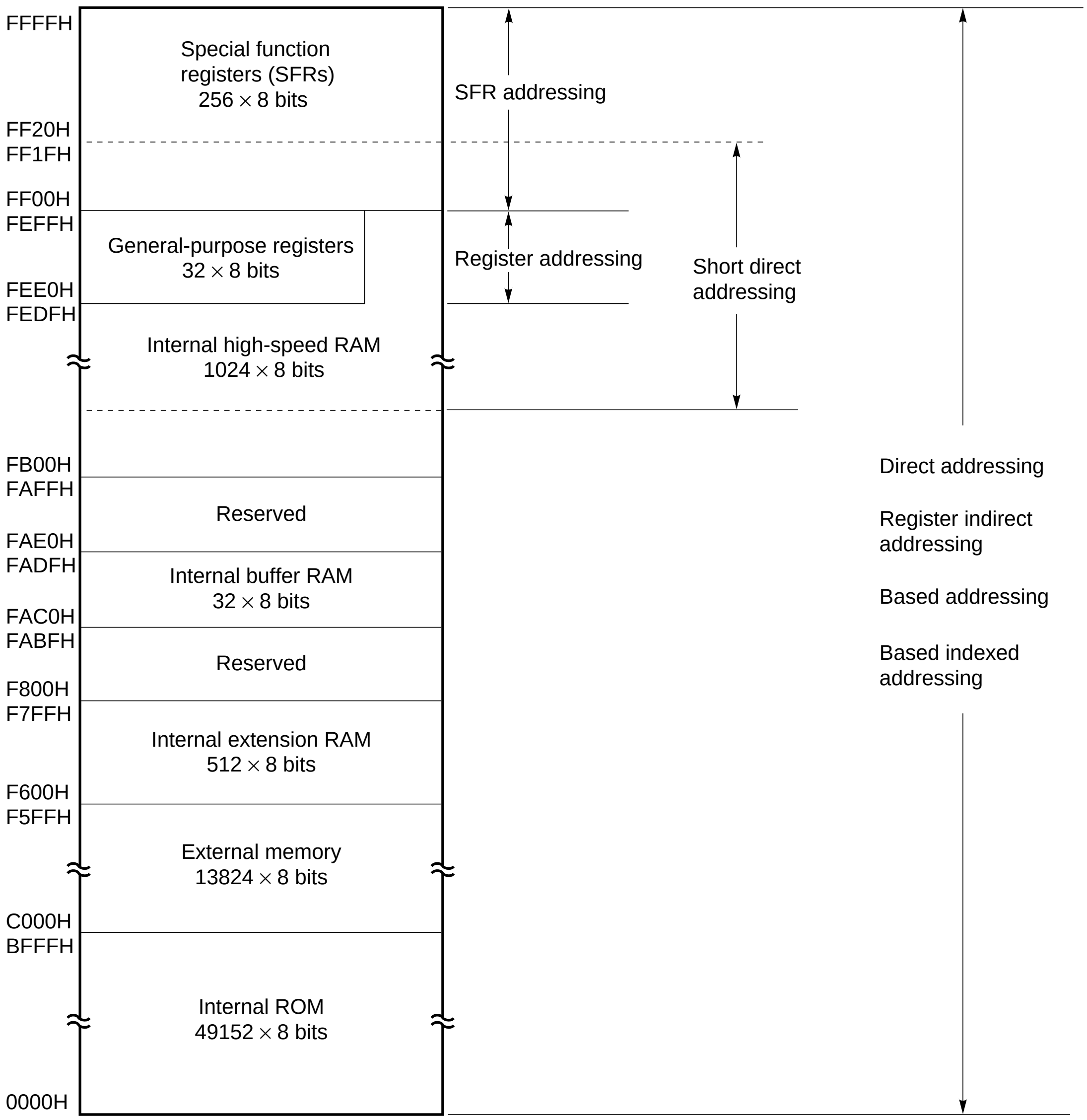
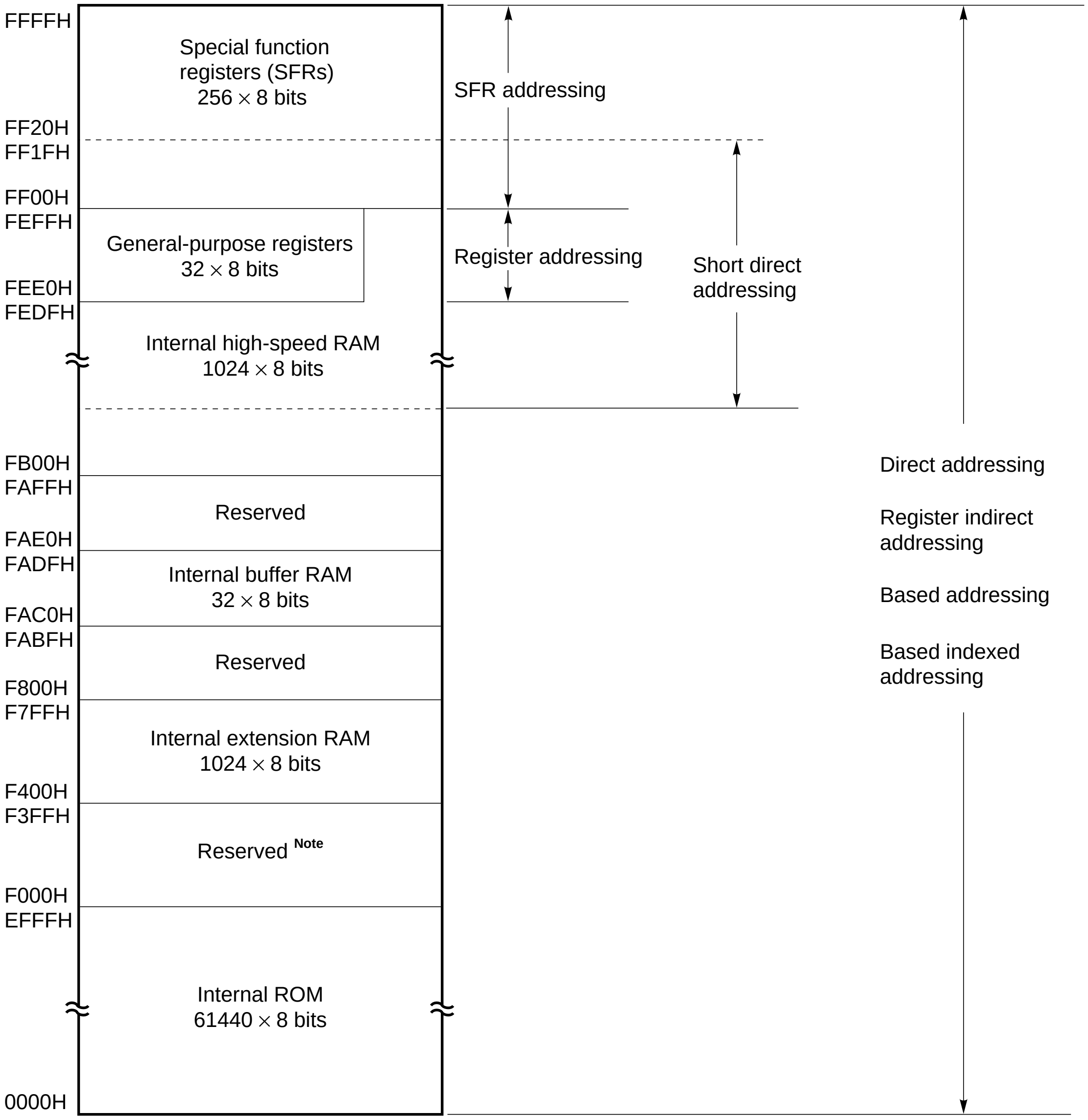
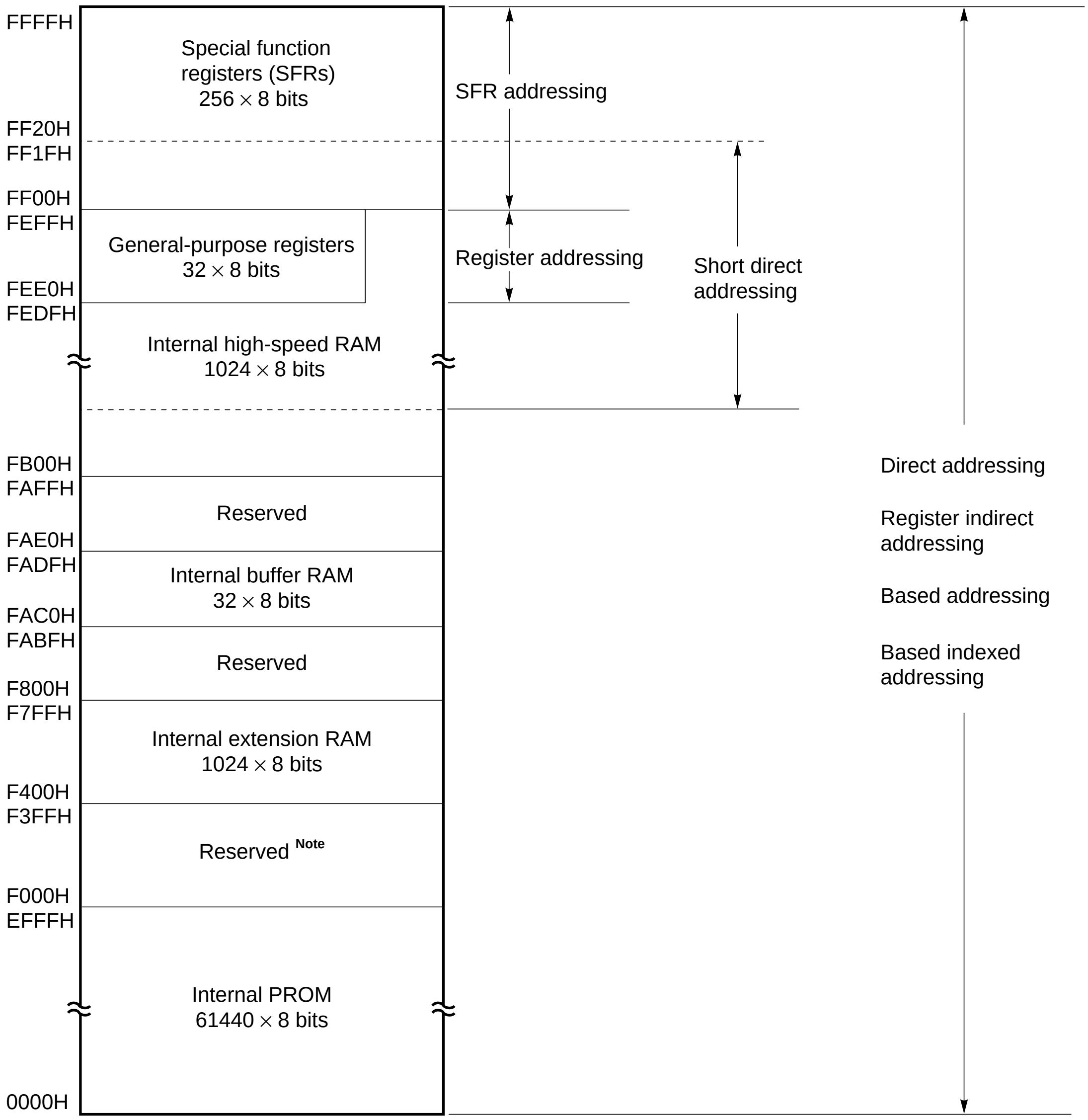

FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
FB00H
FAFFH
Direct addressing
Reserved
Register indirect addressing
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
Based addressing
FAC0H
FABFH
Reserved
Based indexed addressing
F800H
F7FFH
Internal extension RAM 512 × 8 bits
F600H
F5FFH
External memory 13824 × 8 bits
C000H
BFFFH
Internal ROM 49152 × 8 bits
0000H

Figure 5-21. Data Memory Addressing (μPD78018F, 78018FY)


FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
FB00H
FAFFH
Direct addressing
Reserved
Register indirect addressing
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
Based addressing
FAC0H
FABFH
Reserved
Based indexed addressing
F800H
F7FFH
Internal extension RAM 1024 × 8 bits
F400H
F3FFH
Reserved Note
F000H
EFFFH
Internal ROM 61440 × 8 bits
0000H

Note If the internal ROM capacity is 60 KB, an area of F000H-F3FFH cannot be used.
By setting the internal ROM capacity to 56 KB or less with the memory size select register (IMS), an area of F000H-F3FFH can be used as an external memory area.

Figure 5-22. Data Memory Addressing (μPD78P018F, 78P018FY)


FFFFH
Special function registers (SFRs) 256 × 8 bits
SFR addressing
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
Register addressing
Short direct addressing
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
FB00H
FAFFH
Reserved
FAE0H
FADFH
Internal buffer RAM 32 × 8 bits
FAC0H
FABFH
Reserved
F800H
F7FFH
Internal extension RAM 1024 × 8 bits
F400H
F3FFH
Reserved Note
F000H
EFFFH
Internal PROM 61440 × 8 bits
0000H
Direct addressing
Register indirect addressing
Based addressing
Based indexed addressing

Note If the internal PROM capacity is 60 KB, an area of F000H-F3FFH cannot be used.
By setting the internal PROM capacity to 56 KB or less with the memory size select register (IMS), an area of F000H-F3FFH can be used as an external memory area.

5.4.2 Implied addressing

[Function]

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

Of the instruction words of the μPD78018F and 78018FY 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.

5.4.3 Register addressing

[Function]

This addressing mode is used to access a general-purpose register as an operand. The register to be accessed is specified by the register bank select flags (RBS0 and RBS1) and the register specification code (Rn and RPn) in the instruction code.

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]

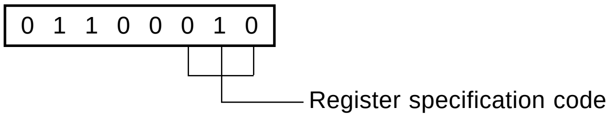
Representation	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

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

[Example]

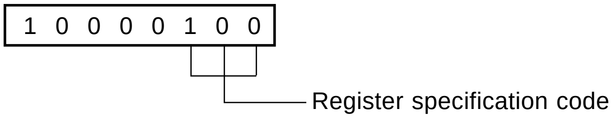
MOV A, C; To select C register as r

Instruction code



INCW DE; To select DE register pair as rp

Instruction code



5.4.4 Direct addressing

[Function]

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

[Operand Format]

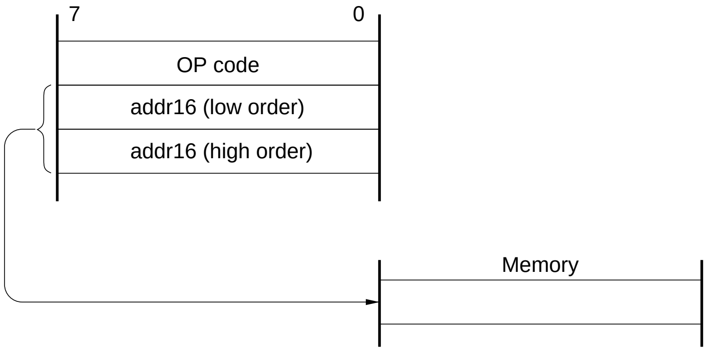
Representation	Description
addr16	Label or 16-bit immediate data

[Example]

MOV A, !0FE00H; To specify FE00H as !addr16

Instruction code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Operation]



5.4.5 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.

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

The SFR area (FF00H-FF1FH) to which short direct addressing is applied is one part of all the SFR areas. In this area, ports, compare and capture registers of timer/event counters that are frequently accessed on program are mapped. 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-FFH, and 1 if the data is in a range of 00H-1FH. Refer to **[Operation]** on the next page.

[Operand Format]

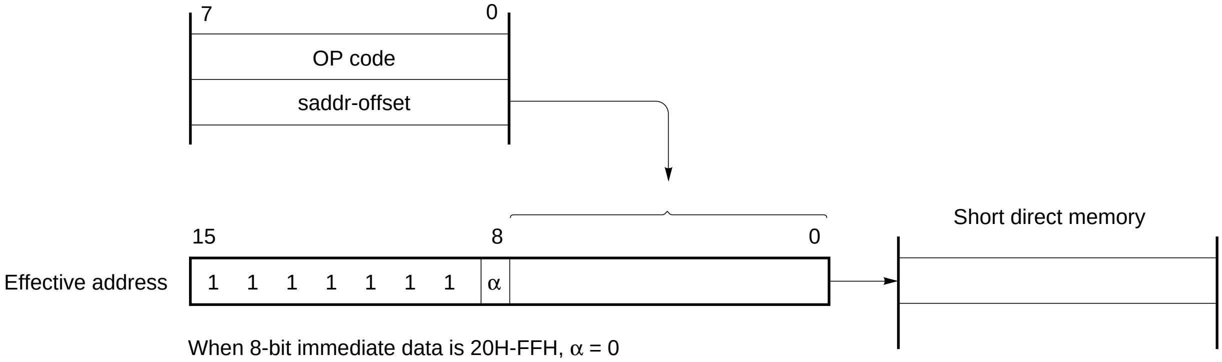
Representation	Description
saddr	Label or immediate data FE20H-FF1FH
saddrp	Label or immediate data FE20H-FF1FH (even address only)

[Example]

MOV 0FE30H, #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-FFH, α = 0

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

5.4.6 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-FFCFH and FFE0H-FFFFH. However, the SFRs mapped to an area of FF00H-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)

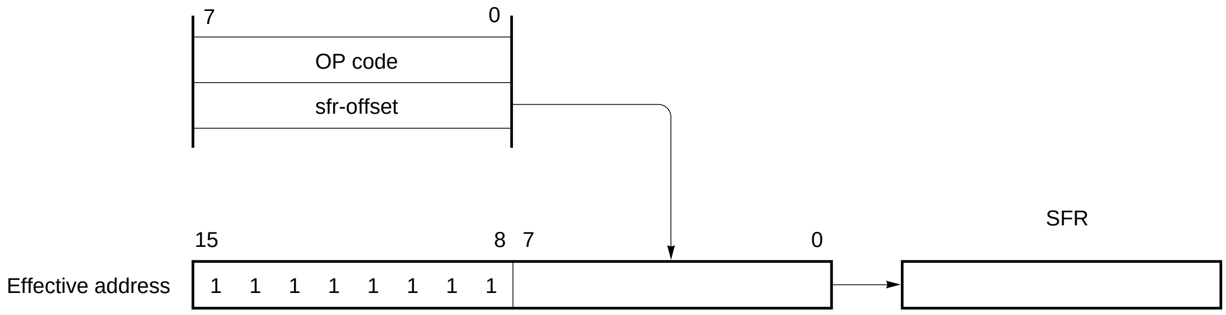
[Example]

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

Instruction code 1 1 1 1 0 1 1 0 OP code

0 0 1 0 0 0 0 0 20H (sfr-offset)

[Operation]



5.4.7 Register indirect addressing

[Function]

This addressing mode is used to address the memory by using the contents of a register pair specified as an operand. The register pair to be accessed is specified by the register bank select flags (RBS0 and RBS1) and the register pair specification code in the instruction code. This addressing mode can be used across the entire memory space.

[Operand Format]

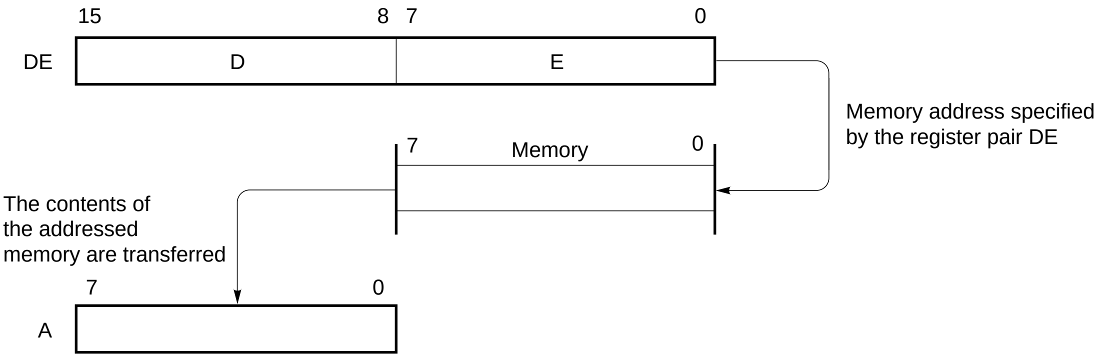
Representation	Description
—	[DE], [HL]

[Example]

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

Instruction code	1 0 0 0 0 1 0 1

[Operation]


15
8
7
0
DE
D
E
Memory address specified by the register pair DE
7
Memory
0
The contents of the addressed memory are transferred
7
0
A

5.4.8 Based addressing

[Function]

This addressing mode is used to address the memory 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 to be accessed is in the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is executed by extending the offset data to a 16-bit positive number. A carry from the 16th bit is ignored if any. This addressing mode can be used across the entire memory space.

[Operand Format]

Representation	Description
—	[HL + byte]

[Example]

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

Instruction code 1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

5.4.9 Based indexed addressing

[Function]

This addressing mode is used to address the memory by using the result of adding the contents of the B or C register specified in the instruction word to the HL register pair as a base register. The HL, B, and C registers to be accessed are in the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is executed by extending the contents of the B or C register to a 16-bit positive number. A carry from the 16th bit is ignored if any. This addressing mode can be used across 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 |

5.4.10 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 |

Phase-out/Discontinued

[MEMO]

CHAPTER 6 PORT FUNCTIONS

6.1 Functions of Ports

The μPD78018F, 78018FY subseries is provided with two input port pins and 51 I/O port pins. Figure 6-1 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 6-1. Types of Ports

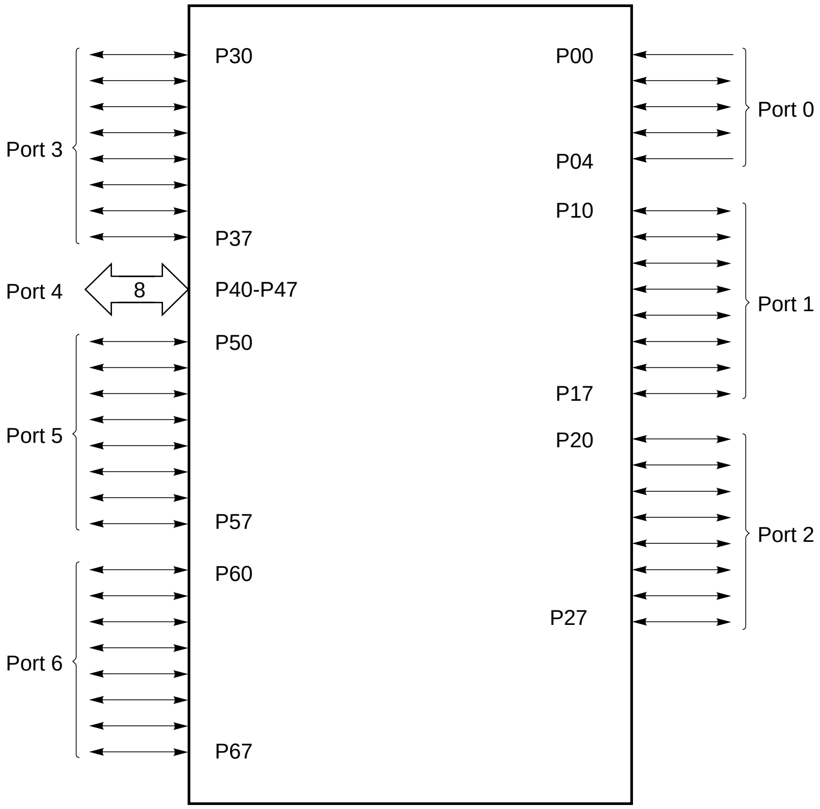


Table 6-1. Port Functions (μPD78018F Subseries)

Pin Name		Function		Shared by:
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-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-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			$\overline{SCK1}$
	P23			STB
	P24			BUSY
	P25			SI0/SB0
	P26			SO0/SB1
	P27			$\overline{SCK0}$
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-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-AD7
Port 5	P50-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-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 with mask ROM model only. Can directly drive LED.	–
	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

Table 6-2. Port Functions (μPD78018FY Subseries)

Pin Name		Function		Shared by:
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-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-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/SDA0
	P26			SO0/SB1/SDA1
	P27			$\overline{\text{SCK0}}$/SCL
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-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-AD7
Port 5	P50-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-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 with mask ROM model only. 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

6.2 Port Configuration

A port consists of the following hardware:

Table 6-3. 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 specifies the input/output mode of port 4.

6.2.1 Port 0

This is a 5-bit I/O port with output latch. P01-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-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 6-2 through 6-4 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 6-2. Block Diagram of P00

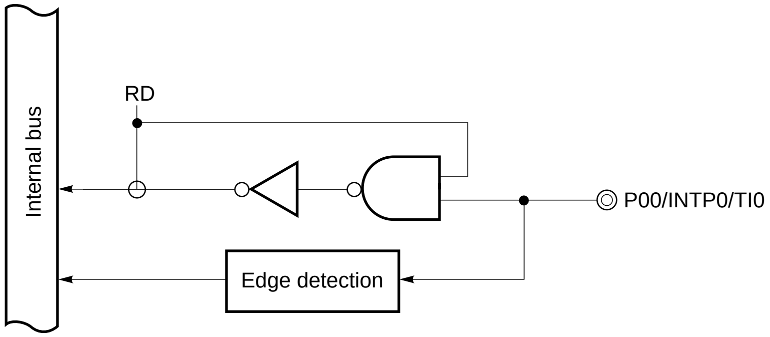
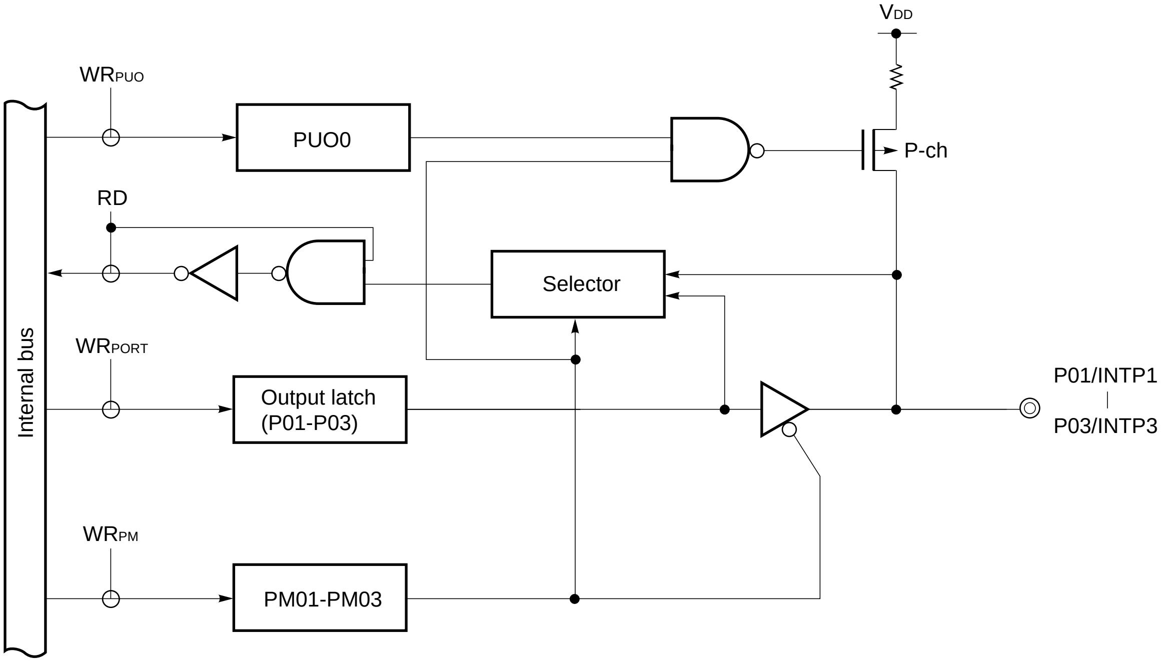
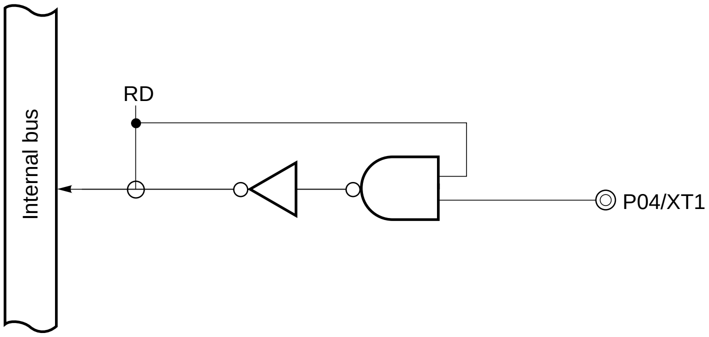


Figure 6-3. Block Diagram of P01-P03



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

Figure 6-4. Block Diagram of P04



6.2.2 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-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).

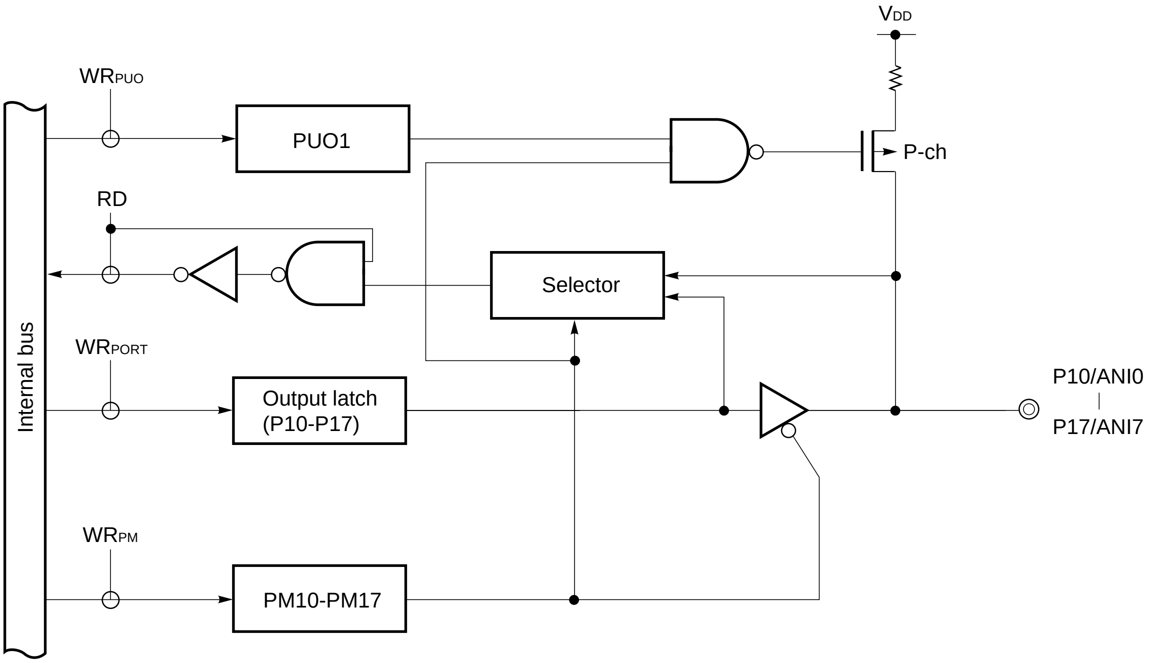
The pins of this port are also used as the analog input pins of the A/D converter.

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

Figure 6-5 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 6-5. Block Diagram of P10-P17



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

6.2.3 Port 2 (μPD78018F subseries)

This is an 8-bit I/O port with output latch. P20-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-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.

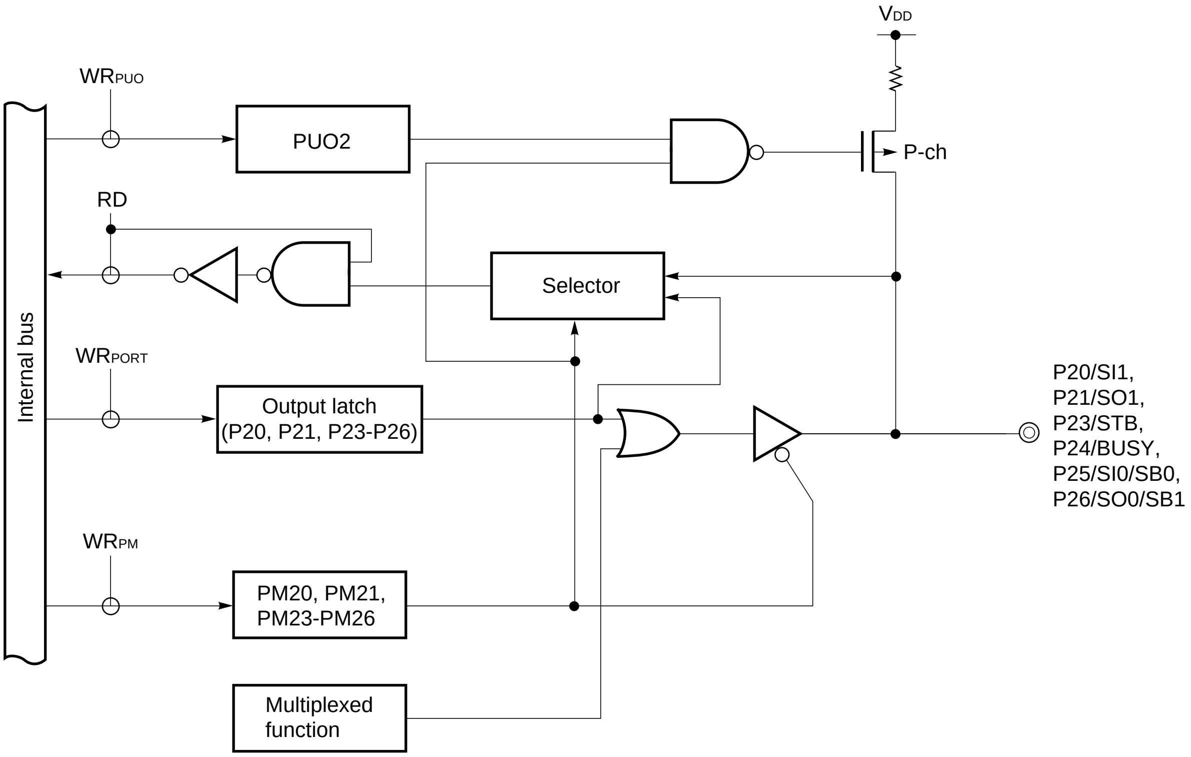
This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 6-6 and 6-7 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 15-3 Format of Serial Operation Mode Register 0 and Figure 17-3 Format of Serial Operation Mode Register 1.

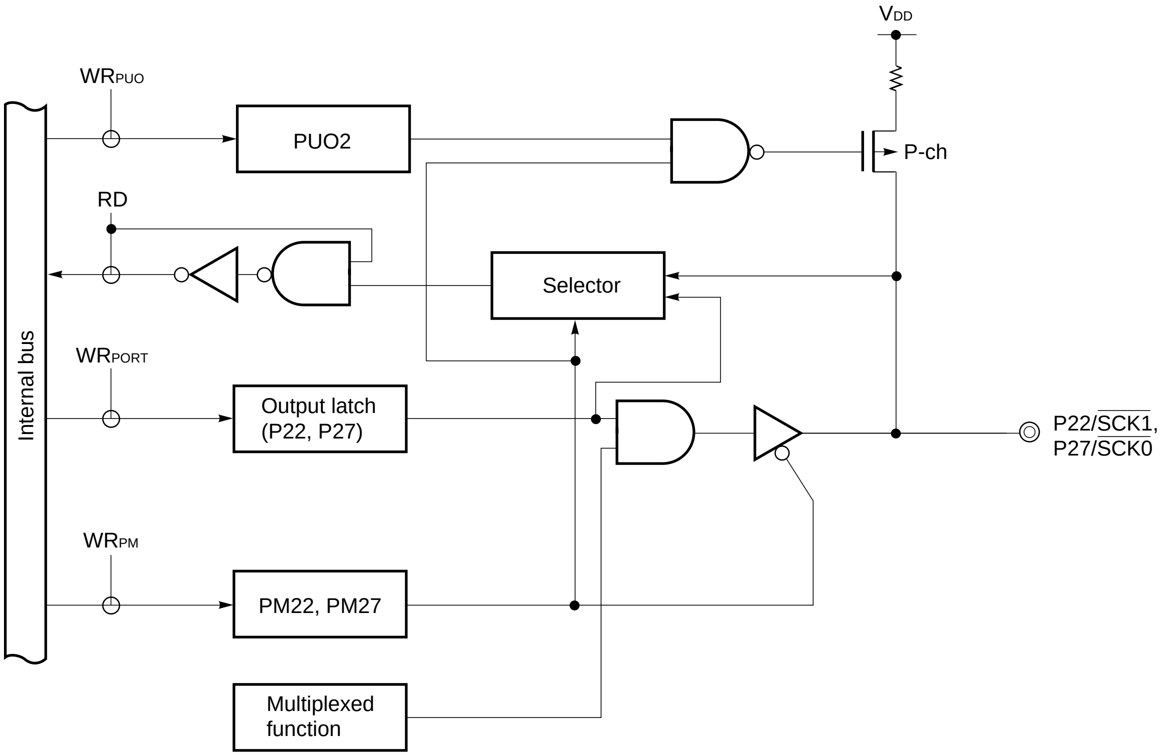
2. To read the status of the pin in the SBI mode, set PM2n bit of PM2 to 1 (n = 5, 6) (refer to 15.4.3 (10) Method to judge busy state of a slave).

Figure 6-6. Block Diagram of P20, P21, and P23-26 (μPD78018F Subseries)



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

Figure 6-7. Block Diagram of P22 and P27 (μPD78018F Subseries)



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

6.2.4 Port 2 (μPD78018FY subseries)

This is an 8-bit I/O port with output latch. P20-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-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.

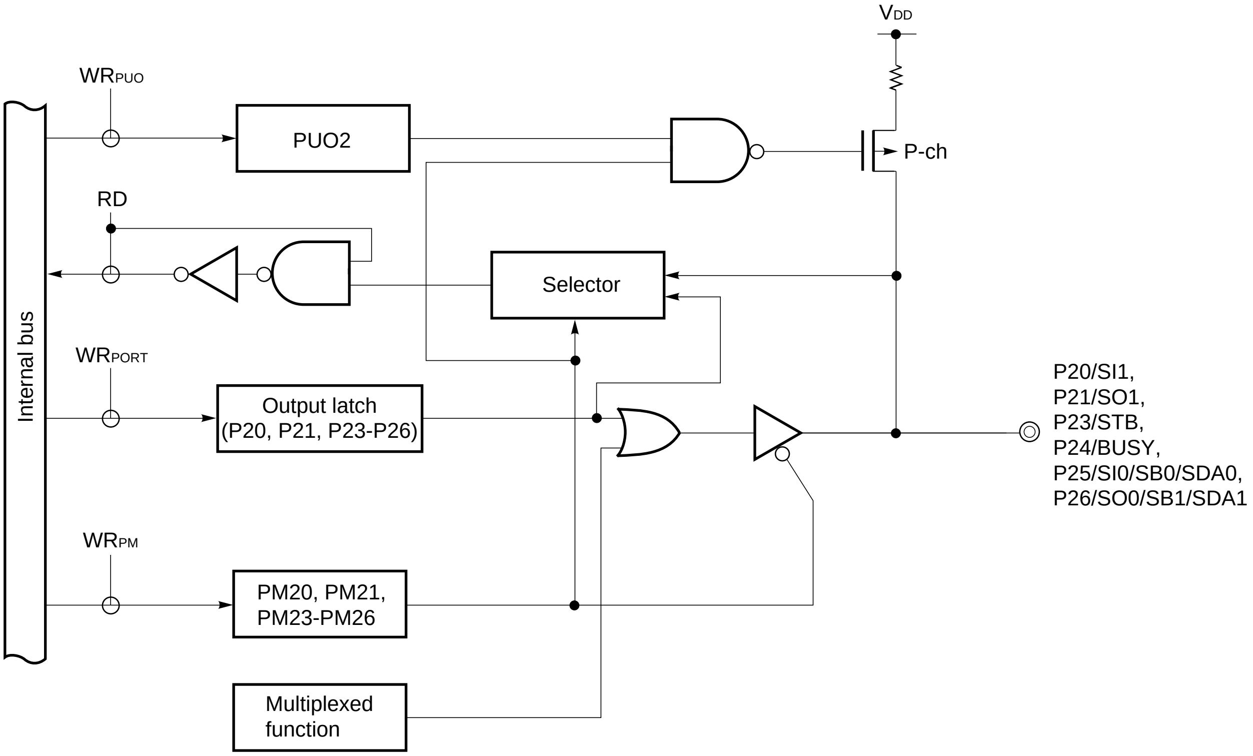
This port is set in the input mode when the $\overline{\text{RESET}}$ signal is input.

Figures 6-8 and 6-9 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 16-3 Format of Serial Operation Mode Register 0 and Figure 17-3 Format of Serial Operation Mode Register 1.

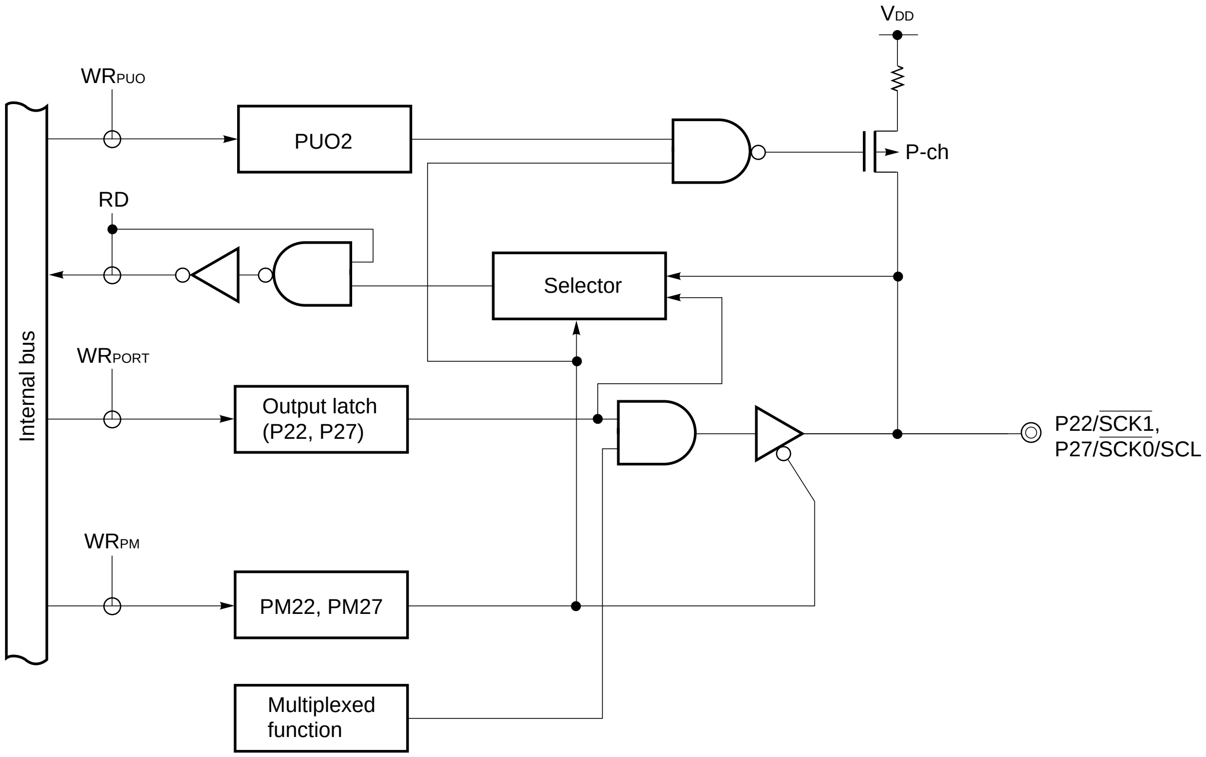
2. To read the status of the pin in the SBI mode, set PM2n to 1 (n = 5, 6).

Figure 6-8. Block Diagram of P20, P21, and P23-26 (μPD78018FY Subseries)



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

Figure 6-9. Block Diagram of P22 and P27 (μPD78018FY Subseries)



PUO : pull-up resistor option register

PM : port mode register

RD : read signal of port 2

WR : write signal of port 2

6.2.5 Port 3

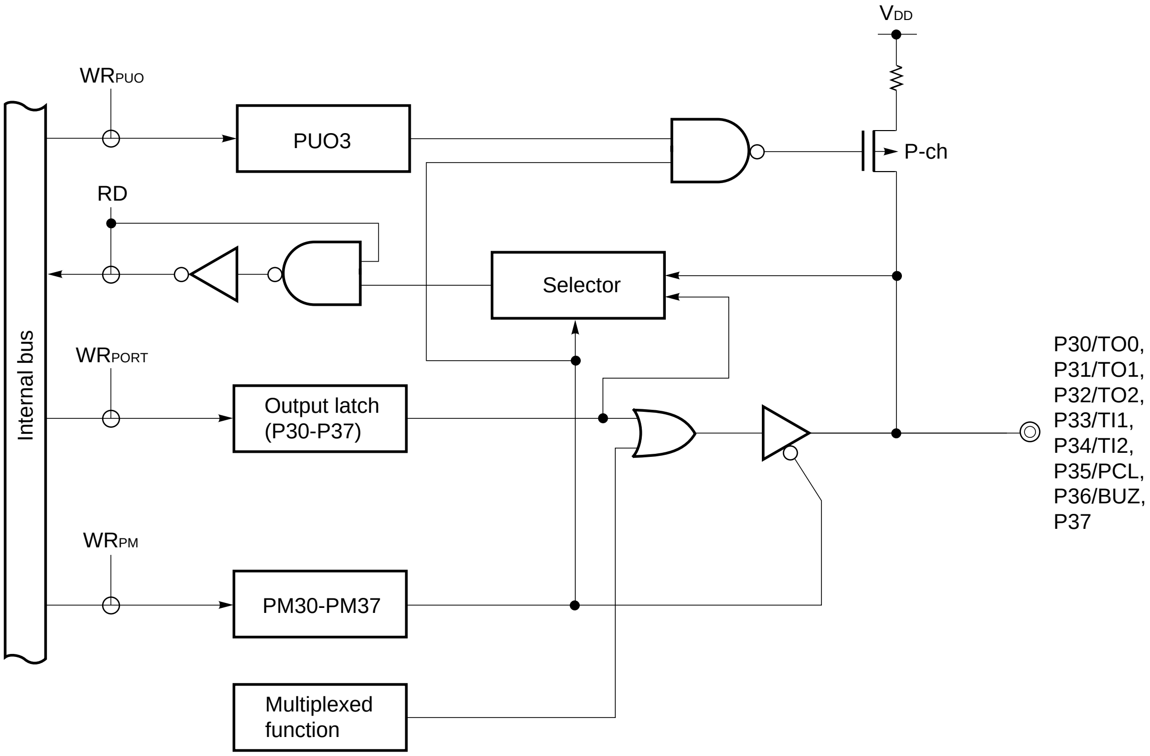
This is an 8-bit I/O port with output latch. P30-P37 pins can be specified in the input or output mode in 1-bit units by using the port mode register 3 (PM3). When using P30-P37 pins as input port pins, internal pull-up resistors can be connected in 8-bit units by using the pull-up resistor option register (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 6-10 shows the block diagram of port 3.

Figure 6-10. Block Diagram of P30-P37



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

6.2.6 Port 4

This is an 8-bit I/O port with output latch. P40-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-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).

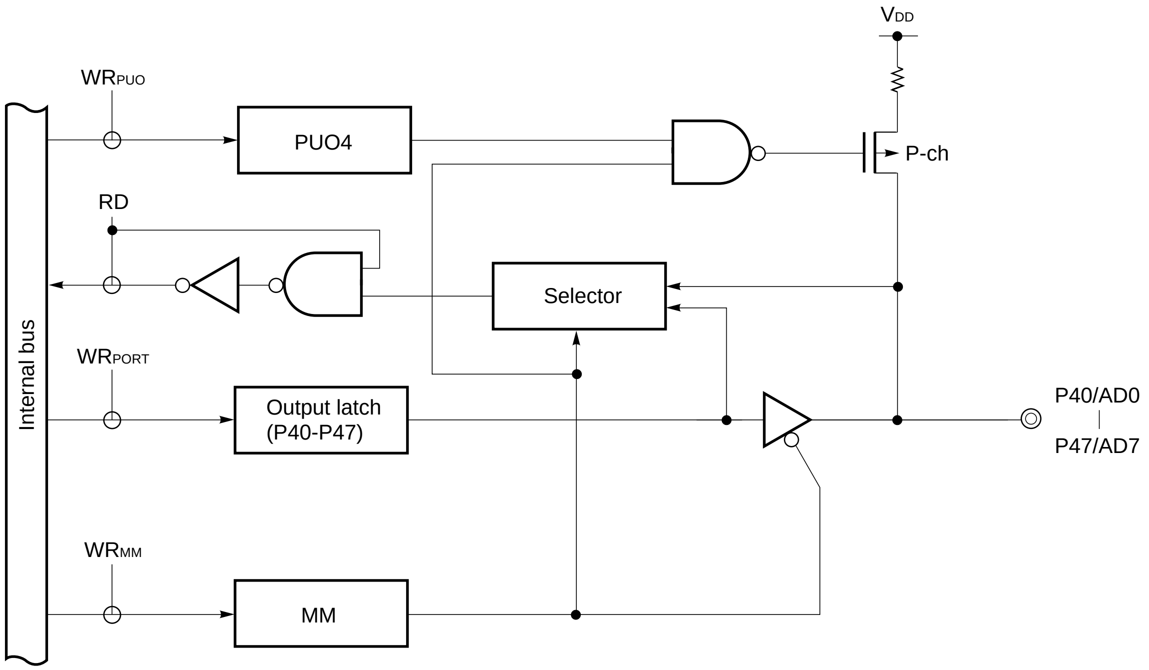
When the falling edge of any of the pins of this port is detected, a test input flag (KRIF) can be set to 1.

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.

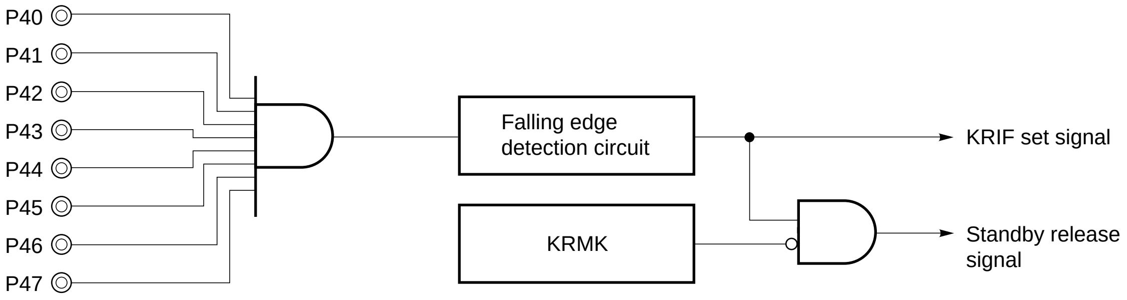
Figure 6-11 shows the block diagram of port 4, and Figure 6-12 shows the block diagram of the falling edge detection circuit.

Figure 6-11. Block Diagram of P40-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 6-12. Block Diagram of Falling Edge Detection Circuit



6.2.7 Port 5

This is an 8-bit I/O port with output latch. P50-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-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).

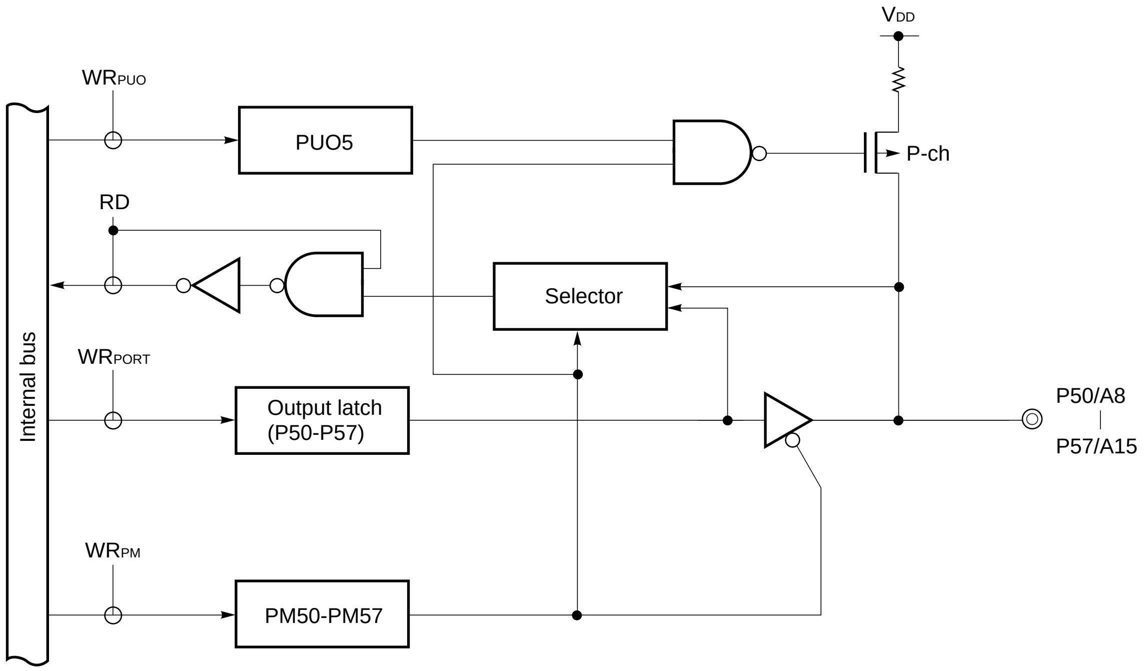
Port 5 can directly drive an LED.

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 6-13 shows the block diagram of port 5.

Figure 6-13. Block Diagram of P50-P57



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

6.2.8 Port 6

This is an 8-bit I/O port with output latch. P60-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-order or low-order 4 bits of the port are involved and also depending on whether the product is ROM or PROM model.

Table 6-4. Pull-Up Resistors in Port 6

	High-order 4 Bits (P64-P67 pins)	Low-order 4 Bits (P60-P63 pins)
Mask ROM model	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
PROM model		Internal pull-up resistor is not connected

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

P60-P63 pins can directly drive an LED.

P64-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 6-14 and 6-15 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:

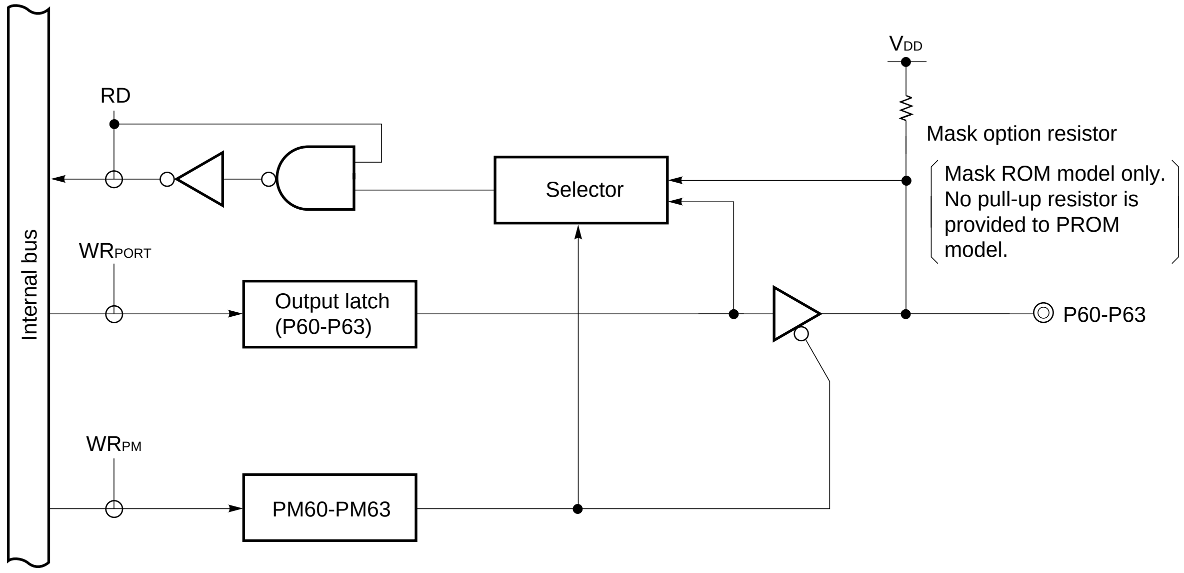
[Mask ROM model]

- 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.)

[PROM model]

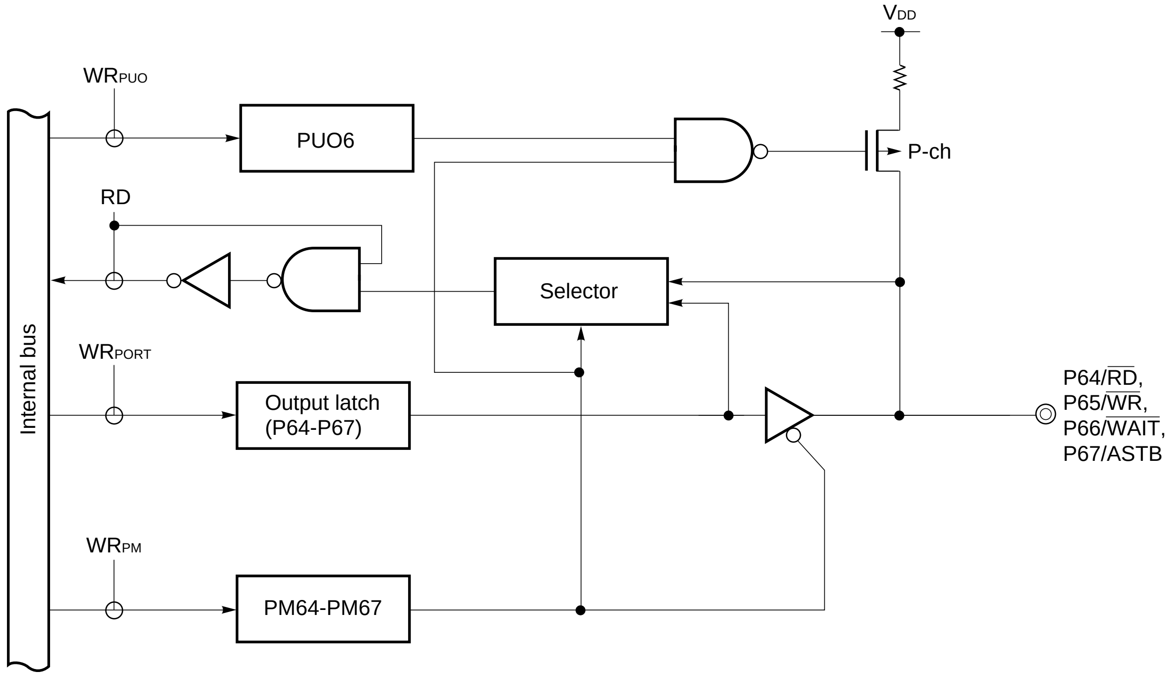
- For duration of 3 clocks (without wait) when a read instruction is executed to port 6 (P6) or port mode register 6 (PM6) : –200 μA (MAX.)
- Others : –3 μA (MAX.)

Figure 6-14. Block Diagram of P60-P63



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

Figure 6-15. Block Diagram of P64-P67



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

6.3 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 6-5.

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

2. P40-P47 pins are specified in the input or output mode by the memory extension mode register.

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 6-5. 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-P47	AD0-AD7	I/O	× Note 2	
P01-P03	INTP1-INTP3	Input	1	×	P50-P57	A8-A15	Output	× Note 2	
P04 Note 1	XT1	Input	1 (fixed)	None	P64	$\overline{\text{RD}}$	Output	× Note 2	
P10-P17 Note 1	ANI0-ANI7	Input	1	×	P65	$\overline{\text{WR}}$	Output	× Note 2	
P30-P32	TO0-TO2	Output	0	0	P66	$\overline{\text{WAIT}}$	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-P47, P50-P57, and P64-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 15-3 or 16-3 Format of Serial Operation Mode Register 0 and Figure 17-3 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 6-16. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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=0-7)
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-P67 pins, the internal pull-up resistor cannot be used even when PUOm is set to 1 (m = 1, 4-6).

3. P60-P63 pins of only the mask ROM model can be connected to the internal pull-up resistor with mask option.

Figure 6-17. Format of Pull-Up Resistor Option Register

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	On 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 6-18. Format of Memory Extension Mode Register

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

<table>
<tr><td rowspan="2">MM2</td><td rowspan="2">MM1</td><td rowspan="2">MM0</td><td rowspan="2" colspan="2">Selects single-chip/ memory extension mode</td><td colspan="6">P40-P47, P50-P57, P64-P67 pin status</td></tr>
<tr><td colspan="2">P40-P47</td><td>P50-P53</td><td>P54, P55</td><td>P56, P57</td><td>P64-P67</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td rowspan="2" colspan="2">Single-chip mode</td><td rowspan="2">Port mode</td><td>Input</td><td rowspan="2" colspan="4">Port mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Out-put</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td rowspan="4">Memory extension mode</td><td>256 B mode</td><td rowspan="4" colspan="2">AD0-AD7</td><td colspan="3">Port mode</td><td rowspan="4">P64=$\overline{\text{RD}}$
P65=$\overline{\text{WR}}$
P66=$\overline{\text{WAIT}}$
P67=ASTB</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4 KB mode</td><td rowspan="3">A8-A11</td><td colspan="2">Port mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>16 KB mode</td><td rowspan="2">A12, A13</td><td>Port mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Full address mode[Note]</td><td>A14, A15</td></tr>
<tr><td colspan="3">Others</td><td colspan="8">Setting prohibited</td></tr>
</table>

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 KB address space, except the internal ROM, RAM, SFR, and prohibited areas, can be externally extended.

Remark P60-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 6-19. Format of Key Return Mode Register

Symbol	7	6	5	4	3	2	<1>	<0>	Address	On 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).

6.4 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.

6.4.1 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.

6.4.2 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.

6.4.3 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.

6.5 Mask Option

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

No mask option is available for the μPD78P018F, 78P018FY, and no internal pull-up resistor is provided for P60-P63 pins.

CHAPTER 7 CLOCK GENERATION CIRCUIT

7.1 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 consumption in the STOP mode can be reduced.

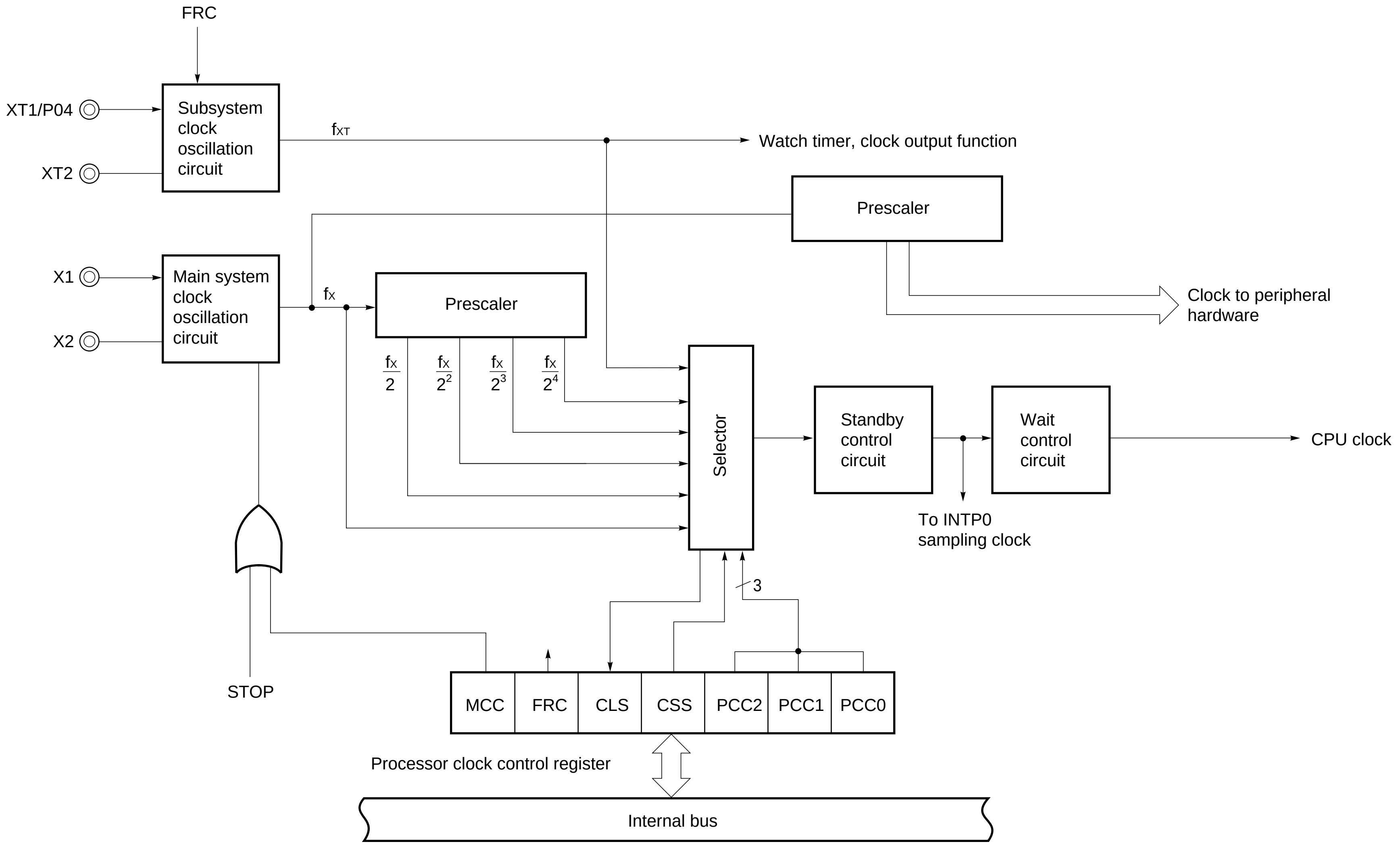
7.2 Configuration of Clock Generation Circuit

The clock generation circuit consists of the following hardware:

Table 7-1. 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 7-1. Block Diagram of Clock Generation Circuit



7.3 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 7-2. Feedback Resistor of Subsystem Clock

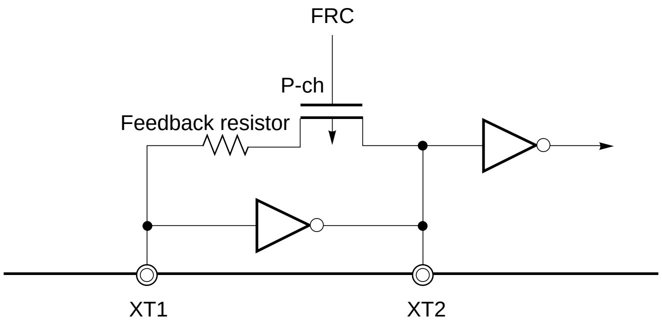


Figure 7-3. Format of Processor Clock Control Register

Symbol	<7>	<6>	<5>	<4>	3	2	1	0	Address	On 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 (0.4 µs)
	0	0	1	$f_X/2$ (0.8 µs)
	0	1	0	$f_X/2^2$(1.6 µs)
	0	1	1	$f_X/2^3$(3.2 µs)
	1	0	0	$f_X/2^4$(6.4 µs)
1	0	0	0	f_{XT} (122 µs)
	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

3. ( ) : minimum instruction execution time at f_X = 10.0 MHz or f_{XT} = 32.768 kHz: 4/f_{CPU}

The fastest instruction of the μPD78018F, 78018FY subseries is executed within four CPU clocks. Therefore, the relation between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in Table 7-2.

★ **Table 7-2. Relation 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

7.4 System Clock Oscillation Circuits

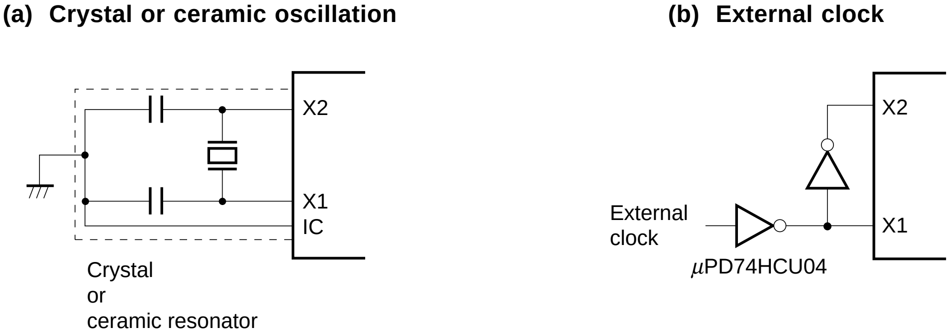
7.4.1 Main system clock oscillation circuit

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 7-4 shows the external circuit of the main system clock oscillation circuit.

Figure 7-4. External Circuit of Main System Clock Oscillation Circuit



(a) Crystal or ceramic oscillation

(b) External clock

Caution Do not execute the STOP instruction and do not set MCC (bit 7 of the processor clock control register (PCC)) to 1 when the external clock is input. Otherwise, operation of the main system clock is stopped and the X2 pin is pulled up by V_{DD}.

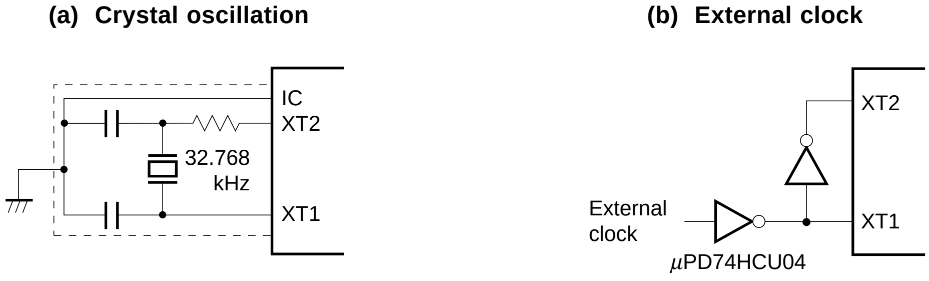
7.4.2 Subsystem clock oscillation circuit

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 7-5 shows the external circuit of the subsystem clock oscillation circuit.

Figure 7-5. External Circuit of Subsystem Clock Oscillation Circuit



(a) Crystal oscillation

(b) External clock

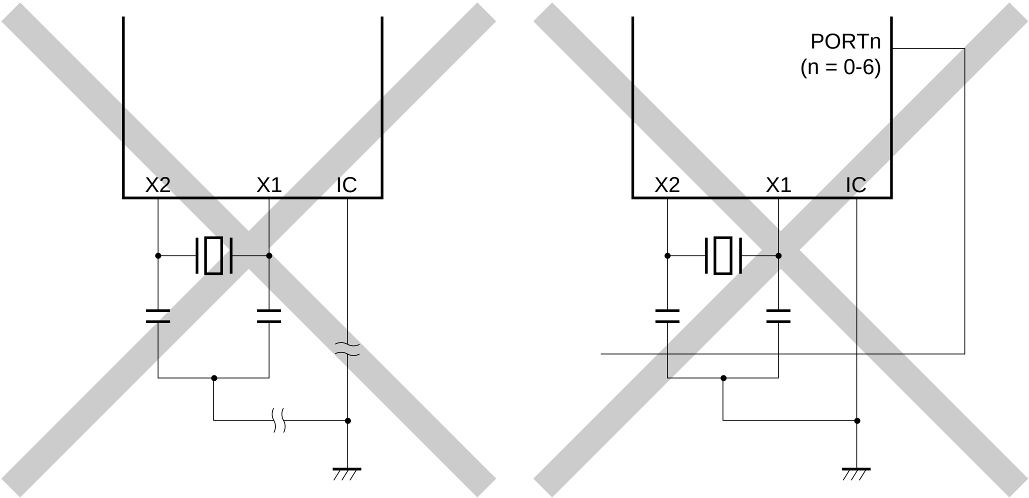
Refer to **Cautions** on the following pages.

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 dotted line in Figures 7-4 and 7-5 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 consumption.
Figure 7-6 shows incorrect examples of resonator connection.

Figure 7-6. Incorrect Examples of Resonator Connection (1/2)

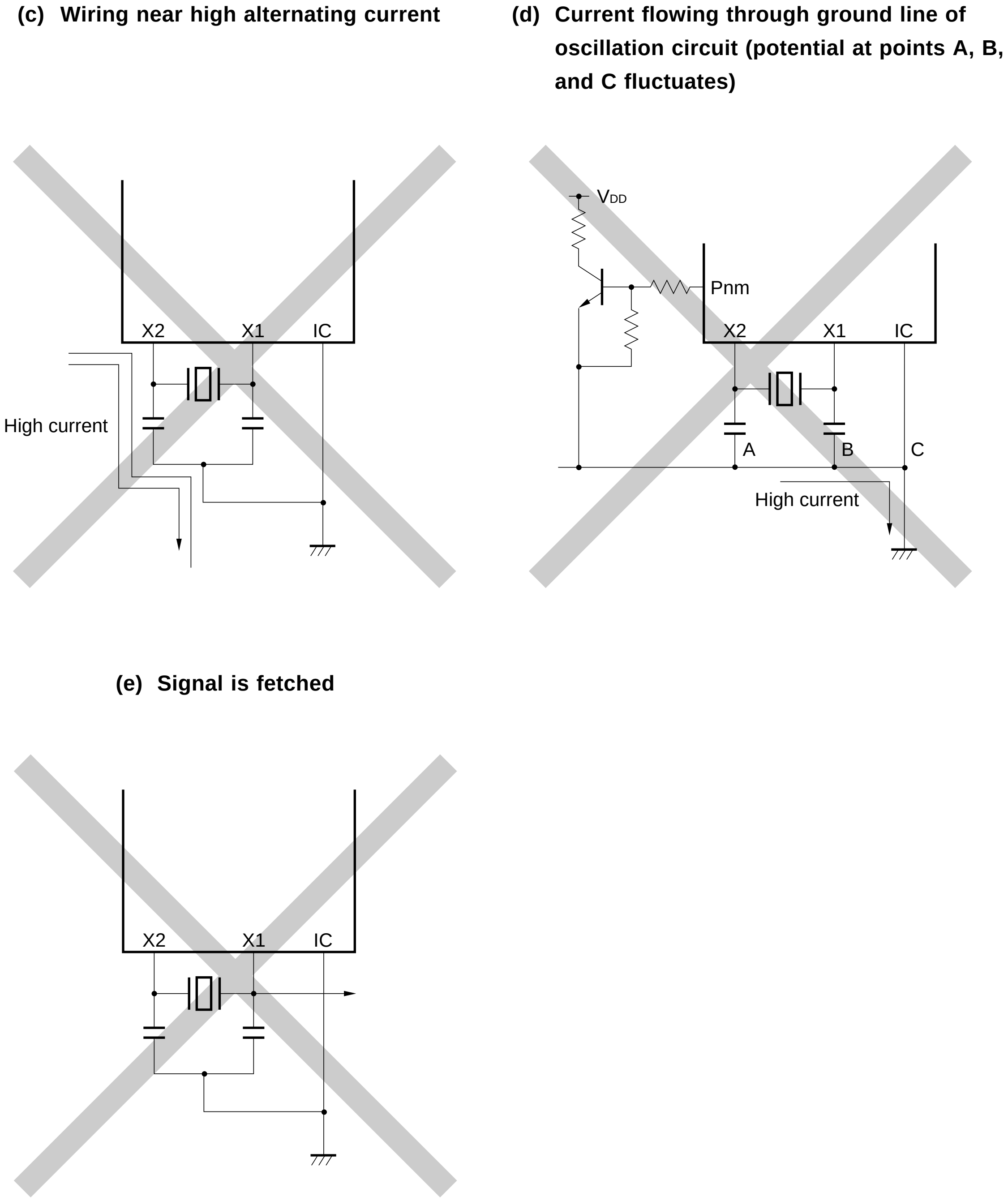


(a) Too long wiring

(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 7-6. 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 is fetched

Remark When using a subsystem clock, replace X1 and X2 with TX1 and TX2, respectively. Also, insert resistors in series on the XT2 side.

Caution 2. In Figure 7-8 (f), XT2 and X1 are wired in parallel. Thus, the cross-talk noise of X1 may increase with XT2, resulting in malfunctioning. To prevent that from occurring, it is recommended to wire XT2 and X1 so that they are not in parallel, and to correct the IC pin between XT2 and X1 directly to V_{SS}.

7.4.3 Divider circuit

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

7.4.4 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 : 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.

7.5 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 microcomputer 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).

7.5.1 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 0-2 (PCC0-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 7-7**).

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

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

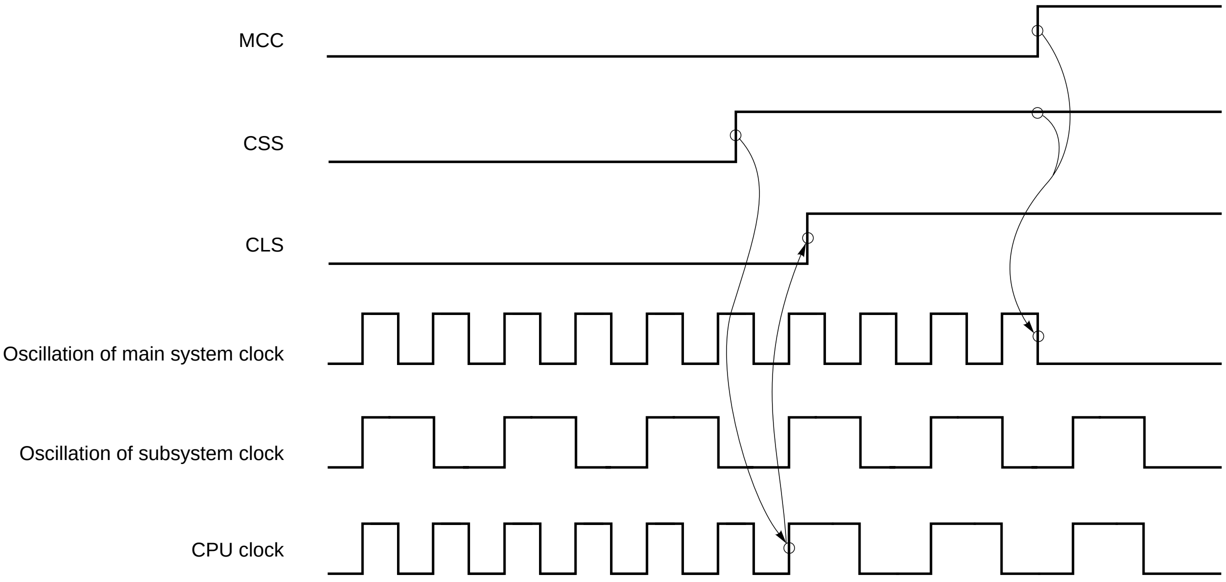
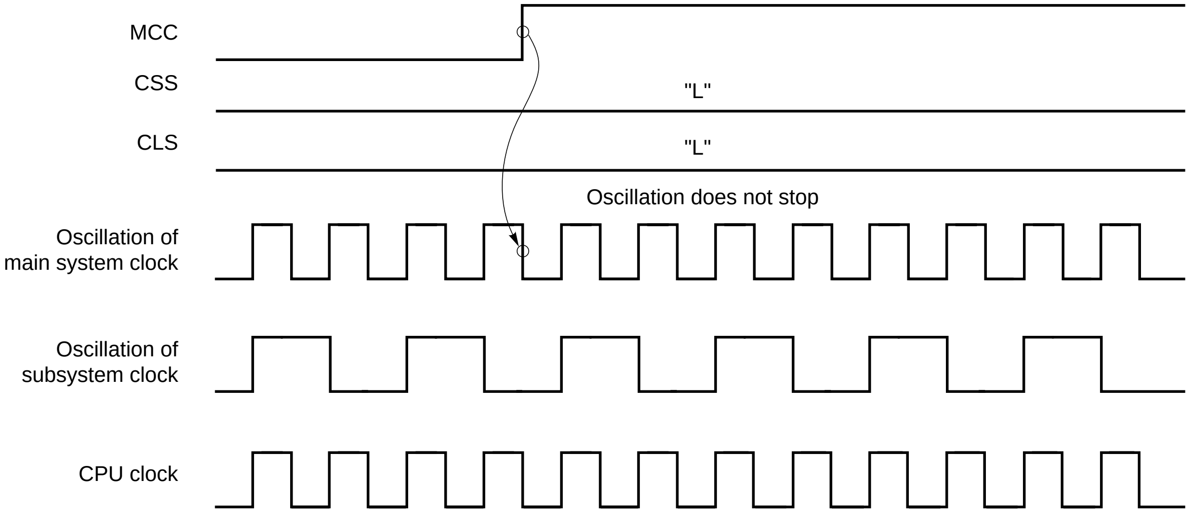
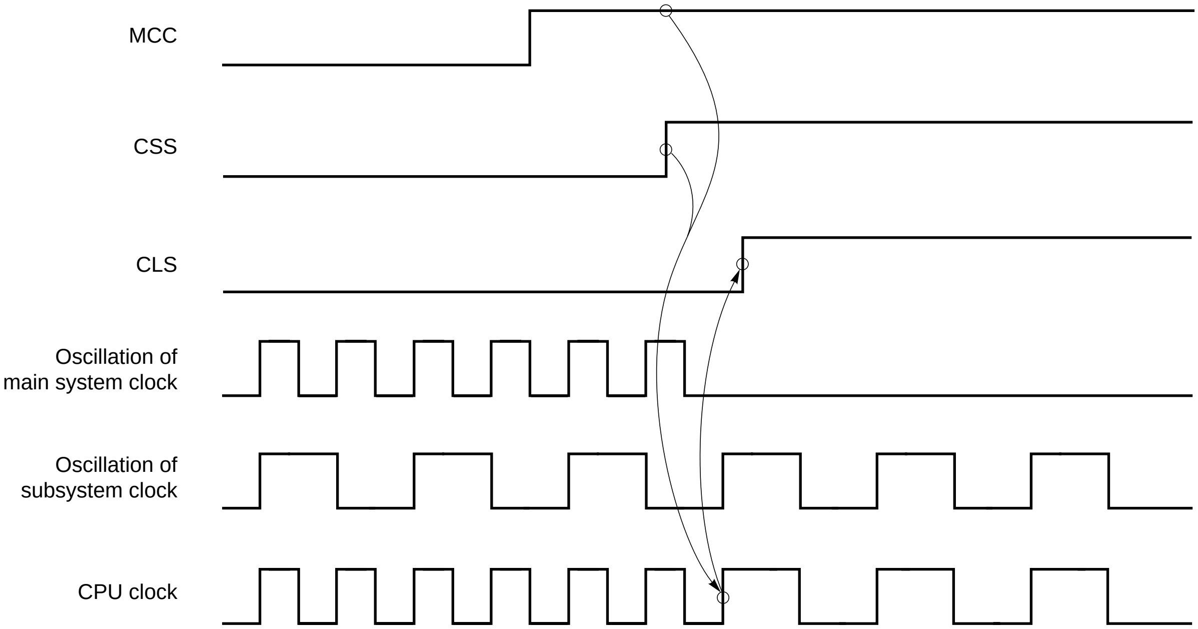

MCC
CSS
CLS
Oscillation of main system clock
Oscillation of subsystem clock
CPU clock

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

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


MCC
CSS
"L"
CLS
"L"
Oscillation does not stop
Oscillation of main system clock
Oscillation of subsystem clock
CPU clock

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


MCC
CSS
CLS
Oscillation of main system clock
Oscillation of subsystem clock
CPU clock

7.5.2 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 0-2 (PCC0-PCC2) of PCC.

(b) The watchdog timer stops counting.

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

7.6 Changing Setting of System Clock and CPU Clock

7.6.1 Time required for switching between system clock and CPU clock

The system clock or CPU clock can be selected by using bits 0 through 2 (PCC0-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 7-3**).

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

Table 7-3. 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-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-PCC2) can be selected and the subsystem clock can be switched to the main system clock at the same time (CSS 1 → 0).

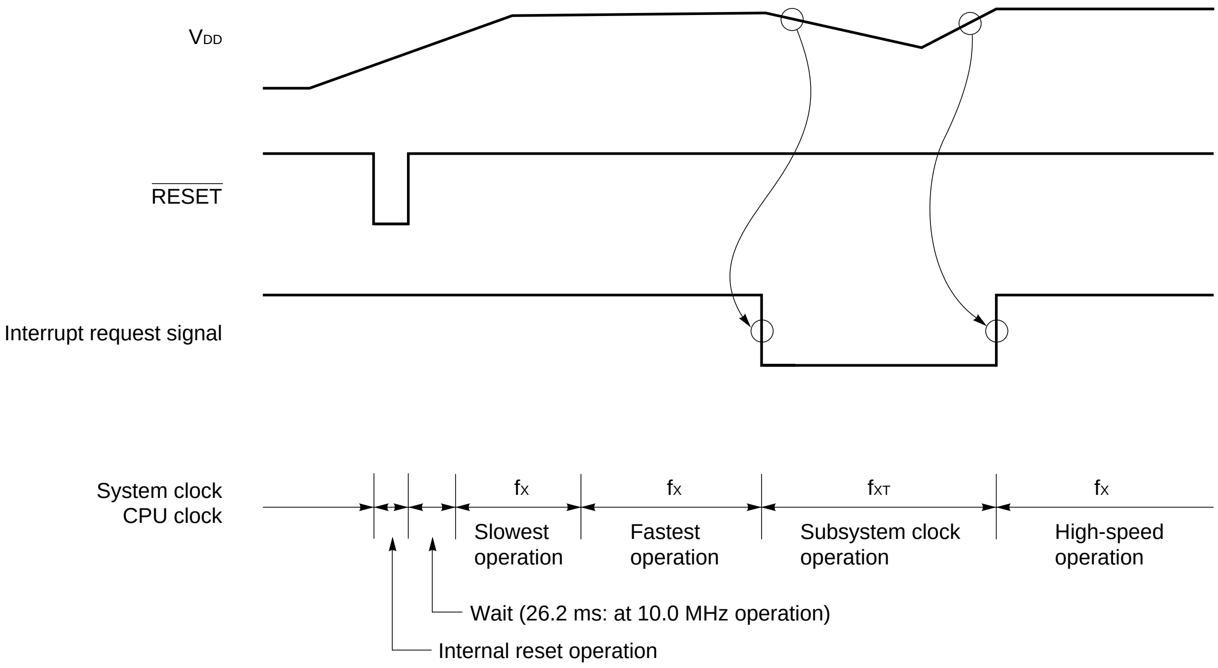
Remarks 1. One instruction is the minimum instruction execution time of the CPU clock before switching.

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

7.6.2 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 7-8. Switching between System Clock and CPU Clock



<1> The CPU is reset when the $\overline{RESET}$ pin is made low on power application. The effect of resetting is released when the $\overline{RESET}$ 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, 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]

★ 8.1 Outline of Timers in μPD78018F, 78018FY Subseries

This chapter explains the 16-bit timer/event counter. For reference, an outline of the timers provided in the μPD78018F, 78018FY subseries is shown below.

(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 9 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 10 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 11 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 12 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 13 BUZZER OUTPUT CONTROL CIRCUIT**).

Table 8-1. 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.

8.2 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 8-2. 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 × 1/fx (400 ns)	2^{17} × 1/fx (13.1 ms)	2 × 1/fx (200 ns)
2^3 × 1/fx (800 ns)	2^{18} × 1/fx (26.2 ms)	2^2 × 1/fx (400 ns)
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. ( ): fx = 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 8-3. Square Wave Output Range 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 × 1/fx (400 ns)	2^{17} × 1/fx (13.1 ms)	2 × 1/fx (200 ns)
2^3 × 1/fx (800 ns)	2^{18} × 1/fx (26.2 ms)	2^2 × 1/fx (400 ns)
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. ( ): fx = 10.0 MHz operation

8.3 Configuration of 16-Bit Timer/Event Counter

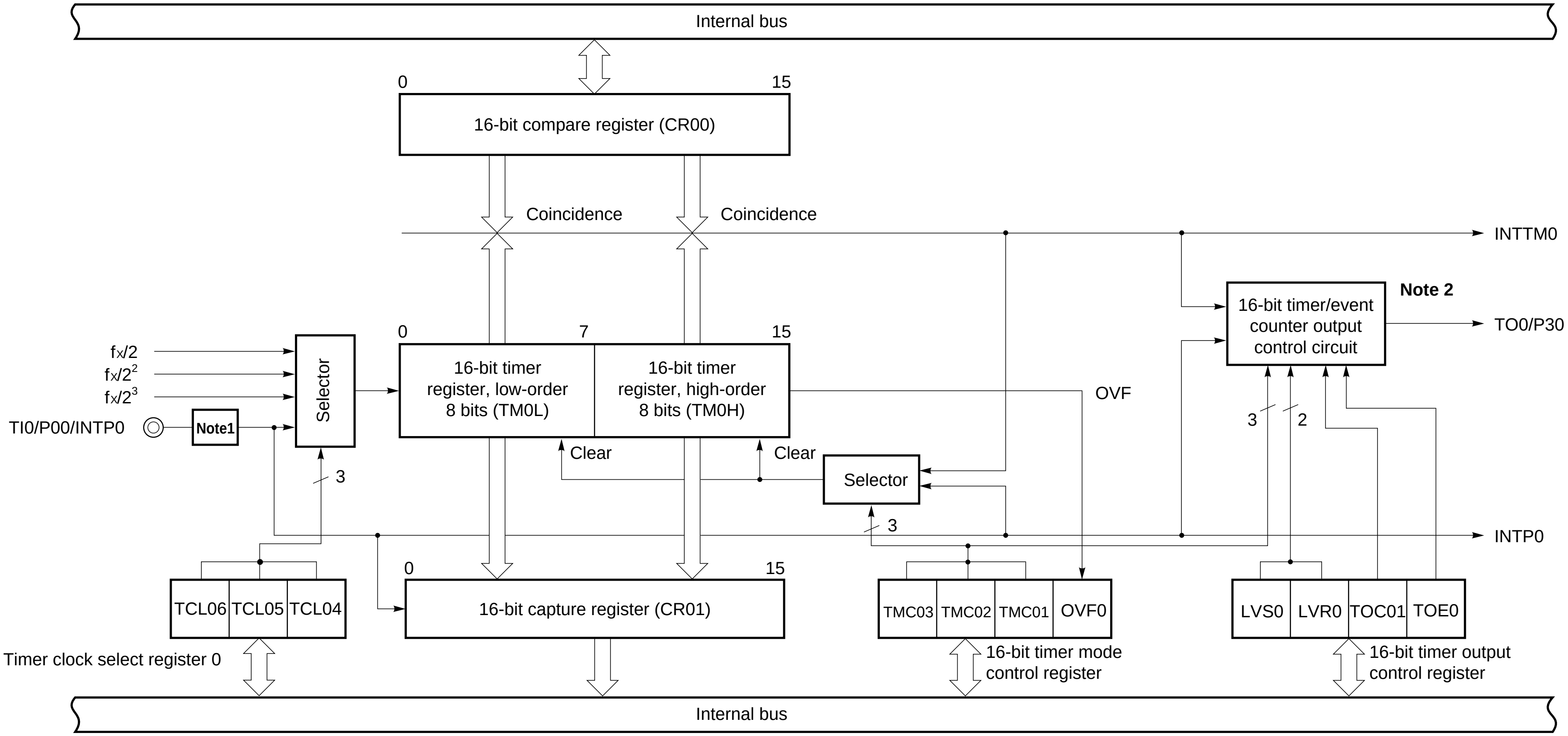
The 16-bit timer/event counter consists of the following hardware:

Table 8-4. 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 18-1 Basic Configuration of Interrupt Function**.

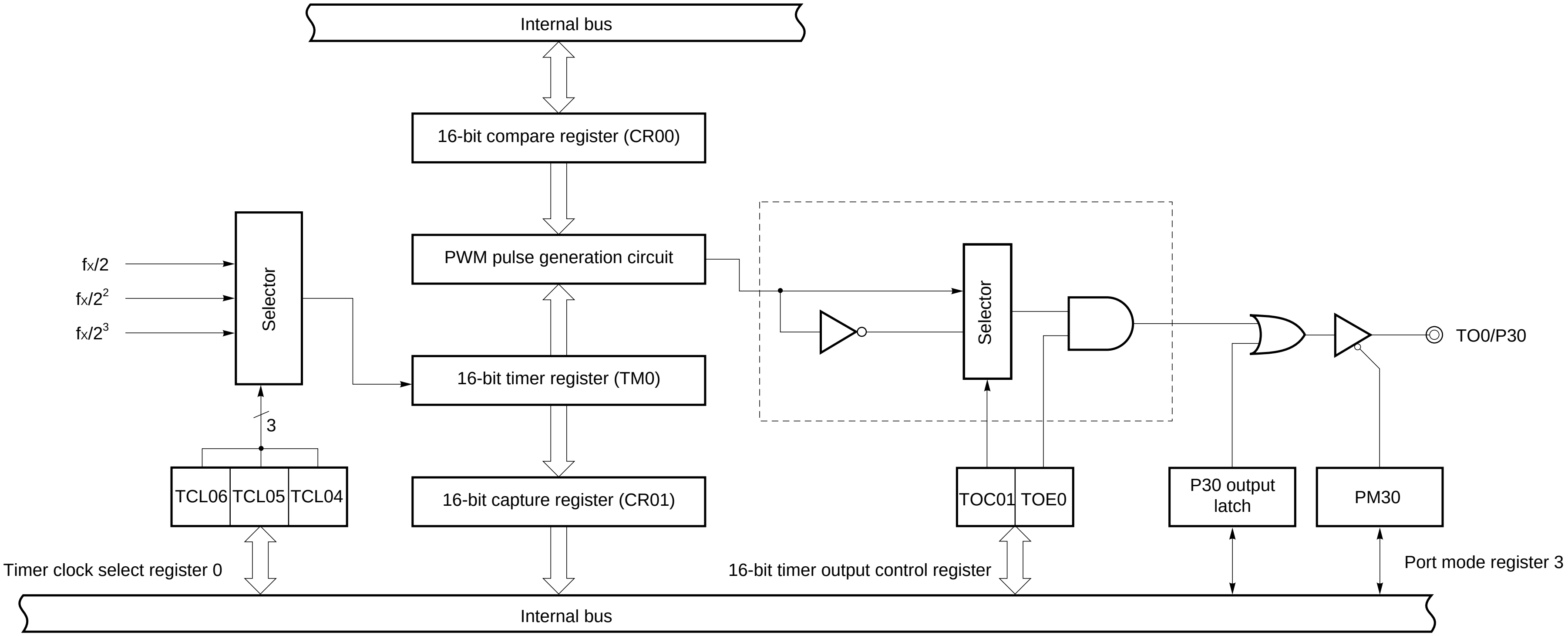
Figure 8-1. Block Diagram of 16-Bit Timer/Event Counter (Timer Mode)


Internal bus
0
15
16-bit compare register (CR00)
Coincidence
Coincidence
INTTM0
Note 2
16-bit timer/event counter output control circuit
TO0/P30
fx/2
fx/2²
fx/2³
TI0/P00/INTP0
Note1
Selector
0
7
15
16-bit timer register, low-order 8 bits (TM0L)
16-bit timer register, high-order 8 bits (TM0H)
OVF
3
2
Clear
Clear
Selector
3
3
INTP0
0
15
TCL06
TCL05
TCL04
16-bit capture register (CR01)
TMC03
TMC02
TMC01
OVF0
LVS0
LVR0
TOC01
TOE0
Timer clock select register 0
16-bit timer mode control register
16-bit timer output control register
Internal bus

Notes 1. Edge detection circuit

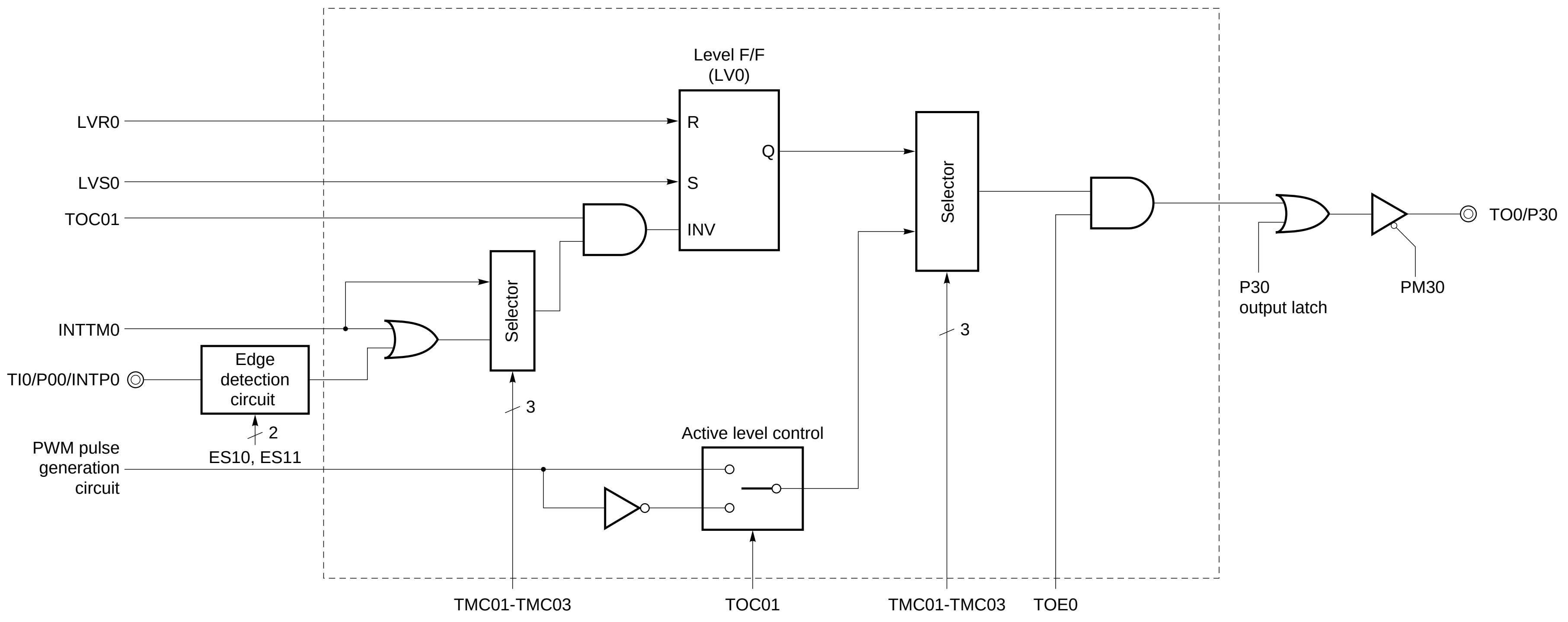
2. For the configuration of the output control circuit of the 16-bit timer/event counter, refer to **Figure 8-3**.

Figure 8-2. Block Diagram of 16-Bit Timer/Event Counter (PWM Mode)


Internal bus
16-bit compare register (CR00)
fx/2
fx/2²
fx/2³
Selector
PWM pulse generation circuit
16-bit timer register (TM0)
3
TCL06
TCL05
TCL04
16-bit capture register (CR01)
Selector
TO0/P30
TOC01
TOE0
P30 output latch
PM30
Timer clock select register 0
16-bit timer output control register
Port mode register 3
Internal bus

Remark The portion enclosed in dotted line is included in the output control circuit.

Figure 8-3. 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 can also be used to hold interval time, and when set as a PWM output operation, this can also be used to set a pulse width.

CR00 is set by a 16-bit memory manipulation instruction in a range of 0001H-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 high-order 14 bits of CR00. At this time, set the low-order 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 (M) is less than the previous value (N), 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.

8.4 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 8-4)

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 8-5)

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-TMC03 (operation stop mode). To stop the operation, set TMC01-TMC03 to 0, 0, 0.

Figure 8-4. Format of Timer Clock Select Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	On 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/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 through 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. **Before writing data other than that already written to TCL0, stop the timer operation.**

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 12 CLOCK OUTPUT CONTROL CIRCUIT**.

Figure 8-5. Format of 16-Bit Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	<0>	Address	On 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
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 8-6. Format of 16-Bit Timer Output Control Register

Symbol	7	6	5	4	<3>	<2>	1	<0>	Address	On 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 8-7. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	On 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 = 0-7)
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-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 8-8. Format of External Interrupt Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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 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 8-9. Format of Sampling Clock Select Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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 = 0-4) set to bits 0 through 2 (PCC0-PCC2) of processor clock control register (PCC)

2. f_X : Main system clock oscillation frequency

3. ( ): At f_X = 10.0 MHz operation

8.5 Operation of 16-Bit Timer/Event Counter

8.5.1 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 through 6 (TCL04-TCL06) of the timer clock select register 0 (TCL0).

For the operation when compare register value is changed during timer count operation, refer to **8.6 Notes on Using 16-bit Timer/Event Counter (3)**.

Figure 8-10. Configuration of Interval Timer

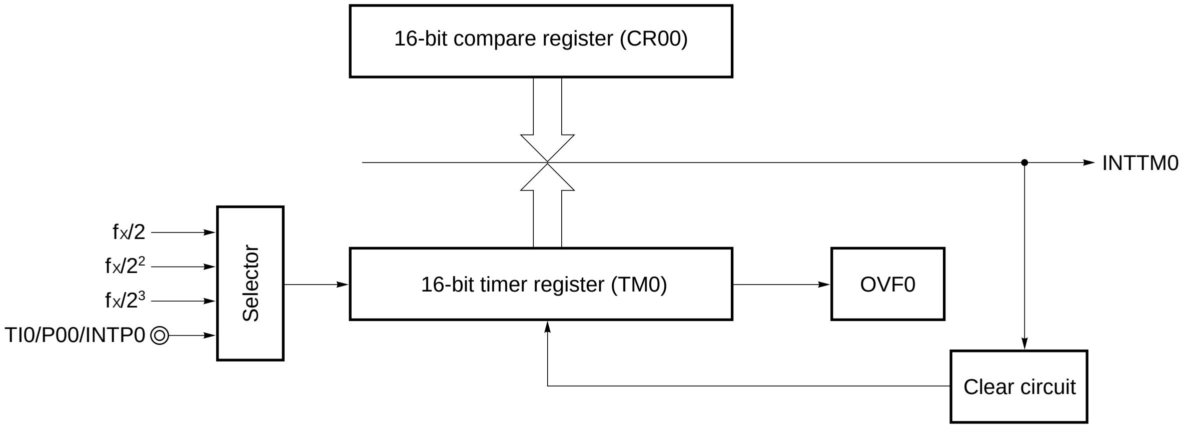
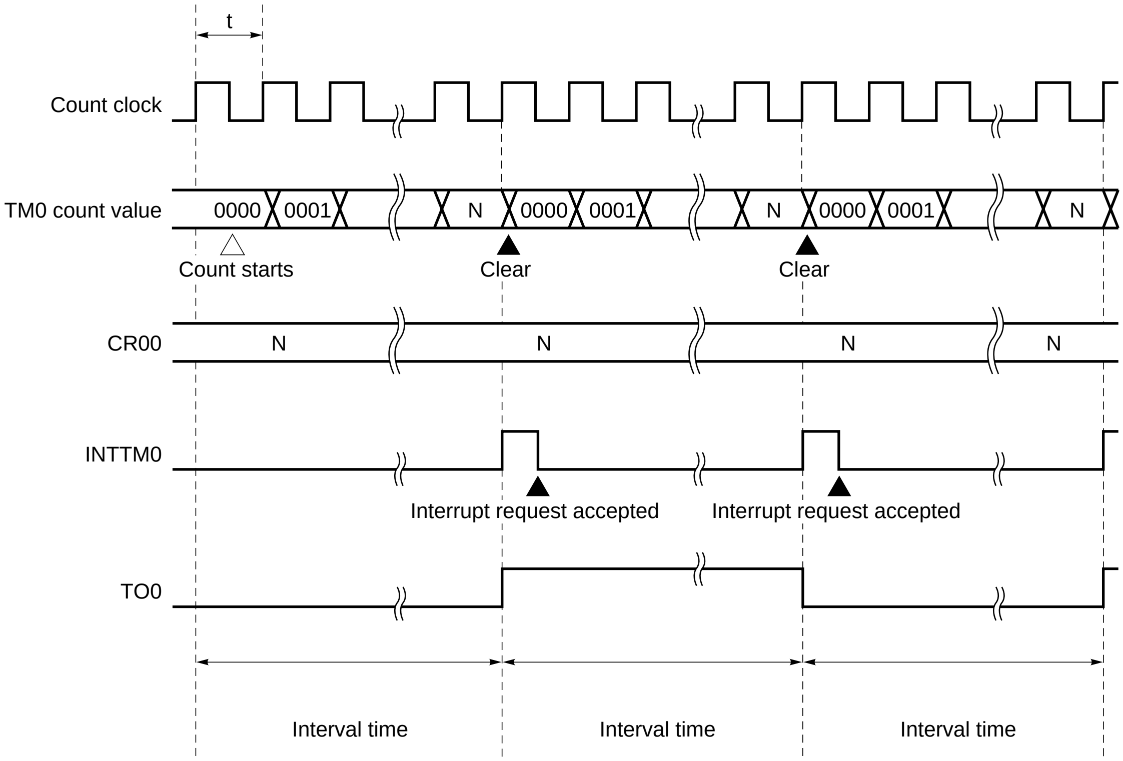


Figure 8-11. Interval Timer Operation Timing



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

Table 8-5. Interval Time of 16-Bit Timer/Event Counter

TCL06	TCL05	TCL04	Minimum Interval Time	Maximum Interval time	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)
Others			Setting prohibited		

Remarks
1. f_X : Main system clock oscillation frequency
2. TCL04-TCL06 : Bits 4 through 6 of timer clock select register 0 (TCL0)
3. ( ) : At f_X = 10.0 MHz operation

8.5.2 Operation as PWM output

The 16-bit timer/event counter performs PWM output when bits 1 through 3 (TMC01-03) 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 high-order 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 through 6 (TCL04-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 8-12, 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 8-12. Example of Configuration of D/A Converter Using PWM Output

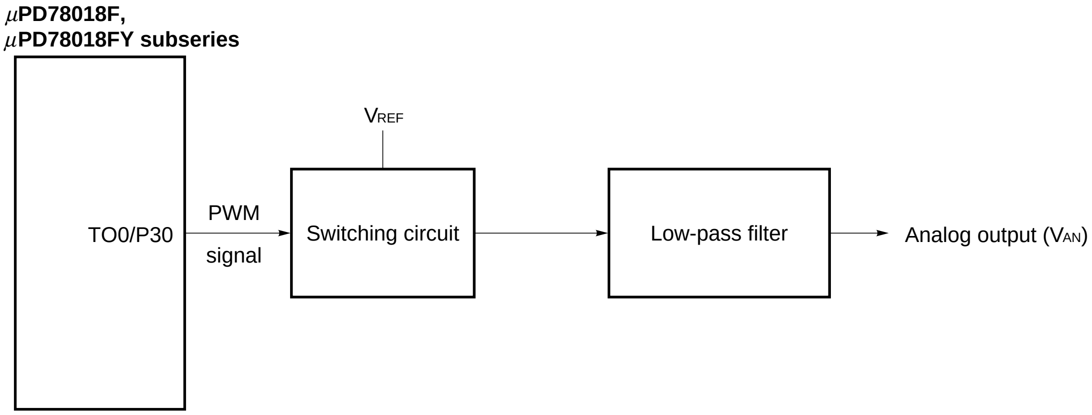
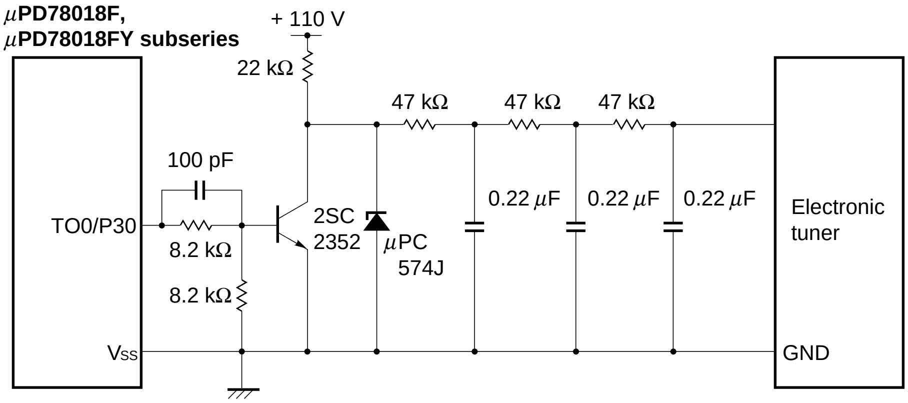


Figure 8-13 shows an example where the PWM output is converted into an analog voltage and applied for a TV tuner of voltage synthesizer type.

Figure 8-13. Example of Application Circuit (TV Tuner)



8.5.3 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 in free running, 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 8-14. Configuration of Pulse Width Measurement by Free Running

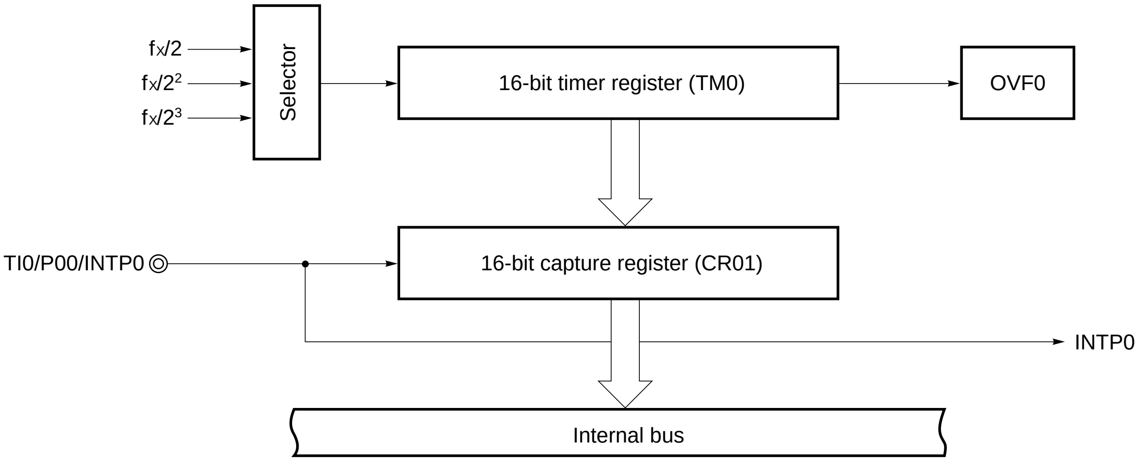
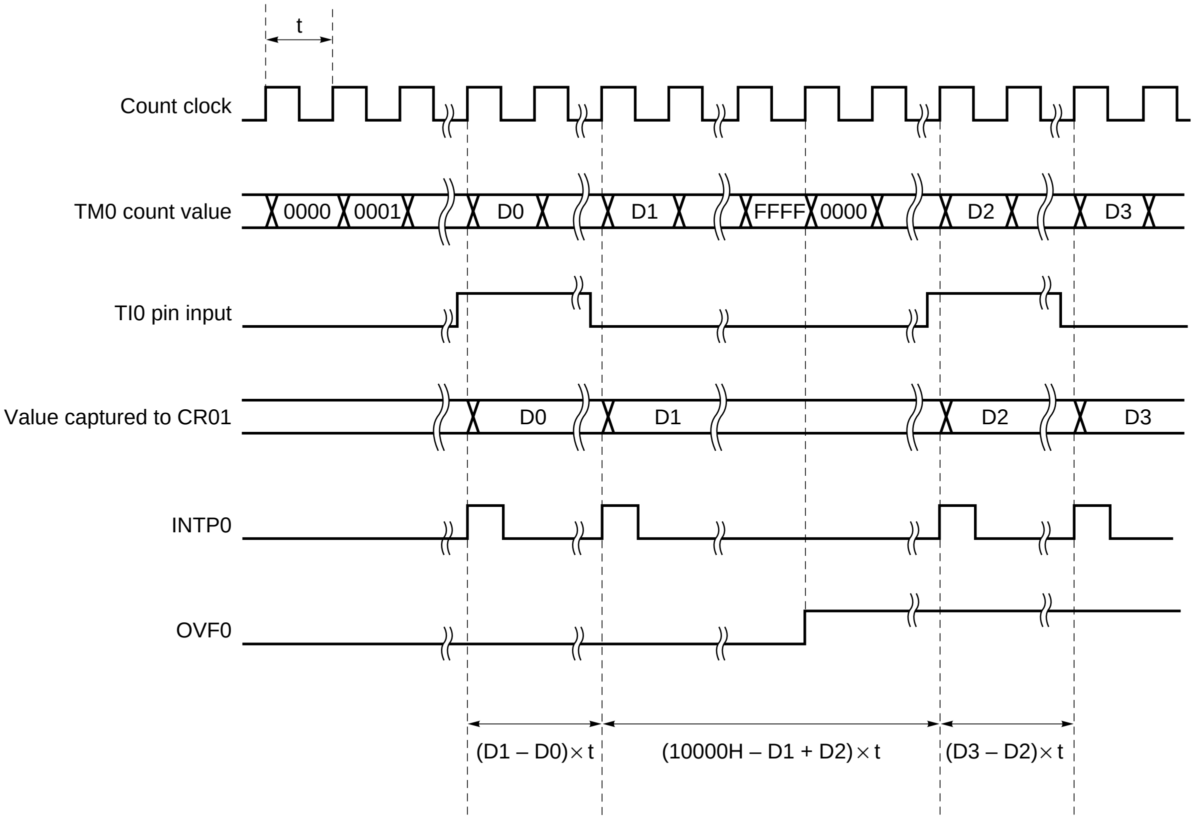


Figure 8-15. Pulse Width Measurement Timing by Free Running (with both rising and falling edges specified)



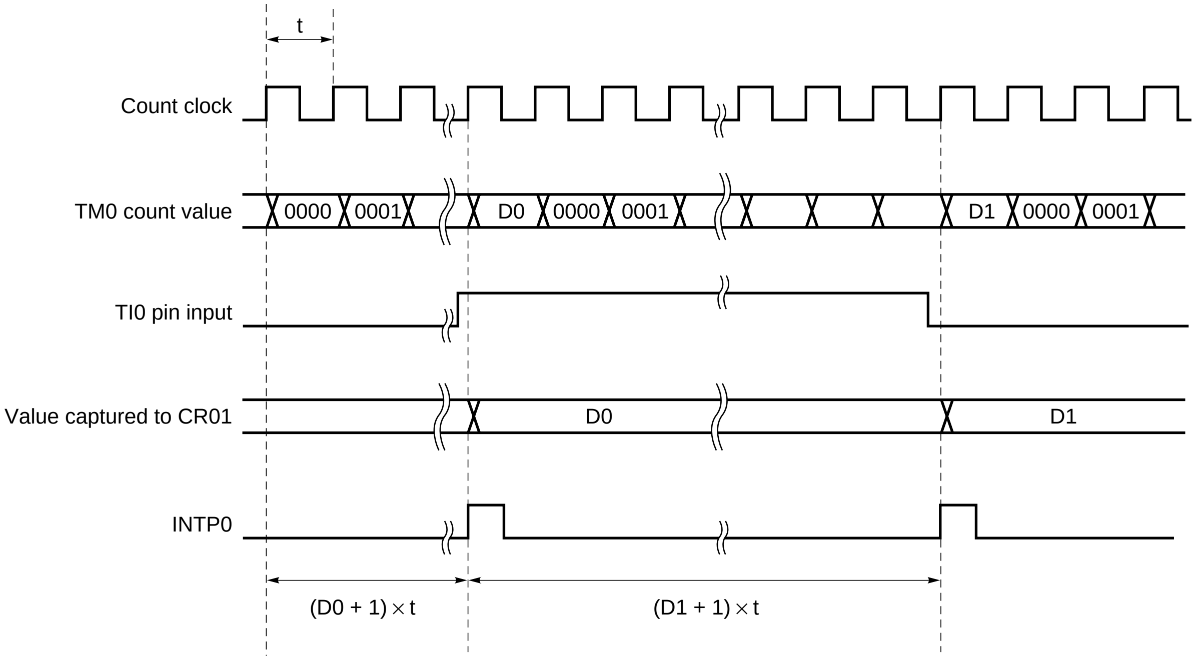
(2) Pulse width measurement by restarting timer

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 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 8-16. Pulse Width Measurement Timing by Restarting Timer (with both rising and falling edges specified)



8.5.4 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 any other value other than 0000H to CR00 (one-pulse count operation 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 8-17. Configuration of External Event Counter

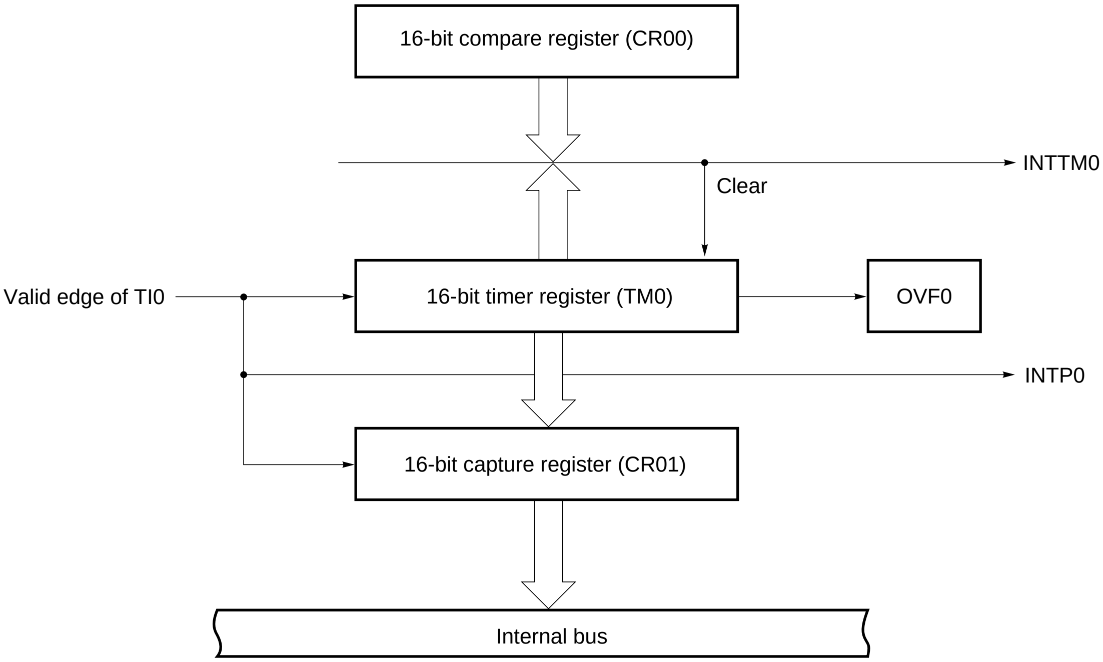
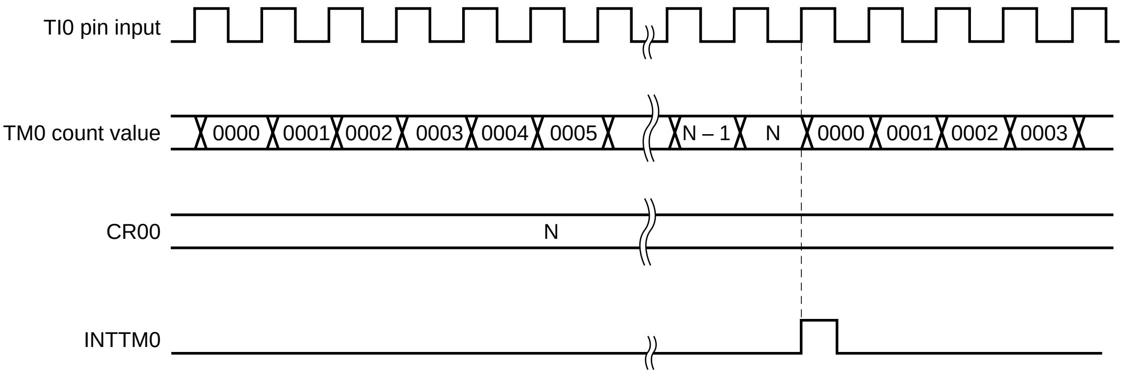

16-bit compare register (CR00)
INTTM0
Clear
Valid edge of TI0
16-bit timer register (TM0)
OVF0
INTP0
16-bit capture register (CR01)
Internal bus

Figure 8-18. External Event Counter Operation Timing (with rising edge specified)



8.5.5 Operation as square wave output

The 16-bit timer/event counters operate as square wave output 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 8-6. Square Wave Output Range of 16-Bit Timer/Event Counter

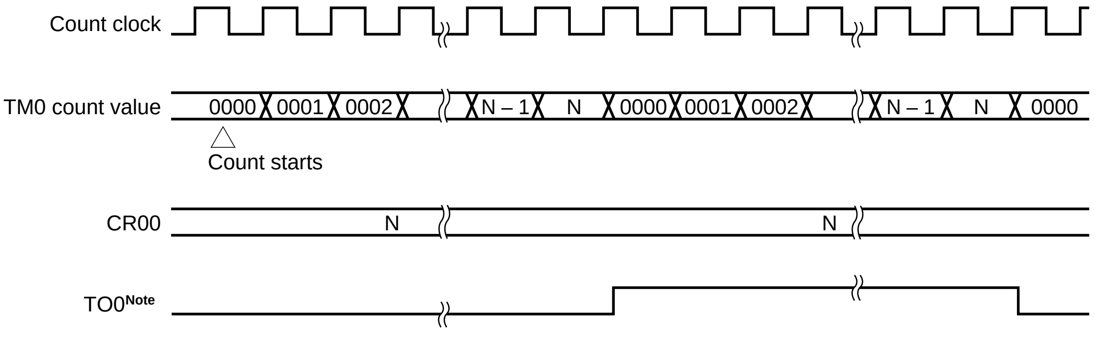
TCL06	TCL05	TCL04	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
0	1	0	2^2 × 1/fx (400 ns)	2^{17} × 1/fx (13.1 ms)	2 × 1/fx (200 ns)
0	1	1	2^3 × 1/fx (800 ns)	2^{18} × 1/fx (26.2 ms)	2^2 × 1/fx (400 ns)
1	0	0	2^4 × 1/fx (1.6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

Remarks **1.** fx : Main system clock oscillation frequency

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

3. ( ) : At fx = 10.0 MHz operation

Figure 8-19. Square Wave Output Timing



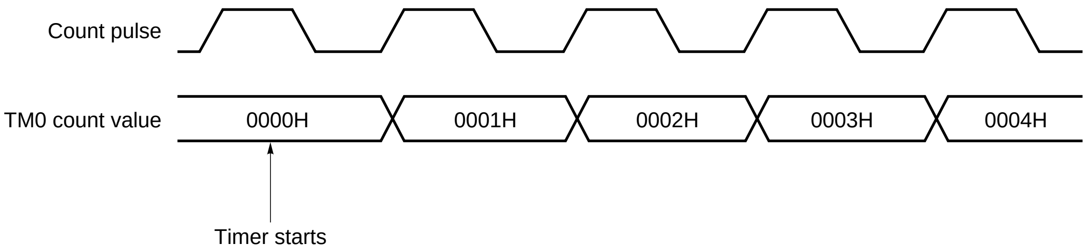
Note The initial value of TO0 output can be set by LVS0 and LVR0.

8.6 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 8-20. Start Timing of 16-Bit Timer Register



(2) Setting of 16-bit compare register

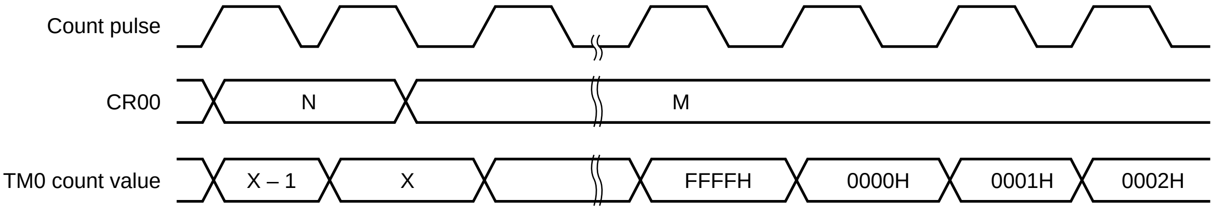
Set a value other than 0000H to the 16-bit compare register (CR00).

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 8-21. Timing after Changing Value of Compare Register during Timer Count Operation

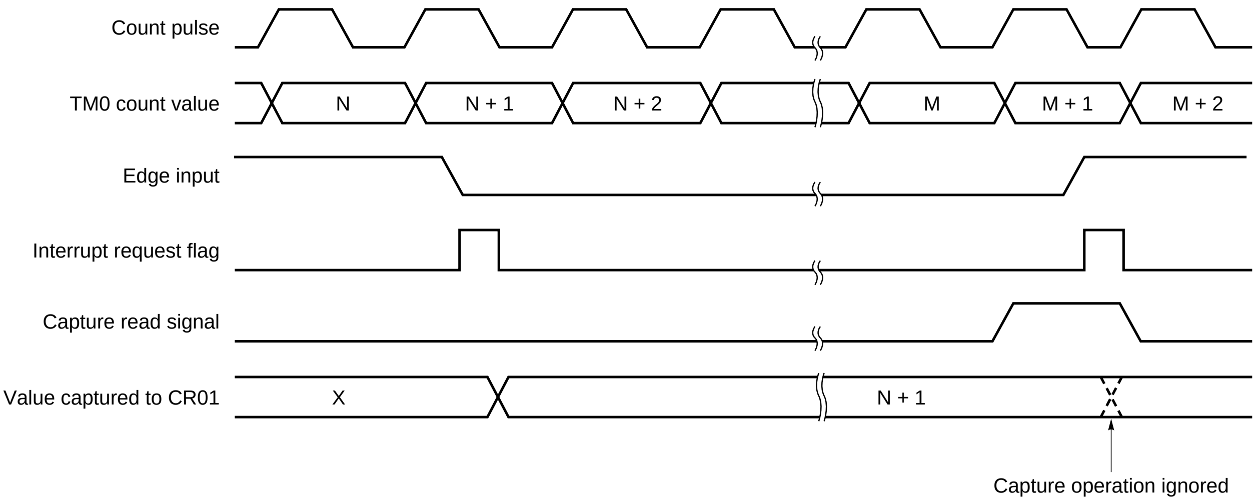


Remark N > X > M

(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 8-22. 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 through 3 (TMC01-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

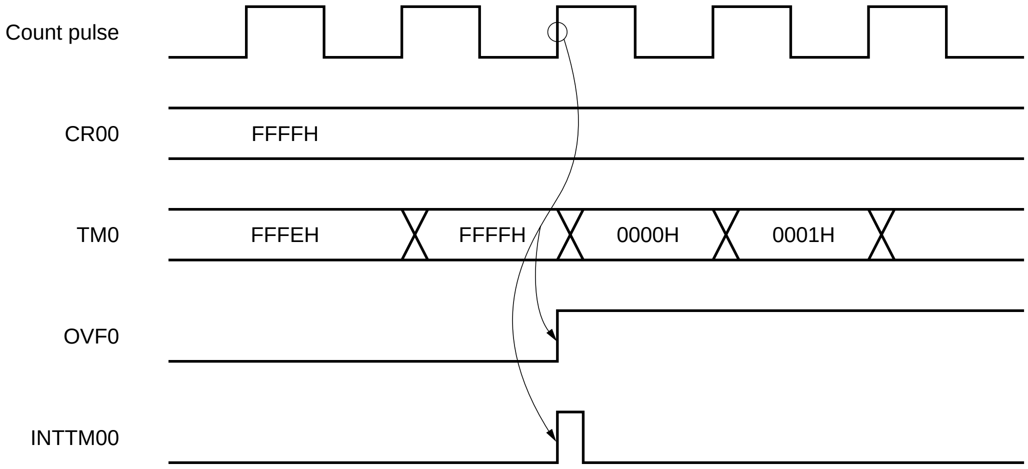
↓

CR00 is set to FFFFH

↓

TM0 counts up from FFFFH to 0000H

Figure 8-23. Operation Timing of OVF0 Flag



CHAPTER 9 8-BIT TIMER/EVENT COUNTER

9.1 Function of 8-Bit Timer/Event Counter

The μPD78018F, 78018FY 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.

9.1.1 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 9-1. 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 9-2. 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

9.1.2 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 9-3. 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 9-4. 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

9.2 Configuration of 8-Bit Timer/Event Counter

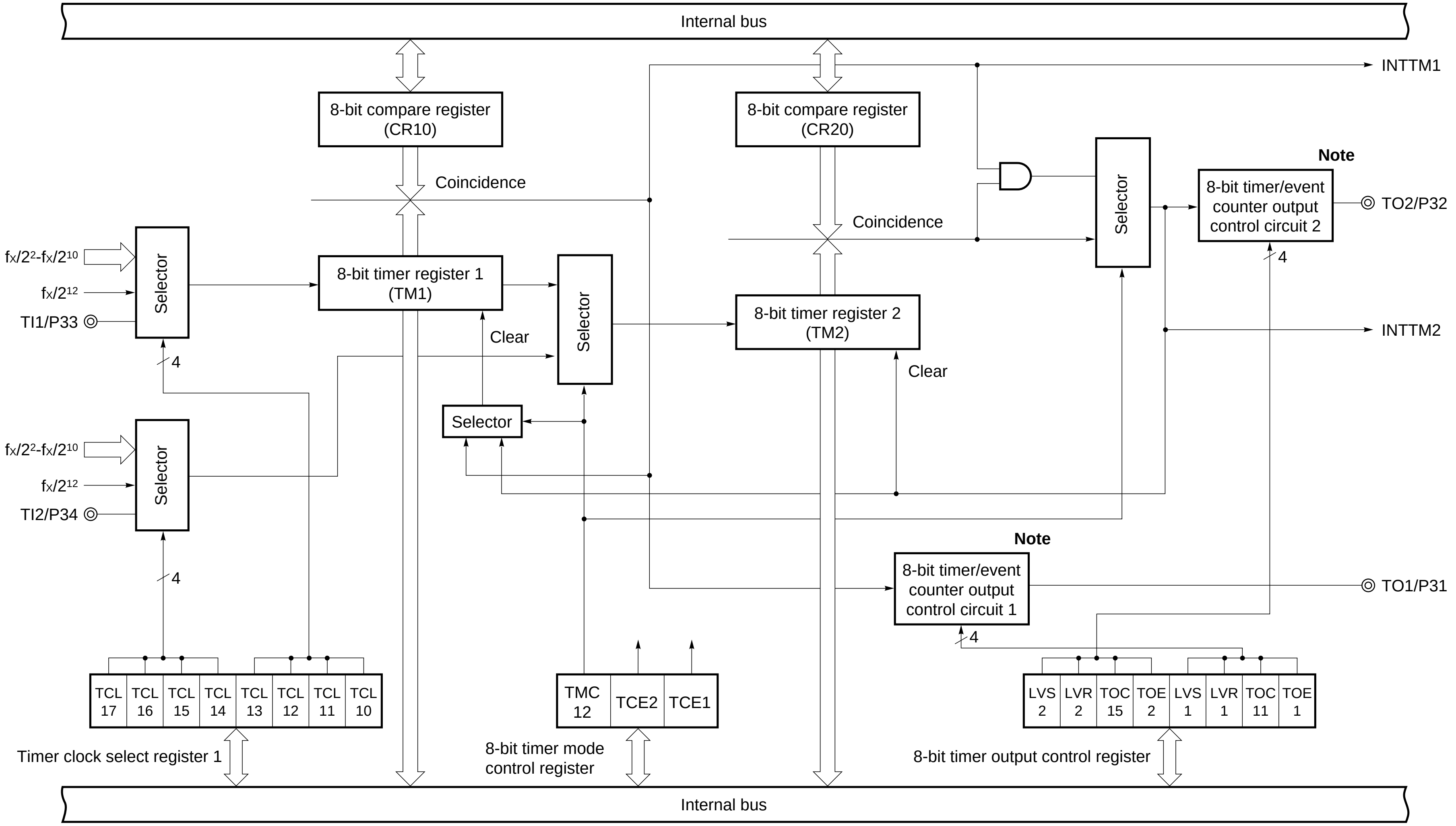
An 8-bit timer/event counter consists of the following hardware:

Table 9-5. 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)

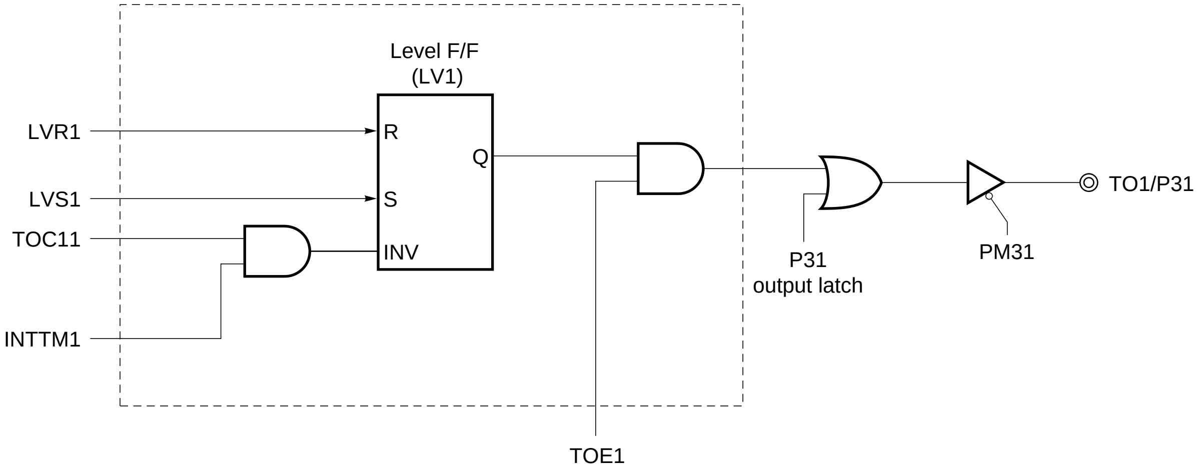
Note Refer to **Figure 6-10 Block Diagram of P30-P37**.

Figure 9-1. Block Diagram of 8-Bit Timer/Event Counter


Internal bus
INTTM1
8-bit compare register (CR10)
8-bit compare register (CR20)
Note
Coincidence
Coincidence
Selector
8-bit timer/event counter output control circuit 2
TO2/P32
4
fx/2²-fx/2¹⁰
fx/2¹²
TI1/P33
Selector
8-bit timer register 1 (TM1)
Selector
8-bit timer register 2 (TM2)
Clear
Clear
INTTM2
4
Selector
fx/2²-fx/2¹⁰
fx/2¹²
TI2/P34
Selector
Note
4
8-bit timer/event counter output control circuit 1
TO1/P31
4
TCL 17
TCL 16
TCL 15
TCL 14
TCL 13
TCL 12
TCL 11
TCL 10
TMC 12
TCE2
TCE1
LVS 2
LVR 2
TOC 15
TOE 2
LVS 1
LVR 1
TOC 11
TOE 1
Timer clock select register 1
8-bit timer mode control register
8-bit timer output control register
Internal bus

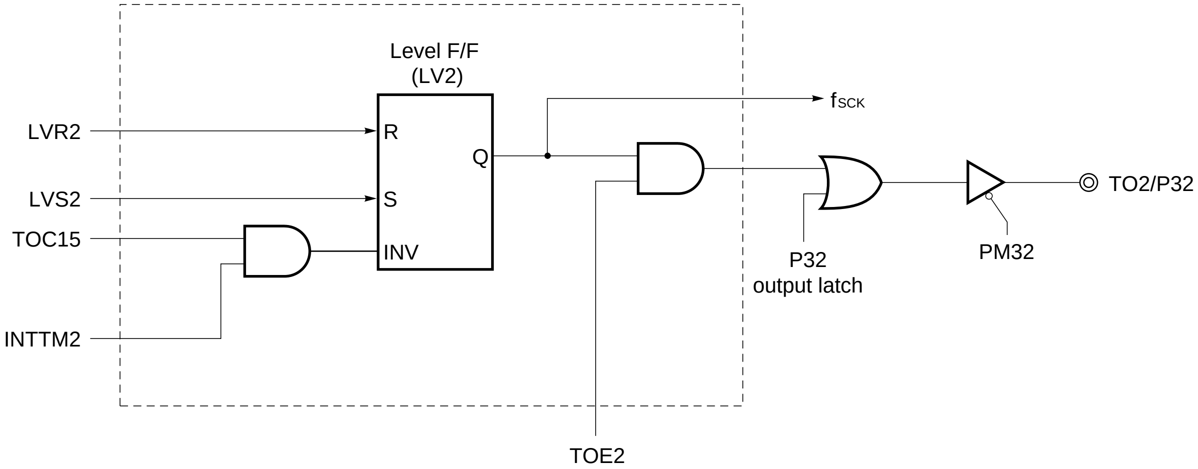
Note For the configuration of the 8-bit timer/event counter output control circuits 1 and 2, refer to **Figures 9-2** and **9-3**.

Figure 9-2. Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 1



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

Figure 9-3. Block Diagram of 8-Bit Timer/Event Counter Output Control Circuit 2



Remarks **1.** The output control circuit is shown enclosed by dotted 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-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-FFFFH can be set to the two compare registers.

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

Caution 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) 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.

9.3 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 9-4. Format of Timer Clock Select Register 1

Symbol	7	6	5	4	3	2	1	0	Address	On 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 Before writing data other than that already written to TCL1, stop the timer operation.

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 9-5. Format of 8-Bit Timer Mode Control Register

Symbol	7	6	5	4	3	2	<1>	<0>	Address	On 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. Before changing the operation mode, stop the timer operations.

2. When TM1 and TM2 are used together as a 16-bit timer register (TMS), set enable or disable of the operation by using the TCE1.

(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 9-6. Format of 8-Bit Timer Output Control Register

Symbol	<7>	<6>	5	<4>	<3>	<2>	1	<0>	Address	On 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 9-7. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	On 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 = 0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

9.4 Operation of 8-Bit Timer/Event Counter

9.4.1 8-bit timer/event counter mode

(1) Operation as interval timer

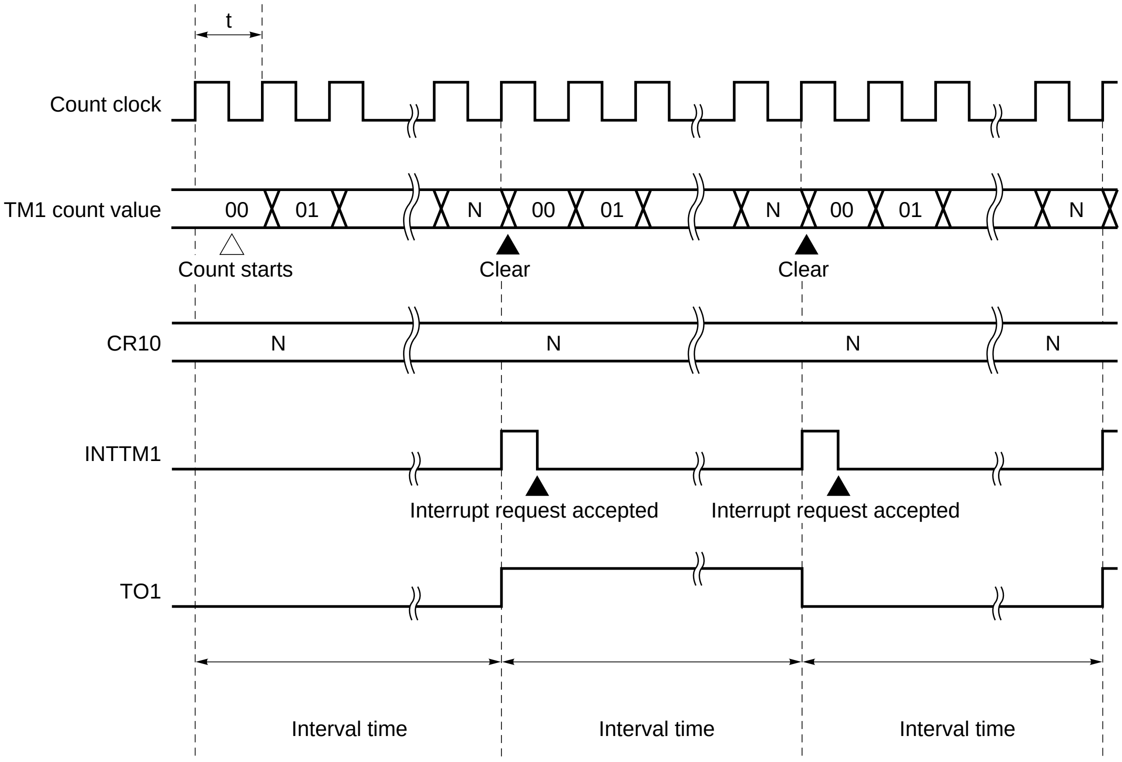
The 8-bit timer/event counters operate as interval timers and repeatedly generate an interrupt request at time intervals specified by the count values set to the corresponding 8-bit compare registers (CR10 and CR20) in advance.

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

For the operation when compare register value is changed during timer count operation, refer to **8.6 Notes on Using 16-bit Timer/Event Counter (3)**.

Figure 9-8. Interval Timer Operation Timing



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

Table 9-6. 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} × TI1 input cycle	TI1 input edge cycle
0	0	0	1	TI1 input cycle	2^{8} × TI1 input cycle	TI1 input edge cycle
0	1	1	0	2^{2} × 1/fx (400 ns)	2^{10} × 1/fx (102.4 μs)	2^{2} × 1/fx (400 ns)
0	1	1	1	2^{3} × 1/fx (800 ns)	2^{11} × 1/fx (204.8 μs)	2^{3} × 1/fx (800 ns)
1	0	0	0	2^{4} × 1/fx (1.6 μs)	2^{12} × 1/fx (409.6 μs)	2^{4} × 1/fx (1.6 μs)
1	0	0	1	2^{5} × 1/fx (3.2 μs)	2^{13} × 1/fx (819.2 μs)	2^{5} × 1/fx (3.2 μs)
1	0	1	0	2^{6} × 1/fx (6.4 μs)	2^{14} × 1/fx (1.64 ms)	2^{6} × 1/fx (6.4 μs)
1	0	1	1	2^{7} × 1/fx (12.8 μs)	2^{15} × 1/fx (3.28 ms)	2^{7} × 1/fx (12.8 μs)
1	1	0	0	2^{8} × 1/fx (25.6 μs)	2^{16} × 1/fx (6.55 ms)	2^{8} × 1/fx (25.6 μs)
1	1	0	1	2^{9} × 1/fx (51.2 μs)	2^{17} × 1/fx (13.1 ms)	2^{9} × 1/fx (51.2 μs)
1	1	1	0	2^{10} × 1/fx (102.4 μs)	2^{18} × 1/fx (26.2 ms)	2^{10} × 1/fx (102.4 μs)
1	1	1	1	2^{12} × 1/fx (409.6 μs)	2^{20} × 1/fx (104.9 ms)	2^{12} × 1/fx (409.6 μs)
Others				Setting prohibited		

Remarks 1. fx : Main system clock oscillation frequency

2. TCL10-TCL13 : Bits 0 through 3 of timer clock select register 1 (TCL1)

3. ( ) : At fx = 10.0 MHz operation

Table 9-7. 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} × TI2 input cycle	TI2 input edge cycle
0	0	0	1	TI2 input cycle	2^{8} × TI2 input cycle	TI2 input edge cycle
0	1	1	0	2^{2} × 1/fx (400 ns)	2^{10} × 1/fx (102.4 μs)	2^{2} × 1/fx (400 ns)
0	1	1	1	2^{3} × 1/fx (800 ns)	2^{11} × 1/fx (204.8 μs)	2^{3} × 1/fx (800 ns)
1	0	0	0	2^{4} × 1/fx (1.6 μs)	2^{12} × 1/fx (409.6 μs)	2^{4} × 1/fx (1.6 μs)
1	0	0	1	2^{5} × 1/fx (3.2 μs)	2^{13} × 1/fx (819.2 μs)	2^{5} × 1/fx (3.2 μs)
1	0	1	0	2^{6} × 1/fx (6.4 μs)	2^{14} × 1/fx (1.64 ms)	2^{6} × 1/fx (6.4 μs)
1	0	1	1	2^{7} × 1/fx (12.8 μs)	2^{15} × 1/fx (3.28 ms)	2^{7} × 1/fx (12.8 μs)
1	1	0	0	2^{8} × 1/fx (25.6 μs)	2^{16} × 1/fx (6.55 ms)	2^{8} × 1/fx (25.6 μs)
1	1	0	1	2^{9} × 1/fx (51.2 μs)	2^{17} × 1/fx (13.1 ms)	2^{9} × 1/fx (51.2 μs)
1	1	1	0	2^{10} × 1/fx (102.4 μs)	2^{18} × 1/fx (26.2 ms)	2^{10} × 1/fx (102.4 μs)
1	1	1	1	2^{12} × 1/fx (409.6 μs)	2^{20} × 1/fx (104.9 ms)	2^{12} × 1/fx (409.6 μs)
Others				Setting prohibited		

Remarks 1. fx : Main system clock oscillation frequency

2. TCL14-TCL17 : Bits 4 through 7 of timer clock select register 1 (TCL1)

3. ( ) : At fx = 10.0 MHz operation

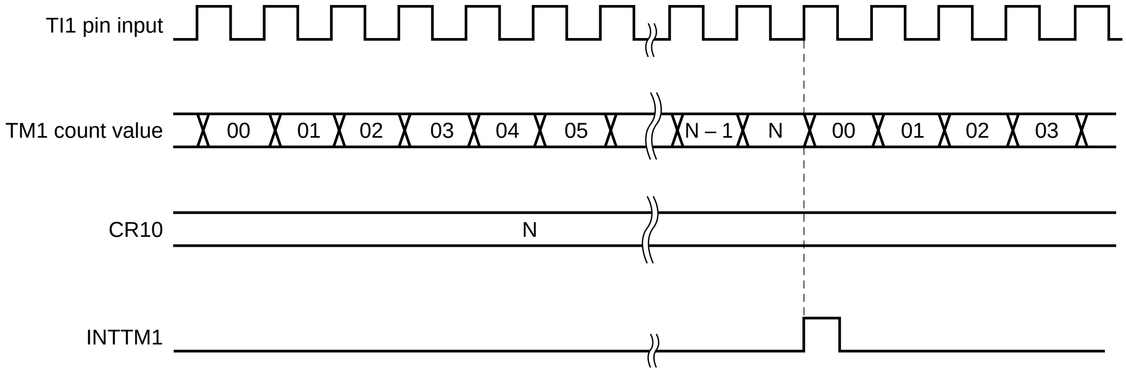
(2) Operation as external event counter

The external event counter counts the number of clock pulses externally input to the TI1/P33 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 9-9. External Event Counter Operation Timing (with rising edge specified)



Remark N = 00H-FFH

(3) Operation as square wave output

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

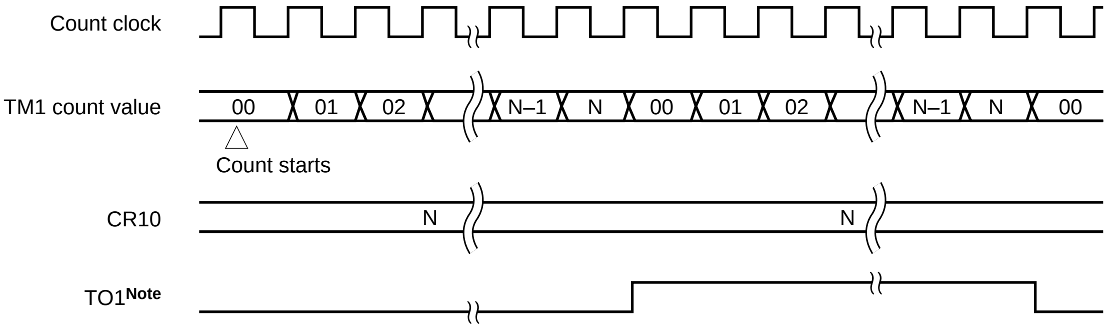
When bit 0 or 4 (TOE1 or TOE2) of the 8-bit timer output control register (TOC1) is set to 1, the output status of the TO1/P31 or TO2/P32 pin is inverted at time intervals specified by the values set to CR10 or CR20 in advance. In this way, square waves of any frequency can be output.

Table 9-8. 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	$2^{2} \times 1/f_X$ (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)

Remarks **1.** f_X : Main system clock oscillation frequency
2. TCL10-TCL13: Bits 0 through 3 of timer clock select register 1 (TCL1)
3. ( ) : At f_X = 10.0 MHz operation

★ **Figure 9-10. Square Wave Output Timing**



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

9.4.2 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.

In this mode, the count clock is selected by using the bits 0 through 3 (TCL10 through TCL13) of the timer clock select register (TCL1), and the overflow signal of the 8-bit timer/event counter 1 (TM1) is used as the count clock of the 8-bit timer/event counter 2 (TM2).

In this mode, counting is disabled or enabled by bit 0 (TCE1) of TMC1.

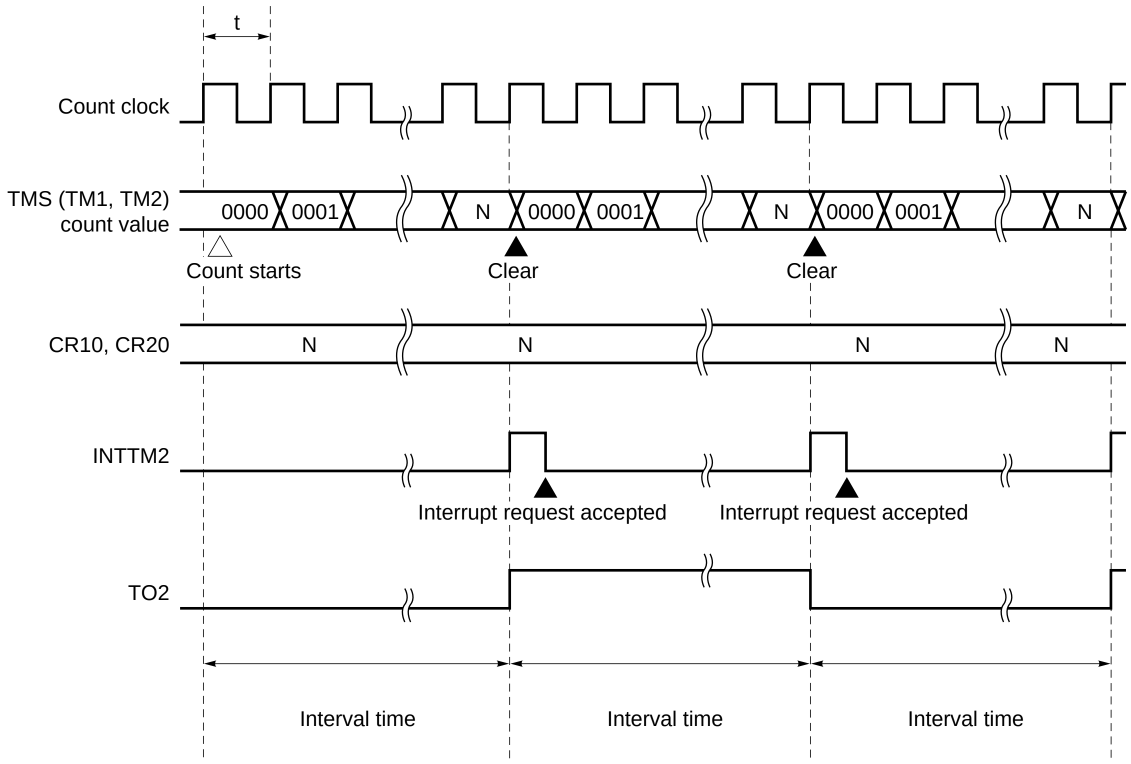
(1) Operation as interval timer

The two channels of 8-bit timer/event counters are used as a 16-bit interval timer that repeatedly generates an interrupt request at time intervals specified by the count values set to the two 8-bit compare registers (CR10 and CR20) in advance. When setting a count value, write the value of the high-order 8 bits to CR20 and the value of the low-order 8 bits to CR10. For the count value that can be set (interval time), refer to **Table 9-9**.

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, an interrupt request signal (INTTM2) is generated. For the operation timing of the interval timer, refer to **Figure 9-11**.

The count clock can be selected by the bits 0 through 3 (TCL10-TCL13) of the timer clock select register 1 (TCL1). The overflow signal of TM1 is used as the count clock for TM2.

Figure 9-11. Interval Timer Operation Timing



Remark Interval time = (N + 1) x t: N = 0000H-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 9-9. 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-TCL13: Bits 0 through 3 of timer clock select register 1 (TCL1)

3. ( ) : At fx = 10.0 MHz operation

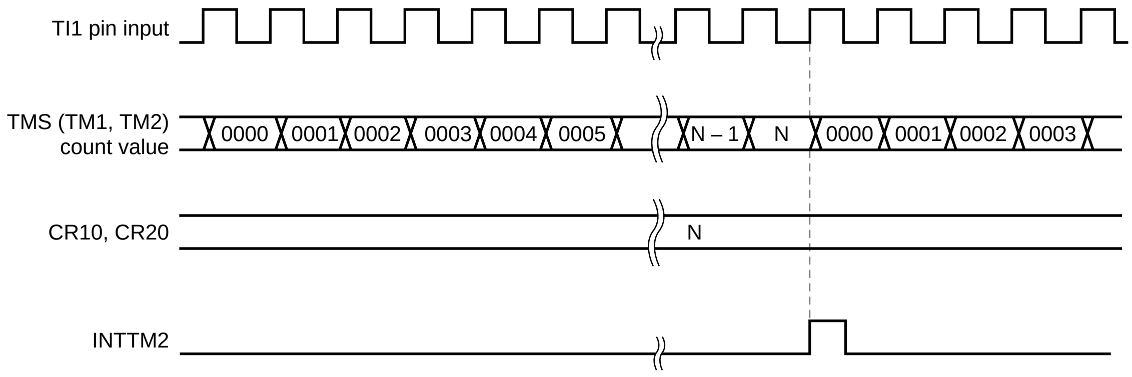
(2) Operation as external event counter

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, the values of TM1 and TM2 are incremented. 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 9-12. 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 operate square wave output 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 a count value, write the value of the high-order 8 bits to CR20 and the value of the low-order 8 bits 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 9-10. Square Wave Output Range when Two 8-Bit Timer/Event Counters (TM1 and TM2) Are Used as One 16-Bit Timer/Event Counter

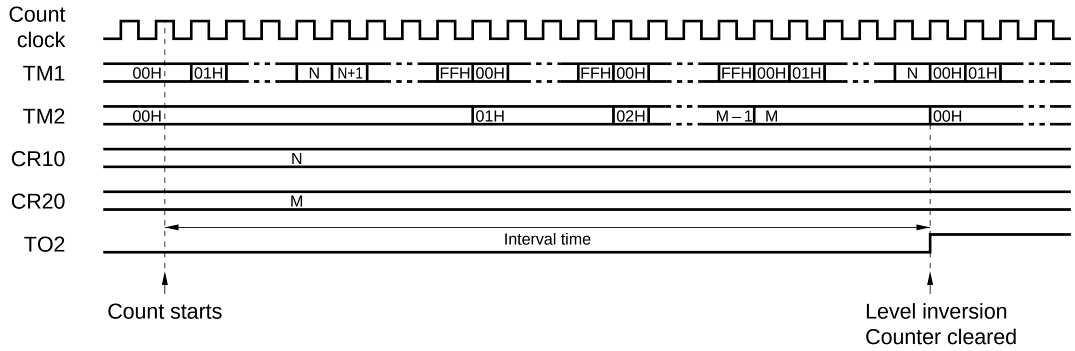
TCL13	TCL12	TCL11	TCL10	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	1	1	0	$2^{2} \times 1/f_X$ (400 ns)	$2^{18} \times 1/f_X$ (26.2 ms)	$2^{2} \times 1/f_X$ (400 ns)
0	1	1	1	$2^{3} \times 1/f_X$ (800 ns)	$2^{19} \times 1/f_X$ (52.4 ms)	$2^{3} \times 1/f_X$ (800 ns)
1	0	0	0	$2^{4} \times 1/f_X$ (1.6 μs)	$2^{20} \times 1/f_X$ (104.9 ms)	$2^{4} \times 1/f_X$ (1.6 μs)
1	0	0	1	$2^{5} \times 1/f_X$ (3.2 μs)	$2^{21} \times 1/f_X$ (209.7 ms)	$2^{5} \times 1/f_X$ (3.2 μs)
1	0	1	0	$2^{6} \times 1/f_X$ (6.4 μs)	$2^{22} \times 1/f_X$ (419.4 ms)	$2^{6} \times 1/f_X$ (6.4 μs)
1	0	1	1	$2^{7} \times 1/f_X$ (12.8 μs)	$2^{23} \times 1/f_X$ (838.9 ms)	$2^{7} \times 1/f_X$ (12.8 μs)
1	1	0	0	$2^{8} \times 1/f_X$ (25.6 μs)	$2^{24} \times 1/f_X$ (1.7 s)	$2^{8} \times 1/f_X$ (25.6 μs)
1	1	0	1	$2^{9} \times 1/f_X$ (51.2 μs)	$2^{25} \times 1/f_X$ (3.4 s)	$2^{9} \times 1/f_X$ (51.2 μs)
1	1	1	0	$2^{10} \times 1/f_X$ (102.4 μs)	$2^{26} \times 1/f_X$ (6.7 s)	$2^{10} \times 1/f_X$ (102.4 μs)
1	1	1	1	$2^{12} \times 1/f_X$ (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. TCL10-TCL13: Bits 0 through 3 of timer clock select register 1 (TCL1)

3. ( ): At f_X = 10.0 MHz operation

★ **Figure 9-13. Square Wave Output Timing**

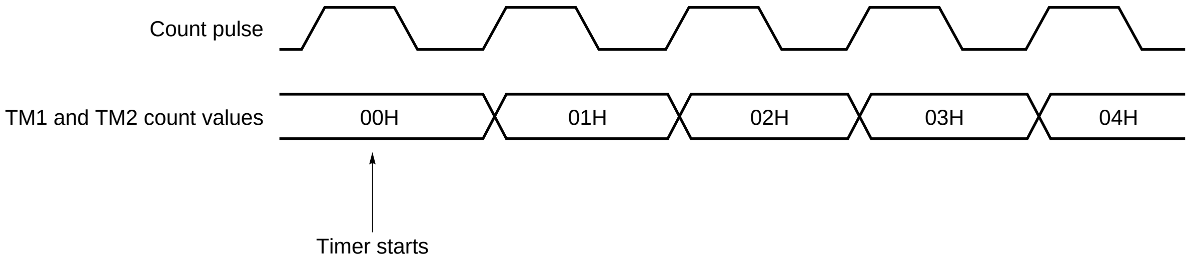


9.5 Notes on Using 8-Bit Timer/Event Counters

(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 9-14. Start Timing of 8-Bit Timer Register



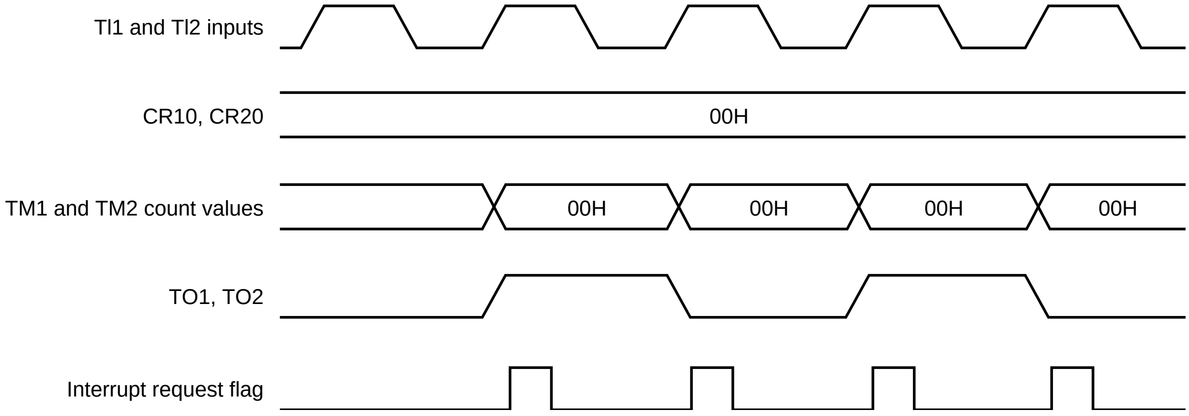
(2) Setting of 8-bit compare registers

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 to 0 and stop the timers, in order to write values to CR10 and CR20.

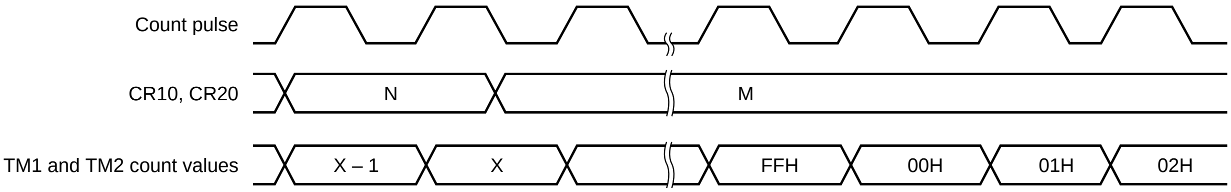
Figure 9-15. External Event Counter Operation Timing



(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 9-16. Timing after Changing Values of Compare Registers during Timer Count Operation



Remark N > X > M

CHAPTER 10 WATCH TIMER

10.1 Functions of Watch Timer

The watch timer has the following functions:

- Watch timer
- Interval timer

The watch timer and 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 10-1. 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

10.2 Configuration of Watch Timer

The watch timer consists of the following hardware:

Table 10-2. Configuration of Watch Timer

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

10.3 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 10-2.**)
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 10-1. Block Diagram of Watch Timer

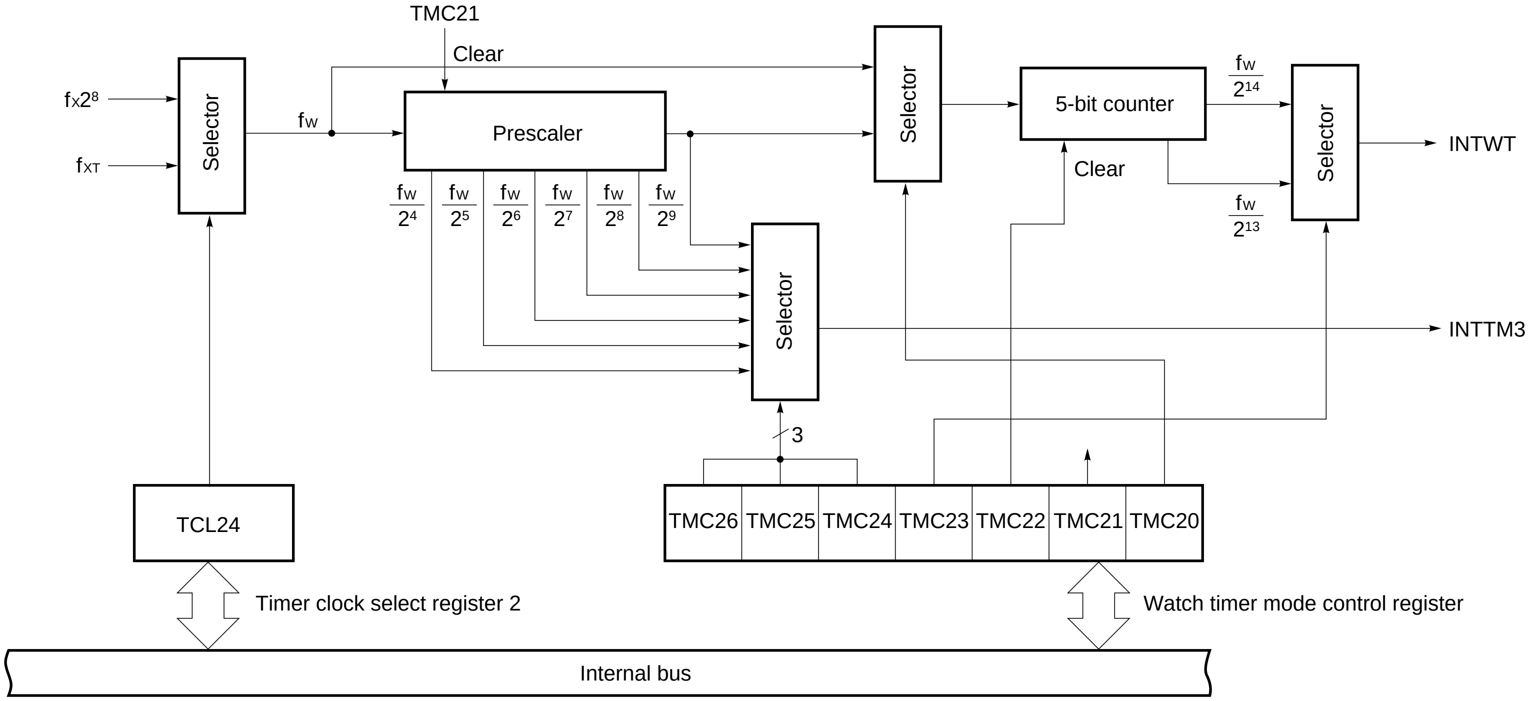


Figure 10-2. Format of Timer Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	On 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 Before writing new data to TCL2, stop the timer operation.

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 10-3. Format of Watch Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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	0	$2^{13}/f_W$ (0.25 s)
0	1	$2^{5}/f_W$ (977 μs)
1	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

10.4 Operation of Watch Timer

10.4.1 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 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).

10.4.2 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 through 6 (TMC24-TMC26) of the watch timer mode control register.

Table 10-3. 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})

[MEMO]

CHAPTER 11 WATCHDOG TIMER

11.1 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 inadvertent program loop. When the inadvertent loop is detected, a non-maskable interrupt request or the $\overline{\text{RESET}}$ signal can be generated.

Table 11-1. Inadvertent Loop Detection Time of Watchdog Timer

Inadvertent Loop Detection Time	At f_X = 10.0 MHz	Inadvertent Loop 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

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 at time intervals set in advance.

Table 11-2. 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

f_X: main system clock oscillation frequency

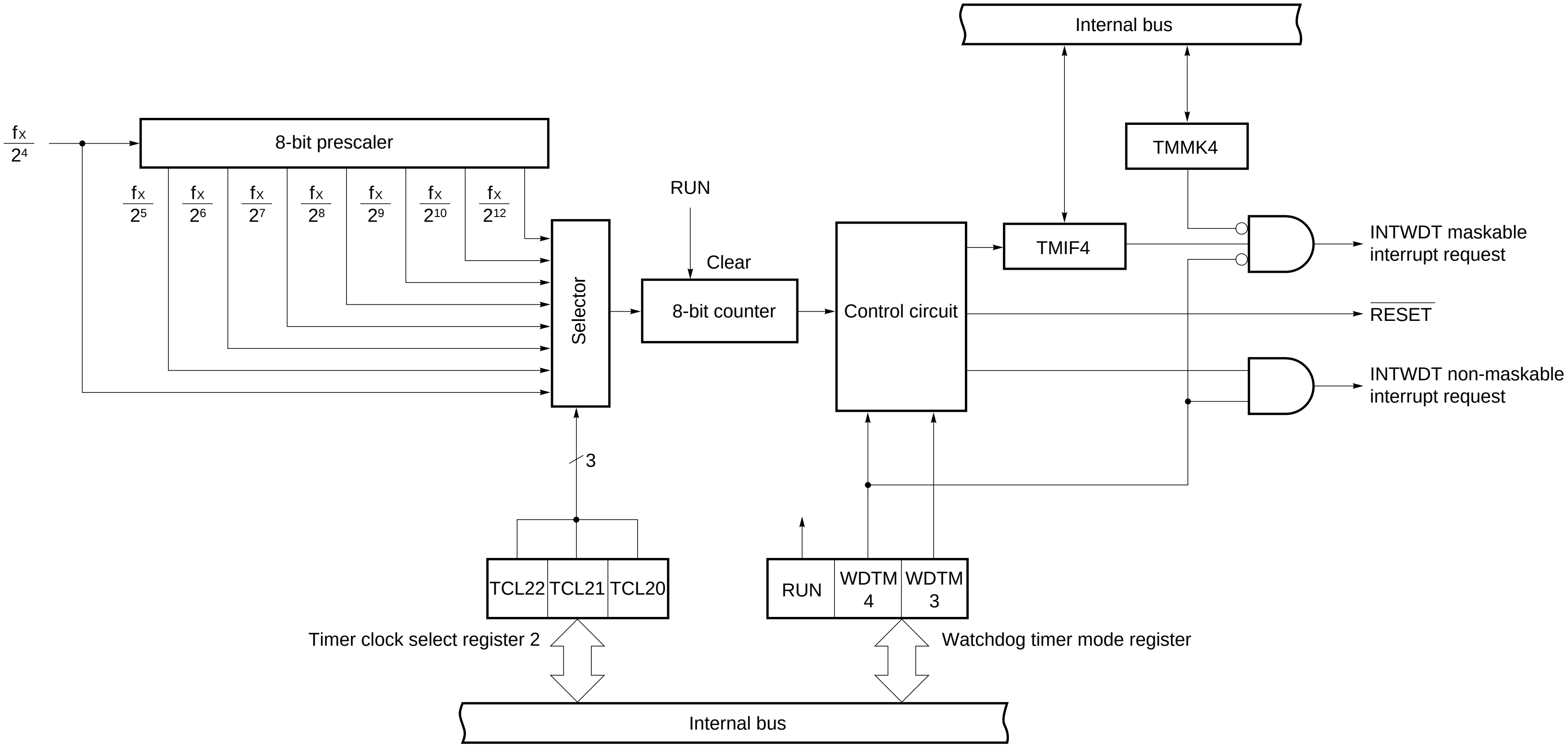
11.2 Configuration of Watchdog Timer

The watchdog timer consists of the following hardware:

Table 11-3. Configuration of Watchdog Timer

Item	Configuration
Control register	Timer clock select register 2 (TCL2) Watchdog timer mode register (WDTM)

★ Figure 11-1. Block Diagram of Watchdog Timer



11.3 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 11-2)

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 11-2. Format of Timer Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	On 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 Before writing new data to TCL2, stop the timer operation.

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

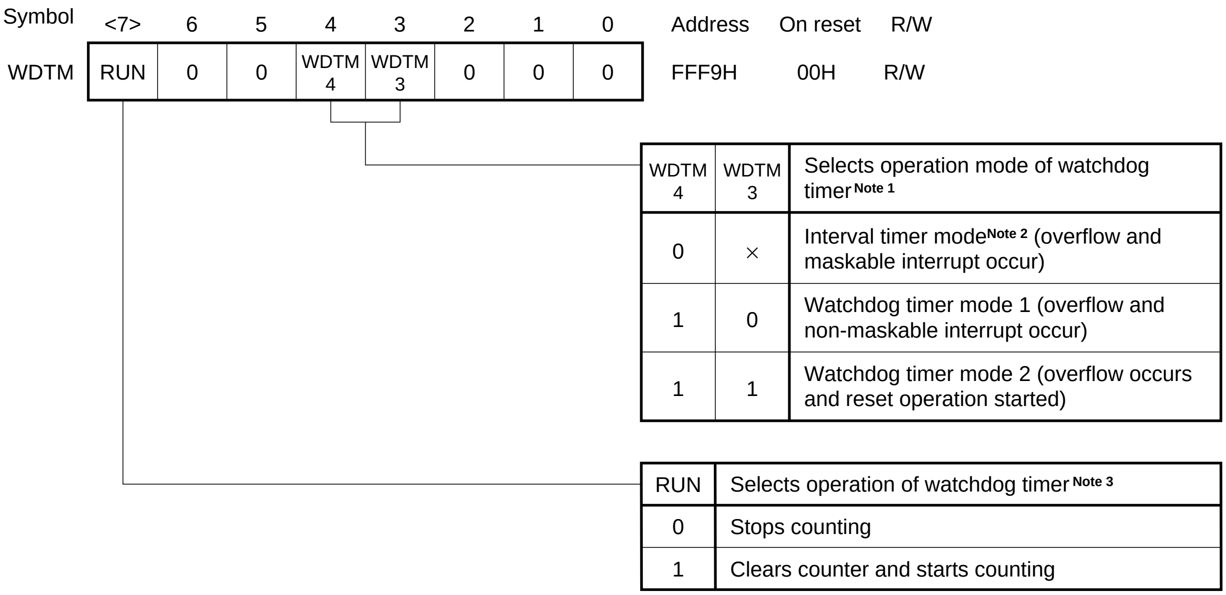
(2) Watchdog timer mode register (WDTM)

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 11-3. Format of Watchdog Timer Mode Register



Symbol	<7>	6	5	4	3	2	1	0	Address	On reset	R/W
WDTM	RUN	0	0	WDTM 4	WDTM 3	0	0	0	FFF9H	00H	R/W

WDTM 4	WDTM 3	Selects operation mode of watchdog timer[Note 1]
0	×	Interval timer mode[Note 2] (overflow and maskable interrupt occur)
1	0	Watchdog timer mode 1 (overflow and non-maskable interrupt 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 is generated regardless of the contents of WDTM3.

Remark ×: Don’t care

11.4 Operation of Watchdog Timer

11.4.1 Operation as watchdog timer

The watchdog timer detects an inadvertent program loop when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1.

The count clock (inadvertent loop detection time interval) of the watchdog timer can be selected by bits 0 through 2 (TCL20-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 inadvertent loop 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 inadvertent loop detection time is exceeded, the system is reset or a non-maskable interrupt 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 inadvertent loop 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 11-4. Inadvertent Loop Detection Time of Watchdog Timer

TCL22	TCL21	TCL20	Inadvertent Loop 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

Remarks 1. f_X : Main system clock oscillation frequency

2. TCL20-TCL22 : Bits 0 through 2 of timer clock select register 2 (TCL2)

11.4.2 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 11-5. 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

Remarks 1. f_X : Main system clock oscillation frequency

2. TCL20-TCL22: Bits 0 through 2 of timer clock select register 2 (TCL2)

CHAPTER 12 CLOCK OUTPUT CONTROL CIRCUIT

12.1 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.

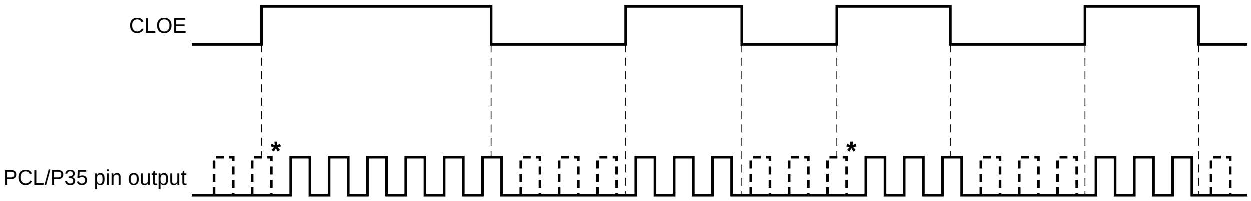
The clock pulse is output in the following procedure:

<1> Select the output frequency of the clock pulse by using bits 0-3 (TCL00-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 (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 12-1**).

Figure 12-1. Application Example of Remote Controller Output



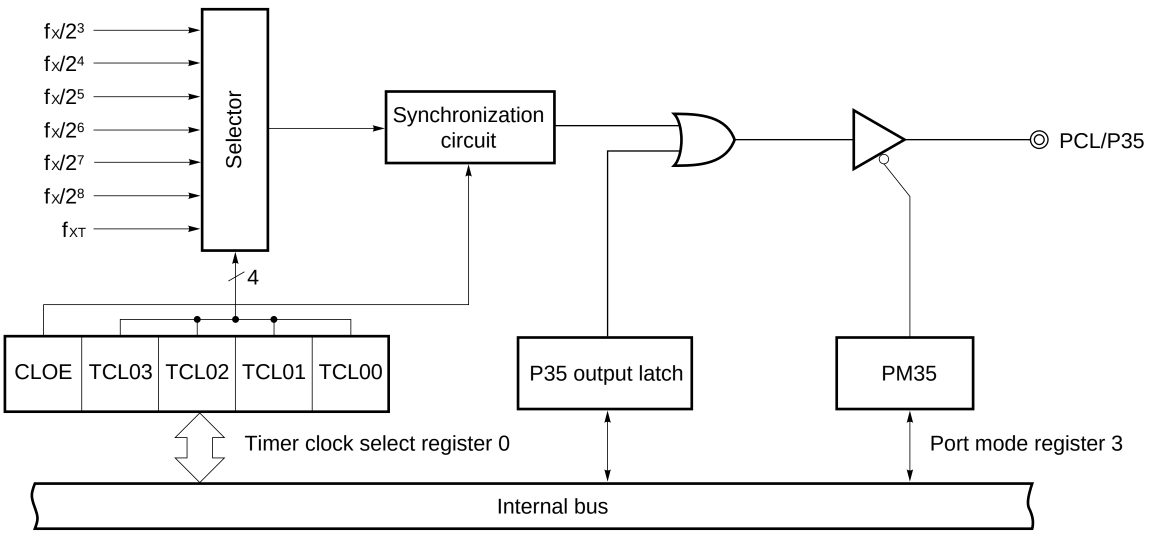
12.2 Configuration of Clock Output Control Circuit

The clock output control circuit consists of the following hardware:

Table 12-1. Configuration of Clock Output Control Circuit

Item	Configuration
Control register	Timer clock select register 0 (TCL0) Port mode register 3 (PM3)

Figure 12-2. Block Diagram of Clock Output Control Circuit



12.3 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 12-3. Format of Timer Clock Select Register 0

Symbol	<7>	6	5	4	3	2	1	0	Address	On 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. The frequency of a sampling clock is selected by the sampling clock select register.

2. To enable PCL output, set TCL00 through 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. Before writing data other than that already written to TCL0, stop the timer operation.

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 12-4. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	On 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 = 0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Phase-out/Discontinued

CHAPTER 13 BUZZER OUTPUT CONTROL CIRCUIT

13.1 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 (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 through 7 (TCL25-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 (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.

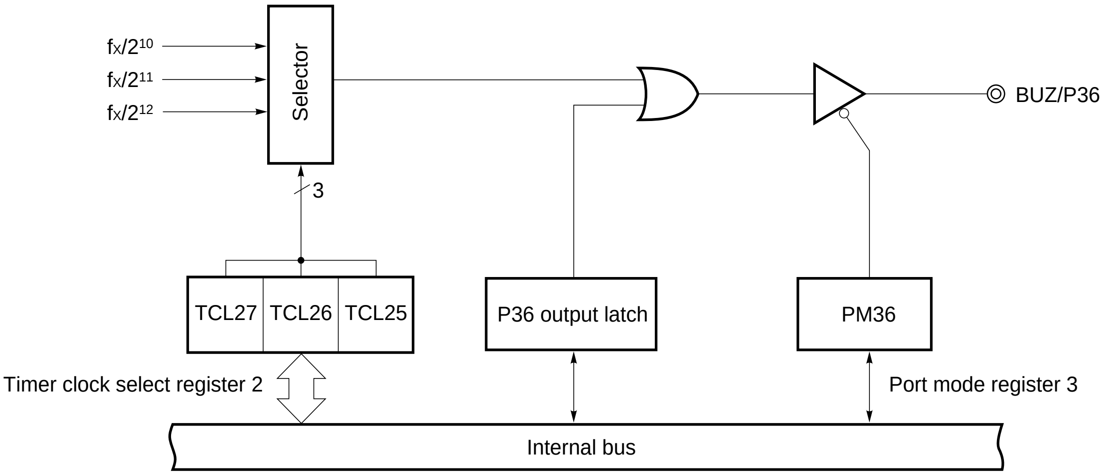
13.2 Configuration of Buzzer Output Control Circuit

The buzzer output control circuit consists of the following hardware:

Table 13-1. Configuration of Buzzer Output Control Circuit

Item	Configuration
Control register	Timer clock select register 2 (TCL2) Port mode register 3 (PM3)

Figure 13-1. Block Diagram of Buzzer Output Control Circuit



13.3 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 13-2. Format of Timer Clock Select Register 2

Symbol	7	6	5	4	3	2	1	0	Address	On 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 Before writing data other than that already written to TCL2, stop the timer operation.

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 13-3. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	On 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 = 0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

CHAPTER 14 A/D CONVERTER

14.1 Function of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of eight channels (ANI0-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 of analog input from ANI0-ANI7, and execute A/D conversion. When A/D conversion has been started by means of hardware start, conversion is stopped after the operation has been completed, and an interrupt request (INTAD) is generated. When A/D conversion has been started by means of software start, conversion is repeatedly performed. Each time conversion has been completed once, an interrupt request (INTAD) is generated.

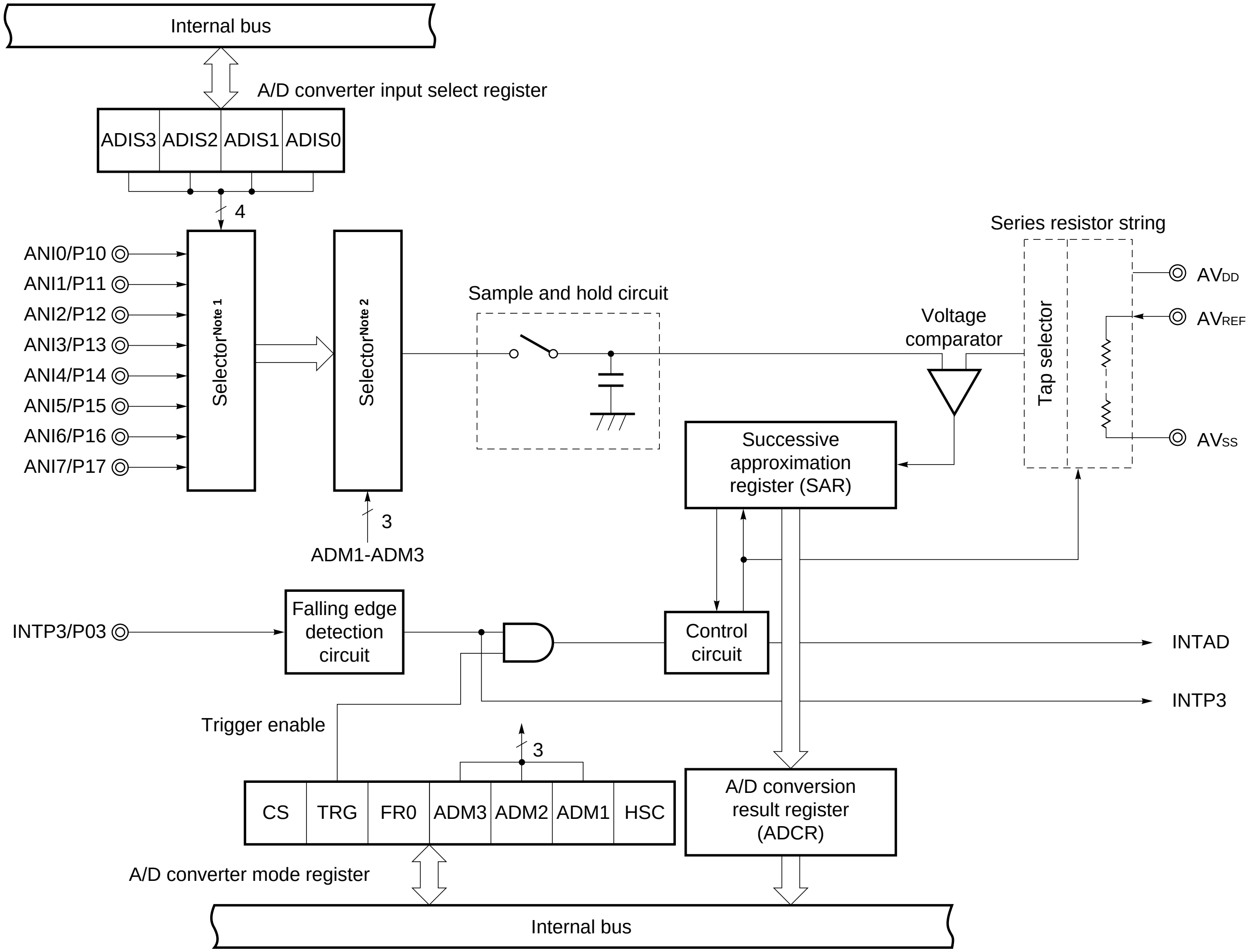
14.2 Configuration of A/D Converter

The A/D converter consists of the following hardware:

Table 14-1. Configuration of A/D Converter

Item	Configuration
Analog input	8 channels (ANI0-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 14-1. 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 held down 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.

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-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-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. The analog input pins (ANI0-ANI7) are also used as I/O port pins (port 1). When A/D conversion is performed with any of ANI0-ANI7 selected, do not execute the input instruction for 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.

(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-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, 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.**

(8) AV_{SS} pin

This is a ground pin of the A/D converter. 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 is an analog pin of the A/D converter. 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.

14.3 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 $\overline{RESET}$ signal is input.

Figure 14-2. Format of A/D Converter Mode Register

Symbol	<7>	<6>	5	4	3	2	1	0	Address	On 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[Note1]				
				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[Note2]	19.1 μs	32.0 μs	38.1 μs
0	1	1	80/f_X	Setting prohibited[Note2]	Setting prohibited[Note2]	Setting prohibited[Note2]	19.1 μs
1	0	0	100/f_X	Setting prohibited[Note2]	Setting prohibited[Note2]	20.0 μs	23.9 μs
1	0	1	200/f_X	20.0 μs	23.9 μs	40.0 μs	47.7 μs
other			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 consumption 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-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.

Figure 14-3. Format of A/D Converter Input Select Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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-P17)
0	0	0	1	1 channel (ANI0, P11-P17)
0	0	1	0	2 channels (ANI0, ANI1, P12-P17)
0	0	1	1	3 channels (ANI0 - ANI2, P13-P17)
0	1	0	0	4 channels (ANI0 - ANI3, P14-P17)
0	1	0	1	5 channels (ANI0 - ANI4, P15-P17)
0	1	1	0	6 channels (ANI0 - ANI5, P16-P17)
0	1	1	1	7 channels (ANI0 - ANI6, P17)
1	0	0	0	8 channels (ANI0 - ANI7)
Others				Setting prohibited

14.4 Operation of A/D Converter

14.4.1 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) is set. 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:

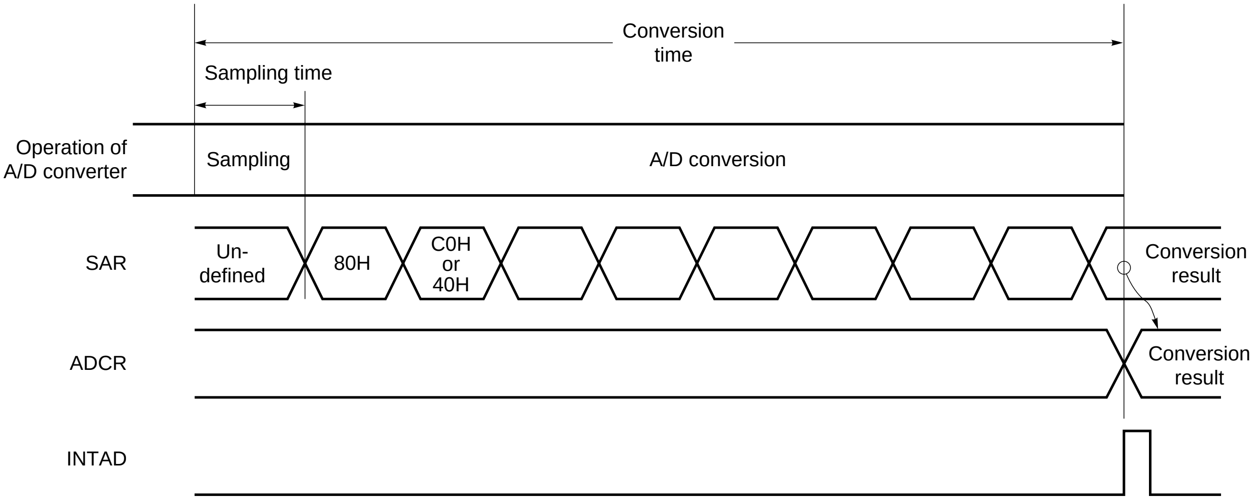
- Analog input voltage ≥ voltage tap : bit 6 = 1
- Analog input voltage < voltage tap : bit 6 = 0

<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 14-4. Basic Operation of A/D Converter



The A/D conversion is performed continuously, until the bit 7 of ADM (CS) 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 register become undefined when the $\overline{\text{RESET}}$ signal is input.

14.4.2 Input voltage and conversion result

The relation between the analog voltage input to the analog input pins (ANI0-ANI7) and A/D conversion result (value stored to A/D conversion result register (ADCR)) is as follows:

$$\mathrm{ADCR} = \mathrm{INT}\left(\frac{V_{IN}}{AV_{REF}} \times 256 + 0.5\right)$$

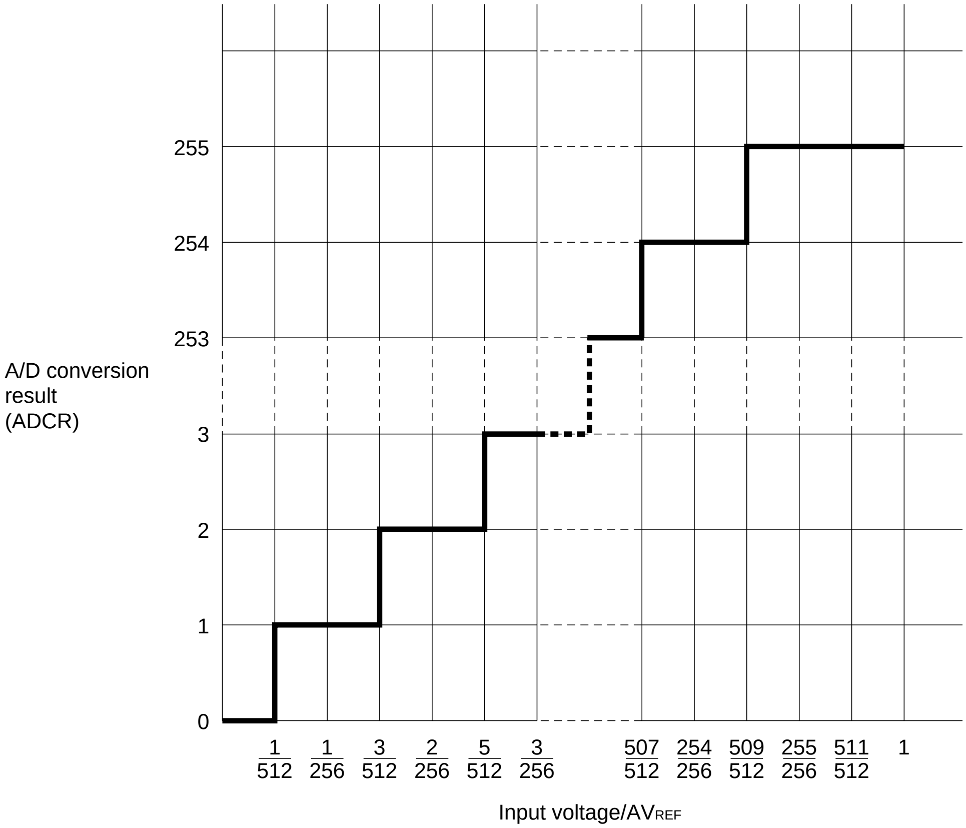
or,

$$(\mathrm{ADCR} - 0.5) \times \frac{AV_{REF}}{256} \leq V_{IN} < (\mathrm{ADCR} + 0.5) \times \frac{AV_{REF}}{256}$$

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 14-5 shows the relations between the analog input voltage and A/D conversion result.

Figure 14-5. Relations between Analog Input Voltage and A/D Conversion Result



14.4.3 Operation mode of A/D converter

Select one analog input channel from ANI0-ANI7 by using the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and start 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

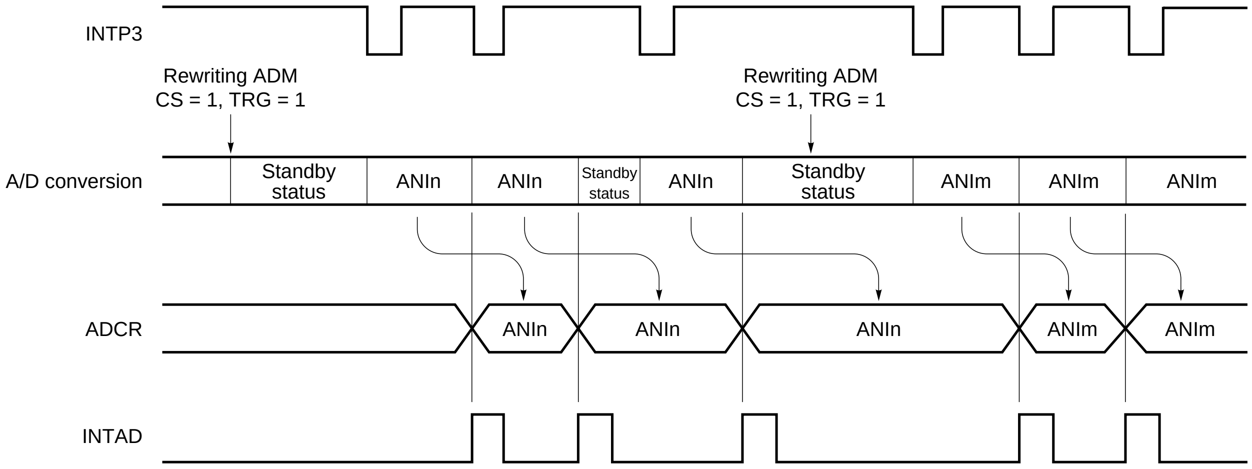
The A/D conversion stands by when both bits 6 (TRG) and 7 (CS) of A/D converter mode register (ADM) are set to 1. When an external trigger signal (INTP3) is input, the voltage applied to the analog input pin specified by bits 1-3 (ADM1-ADM3) of ADM is converted into a digital value.

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 14-6. A/D Conversion by Hardware Start



Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

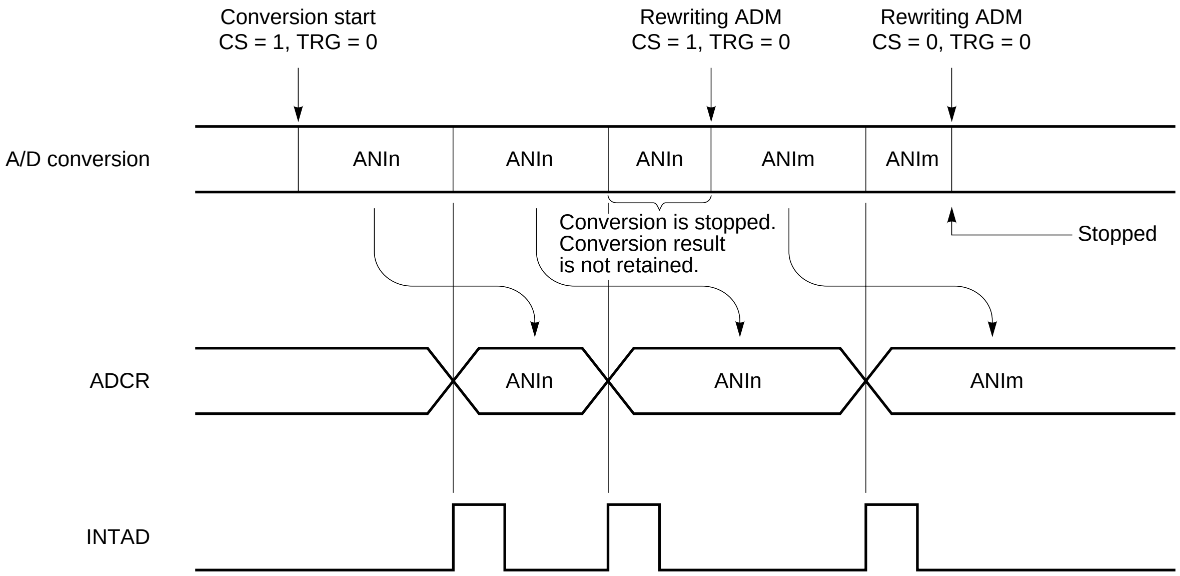
(2) A/D conversion by software start

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 1-3 (ADM1-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 14-7. A/D Conversion by Software Start



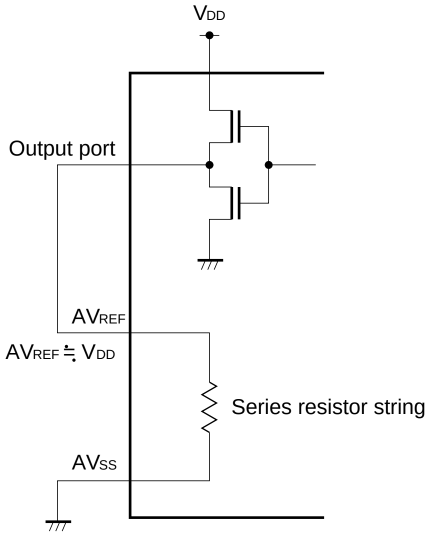
Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

14.5 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 14-8 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 14-8. Example of Reducing Current Consumption in Standby Mode



μPD78018F, 78018FY subseries

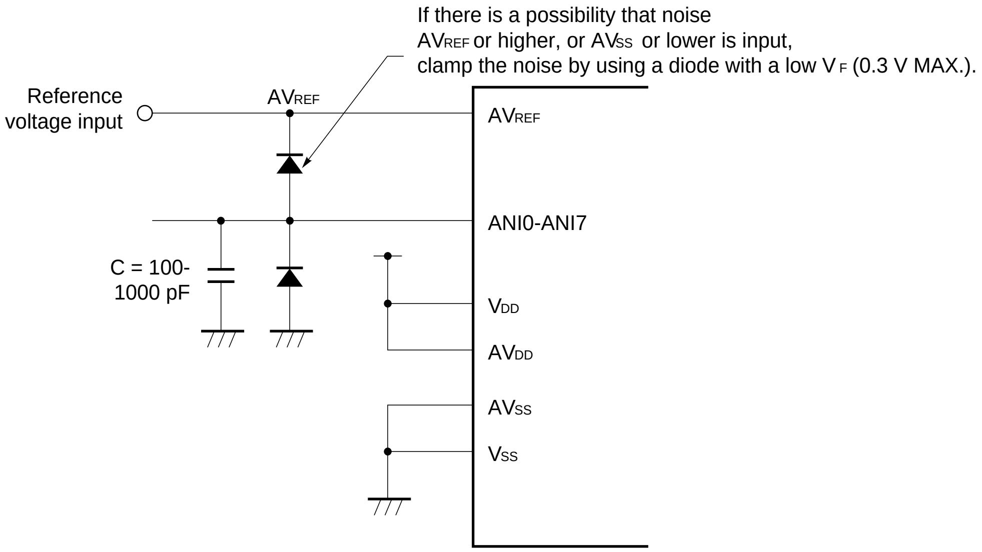
(2) ANI0-ANI7 input range

Observe the rated range of the ANI0-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-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 14-9 is recommended.

Figure 14-9. Processing Analog Input Pin


If there is a possibility that noise
AVREF or higher, or AVSS or lower is input,
clamp the noise by using a diode with a low VF (0.3 V MAX.).
Reference voltage input
AVREF
AVREF
ANI0-ANI7
C = 100-1000 pF
VDD
AVDD
AVSS
VSS

(4) ANI0/P10-ANI7/P17

The analog input pins (ANI0-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-ANI7 selected, do not execute the input instruction for 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.

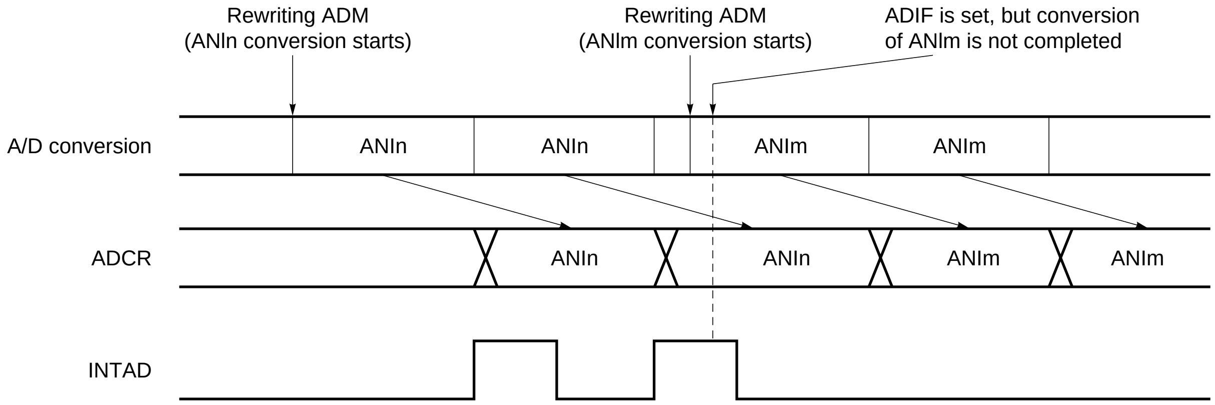
(6) Interrupt request flag (ADIF)

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 and interrupt request flags are set immediately before the contents of ADM are rewritten. Consequently, ADIF is 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 14-10**).

To resume A/D conversion that has been once stopped, clear ADIF before resuming the conversion.

Figure 14-10. A/D Conversion End Interrupt Generation Timing



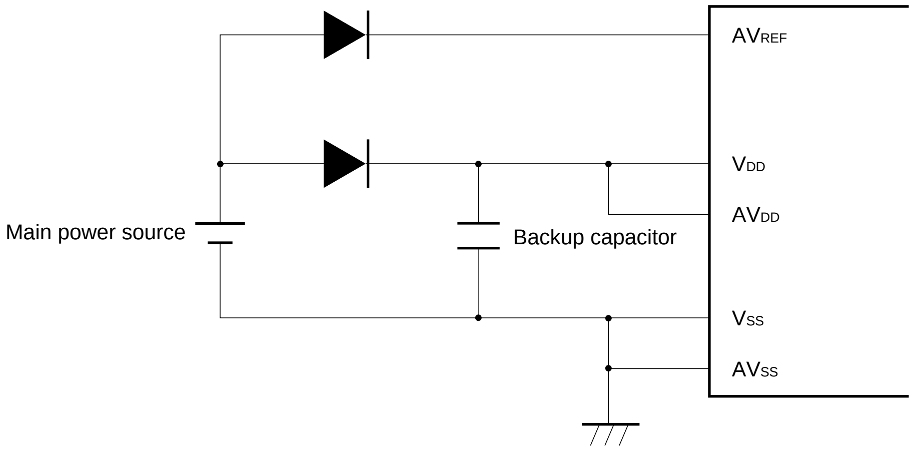
Remark n = 0, 1, ..., 7
m = 0, 1, ..., 7

(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-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 14-11.

Figure 14-11. Processing of AV$_{DD}$ Pin



CHAPTER 15 SERIAL INTERFACE CHANNEL 0 (μPD78018F SUBSERIES)

The μPD78018F 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 17 SERIAL INTERFACE CHANNEL 1**).

Table 15-1. 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 transmit/receive 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)
SBI (serial bus interface)		Available	Not available
2-wire serial I/O			

Note Can be set only when the main system clock oscillates at 4.19 MHz or less.

15.1 Functions of Serial Interface Channel 0

Serial interface channel 0 has the following four modes:

Table 15-2. 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 consumption can be reduced.	–
3-wire serial I/O mode	$\overline{\text{SCK0}}$ (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	$\overline{\text{SCK0}}$ (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 microcomputers 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.	
2-wire serial I/O mode	$\overline{\text{SCK0}}$ (serial clock), SB0 or SB1 (serial data bus)	• Any data transfer format can be supported by the program and lines for hand-shake conventionally necessary for connecting multiple devices can be eliminated.	

Caution Do not change the operation mode (3-wire serial I/O, 2-wire serial I/O, or SBI) while the operation of serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.

15.2 Configuration of Serial Interface Channel 0

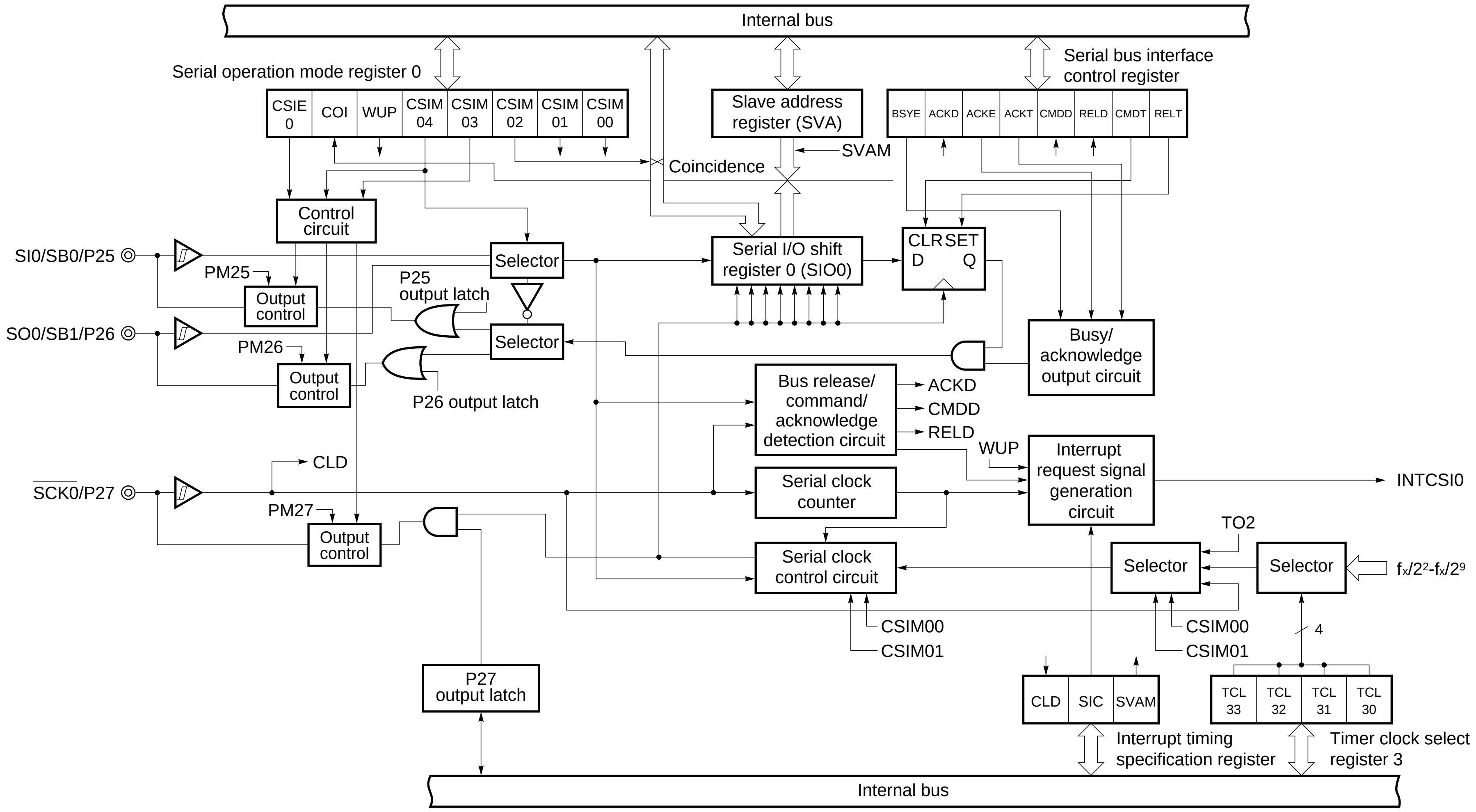
Serial interface channel 0 consists of the following hardware:

Table 15-3. 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]

Note Refer to **Figure 6-6 Block Diagram of P20, P21, and P23-P26 (μPD78018F Subseries)** and **Figure 6-7 Block Diagram of P22 and P27 (μPD78018F Subseries)**.

Figure 15-1. 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. It 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.

The high-order 7 bits of data with its LSB masked by setting the bit 4 (SVAM) of the interrupt timing specification register (SINT) can also compare the slave address.

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.

The wake-up function can be used by setting bit 5 (WUP) of CSIM0 to 1 in the SBI mode. 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 SVA.

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 transmit/receive 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.

15.3 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 15-2)

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 15-3)

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 Do not change the operation mode (3-wire serial I/O, 2-wire serial I/O, or SBI) while the operation of serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.

Figure 15-2. Format of Timer Clock Select Register 3

Symbol	7	6	5	4	3	2	1	0	Address	On 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 Before writing data other than that already written to TCL3, stop the serial transfer.

Remarks 1. f_X : Main system clock oscillation frequency

2. ( ) : At f_X = 10.0 MHz operation

Figure 15-3. Format of Serial Operation Mode Register 0

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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 0-3 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	0	Note 3 ×	Note 3 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open drain I/O)	$\overline{\text{SCK0}}$ (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)	$\overline{\text{SCK0}}$ (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

R

COI	Slave address comparison result flag[Note 5]
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 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).

5. COI is 0 when CSIE0 = 0.

Remark

×	: Don't care
PM××	: Port mode register
P××	: Output latch of 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 15-4. Format of Serial Bus Interface Control Register (1/2)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	On 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 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 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 the serial operation mode register 0 (CSIM0)

Phase-out/Discontinued

Figure 15-4. Format of Serial Bus Interface Control Register (2/2)

R/W	ACKT	Outputs acknowledge signal in synchronization with falling edge of clock of $\overline{\text{SCK0}}$ 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	
	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 $\overline{\text{SCK0}}$ (automatically outputs when ACKE = 1).
		After completion of transfer	Outputs acknowledge signal in synchronization with falling edge of clock of $\overline{\text{SCK0}}$ 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 $\overline{\text{SCK0}}$ immediately after busy mode is released after transfer start instruction has been executed • When CSIE0 = 0 • When $\overline{\text{RESET}}$ is input	• When acknowledge signal ($\overline{\text{ACK}}$) is detected at rising edge of clock of $\overline{\text{SCK0}}$ after completion of transfer

R/W	BSYE[Note]	Synchronous busy signal output control
	0	Disables output of busy signal in synchronization with falling edge of clock of $\overline{\text{SCK0}}$ immediately after instruction that clears this bit to 0 has been executed.
	1	Outputs busy signal from falling edge of clock of $\overline{\text{SCK0}}$ 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 the 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 15-5. Format of Interrupt Timing Specification Register

Symbol	7	<6>	<5>	<4>	3	2	1	0	Address	On 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 0-7
1	Bits 1-7

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 corresponding to INTCSI0

CSIE0 : Bit 7 of the serial operation mode register 0 (CSIM0)

15.4 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

15.4.1 Operation stop mode

Serial transfer is not executed in this mode. Consequently, the power consumption 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/$\overline{\text{SCK0}}$ 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 $\overline{\text{RESET}}$ signal is input.

The shaded portion in the figure indicates the bits used in the operation stop mode.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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

15.4.2 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 ($\overline{\text{SCK0}}$), 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 $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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 0-3 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 **15.4.3 Operation in SBI mode**.)											
1	1	2-wire serial I/O mode (Refer to **15.4.4 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 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	On 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 the serial operation mode register 0 (CSIM0)

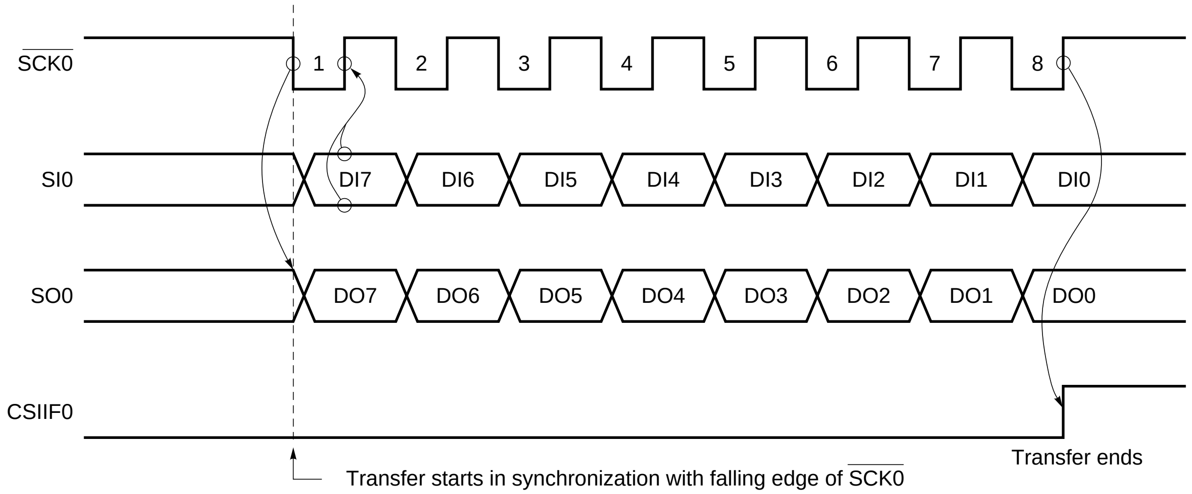
(2) Communication operation

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 ($\overline{\text{SCK0}}$). 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 $\overline{\text{SCK0}}$.

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 15-6. 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).

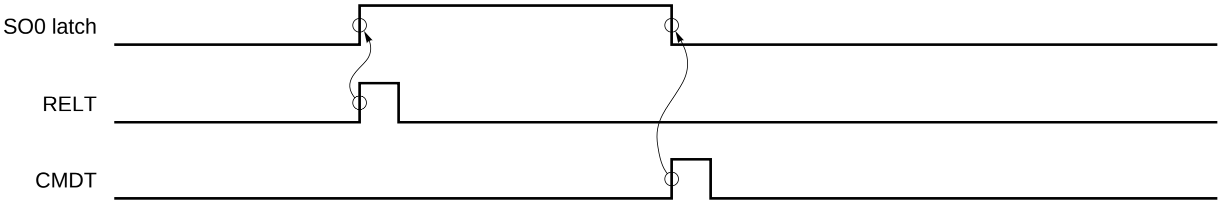
However, do not perform this manipulation during serial transfer.

The output level of the $\overline{\text{SCK0}}$ pin is controlled by manipulating the P27 output latch in the output mode (mode of the internal system clock) (refer to **15.4.5 Manipulating $\overline{\text{SCK0}}$/P27 pin output**).

(3) Signals

Figure 15-7 shows the operations of RELT and CMDT.

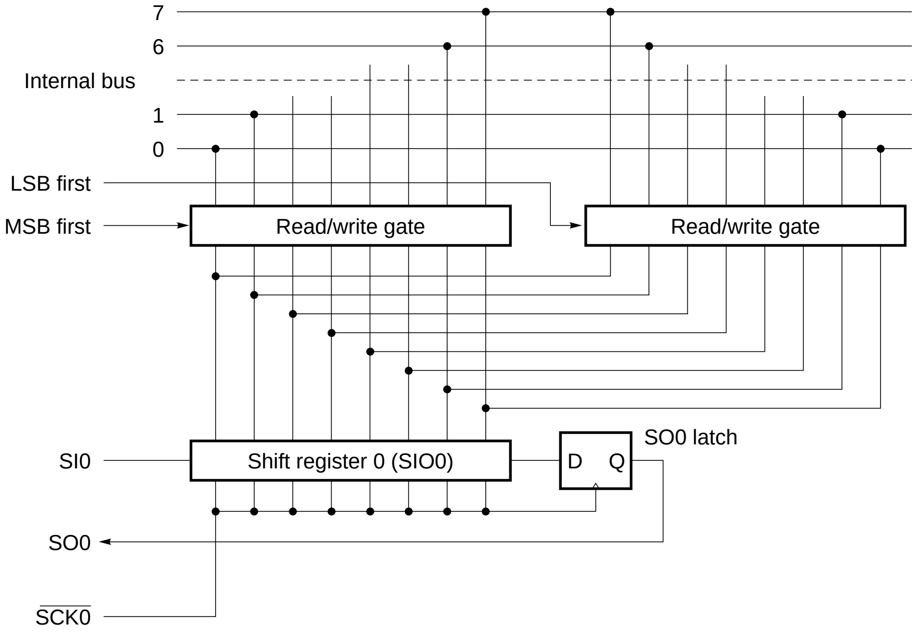
Figure 15-7. Operations of RELT and CMDT



(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 15-8 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 15-8. 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, specify whether the MSB or LSB is first 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 $\overline{\text{SCK0}}$ 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.

15.4.3 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 microcomputers 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 application program which controls the serial interface channel 0.

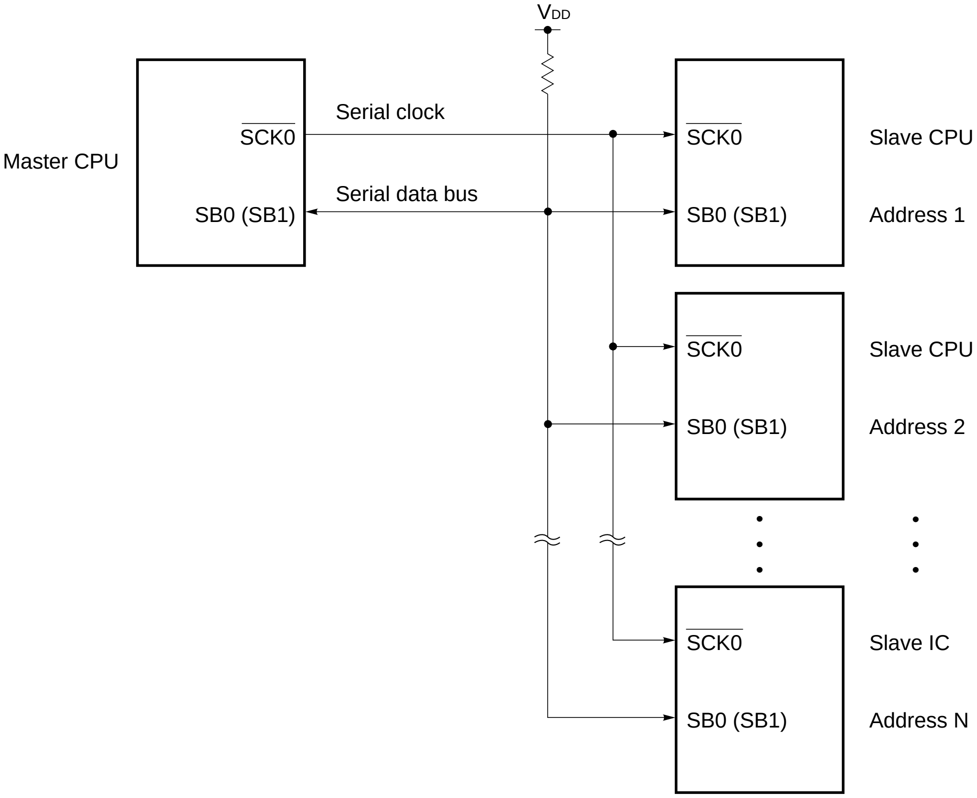
The SBI function is provided to some devices such as the 75X/XL series and the 78K series.

Figure 15-9 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 15-9. 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 $\overline{\text{SCK0}}$ and serial data bus SB0 (SB1). Therefore, the number of ports of the microcomputer 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 ($\overline{\text{ACK}}$) control function

An acknowledge signal is controlled to check reception of serial data.

(e) Busy signal ($\overline{\text{BUSY}}$) control function

A busy signal that indicates the busy status of the slave is controlled.

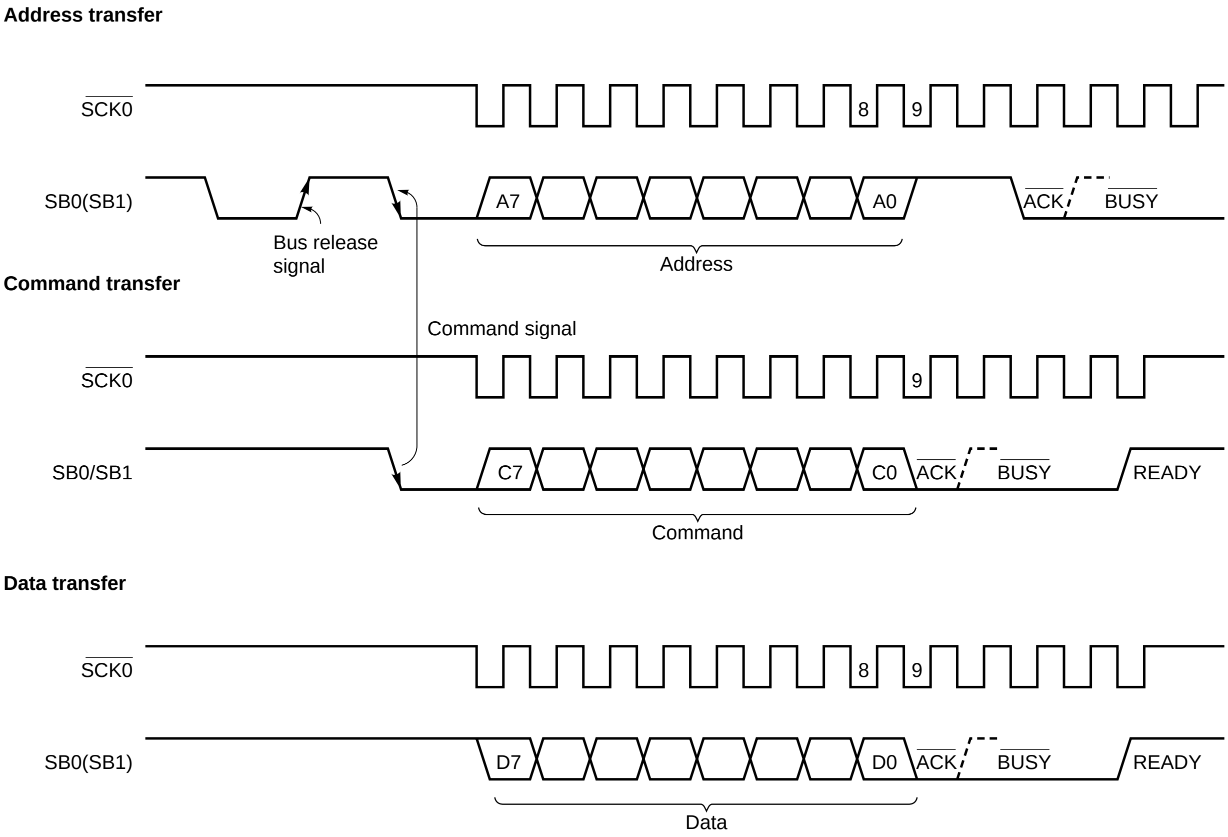
(2) Definition of SBI

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 15-10 shows the transfer timing of the address, command, and data.

Figure 15-10. SBI Transfer Timing



Remark The dotted line indicates the READY status.

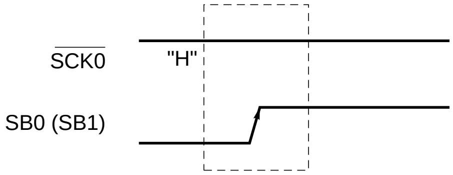
The bus release signal and command signal are output by the master. $\overline{\text{BUSY}}$ is output by the slave. $\overline{\text{ACK}}$ can be output by both the master and slave (usually, this signal is output by the 8-bit data reception side). The master continues outputting the serial clock from 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 $\overline{\text{SCK0}}$ line is high (when the serial clock is not output).

This signal is output by the master.

Figure 15-11. Bus Release Signal



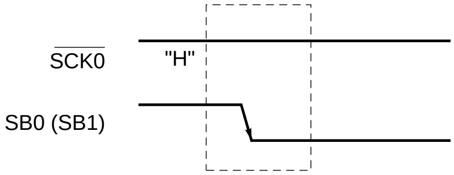
The bus release signal indicates that the master is to transmit an address to the slave. The slave is provided with hardware that detects the bus release signal.

★ **Caution A positive transition of the SB0 (SB1) pin from low to high is recognized as a bus release signal when the $\overline{\text{SCK0}}$ line is high. If the change timing of the bus is shifted due to the influence of the board capacitance, data that is transmitted may be identified as bus release signal by mistake. Exercise care in 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 $\overline{\text{SCK0}}$ line is high (when the serial clock is not output). This signal is output by the master.

Figure 15-12. Command Signal



The command signal indicates that the master is going to transmit a command to the slave (however, the command signal following the bus release signal indicates that an address is to be transmitted).

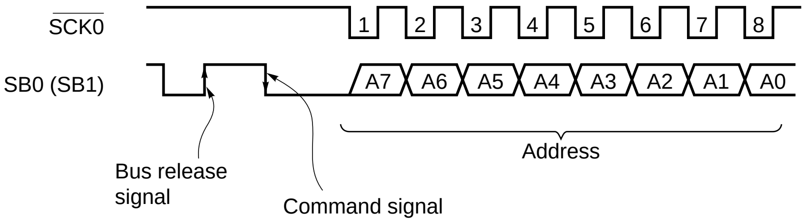
The slave is provided with hardware that detects the command signal.

★ **Caution A positive transition of the SB0 (SB1) pin from high to low is recognized as a command signal when the $\overline{\text{SCK0}}$ line is high. If the change timing of the bus is shifted due to the influence of the board capacitance, data that is transmitted may be identified as command signal by mistake. Exercise care in wiring.**

(c) Address

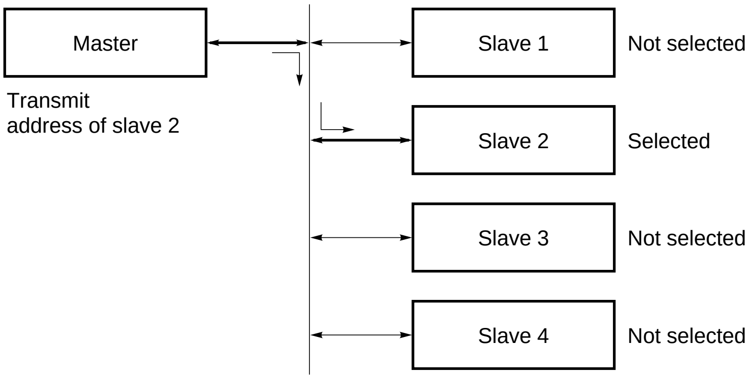
An address is an 8-bit data which the master outputs to the slaves connected to the bus lines in order to select a specific slave.

Figure 15-13. 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 15-14. 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 15-15. Command

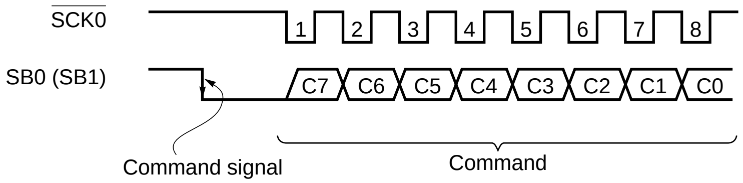
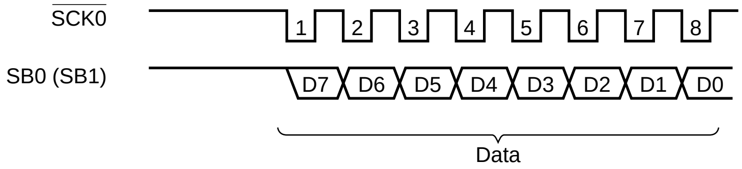


Figure 15-16. Data

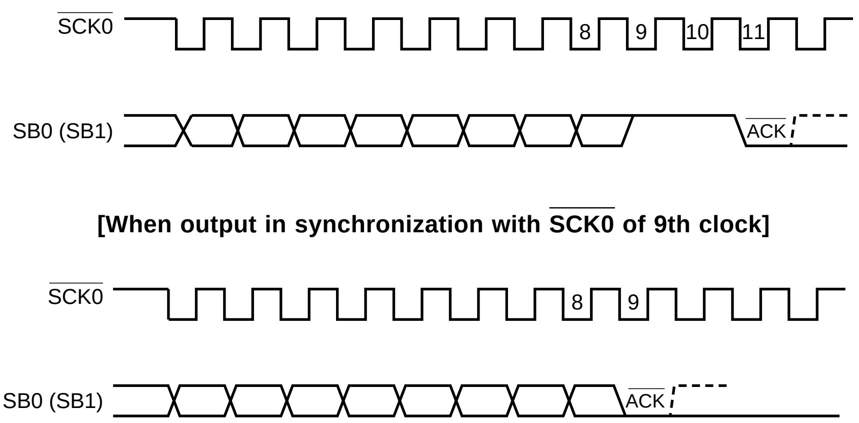


The 8-bit data following the command signal is defined as a command. The 8-bit data that does not follow the command signal is defined as data. The method of using the command and data can be arbitrarily determined by the communication specifications.

(e) Acknowledge signal ($\overline{\text{ACK}}$)

The acknowledge signal is used for confirmation of reception of serial data between the transmission and reception sides.

Figure 15-17. Acknowledge Signal



[When output in synchronization with $\overline{\text{SCK0}}$ of 11th clock]

[When output in synchronization with $\overline{\text{SCK0}}$ of 9th clock]

Remark The dotted line indicates the READY status.

The acknowledge signal is a one-shot pulse synchronized with the falling edge of $\overline{\text{SCK0}}$ after 8-bit data has been transferred. Its position is arbitrary, and may be synchronized with $\overline{\text{SCK0}}$ of clock n.

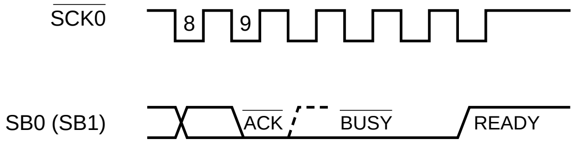
After the transmission side has transferred 8-bit data, it checks whether the reception side has returned an acknowledge signal. If no acknowledge signal is returned within a specific time after data transmission, it is judged that the data was not received correctly.

(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 15-18. Busy Signal and Ready Signal



Remark The dotted 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 SBI outputs the $\overline{\text{BUSY}}$ signal after the $\overline{\text{BUSY}}$ has been cleared and until the next serial clock falls. If WUP is set to 1 by mistake during this period, $\overline{\text{BUSY}}$ will not be cleared. To set WUP to 1, therefore, clear $\overline{\text{BUSY}}$, and make sure that the SB0 (SB1) pin has gone high.

(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	On 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 0-3 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	×	3-wire serial I/O mode (Refer to **15.4.2 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)	$\overline{\text{SCK0}}$ (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 **15.4.4 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
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.

Remark × : Don’t care
PM×× : Port mode register
P×× : Output latch of 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	On 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 . Atter 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 the 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 $\overline{\text{SCK0}}$ (automatically output when ACKE = 1).
	After completion of transfer	Outputs acknowledge signal in synchronization with falling edge of clock of $\overline{\text{SCK0}}$ 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 $\overline{\text{SCK0}}$ immediately after busy mode is released after transfer start instruction has been executed • When CSIE0 = 0 • When $\overline{\text{RESET}}$ signal is input	• When acknowledge signal ($\overline{\text{ACK}}$) is detected at rising edge of clock of $\overline{\text{SCK0}}$ after completion of transfer

R/W

BSYE Note	Controls synchronous busy signal output
0	Disables output of busy signal in synchronization with falling edge of clock of $\overline{\text{SCK0}}$ immediately after instruction that clears this bit to 0 has been executed.
1	Outputs busy signal from falling edge of clock of $\overline{\text{SCK0}}$ 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 the 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	On 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 0-7
1	Bits 1-7

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 corresponding to INTCSI0

CSIE0 : Bit 7 of the serial operation mode register 0 (CSIM0)

(4) Signals

Figures 15-19 through 15-24 show the signals of serial bus interface control register (SBIC) and the operations of the flags of SBIC. Table 15-4 lists the signals of SBI.

Figure 15-19. Operations of RELT, CMDT, RELD, and CMDD (Master)

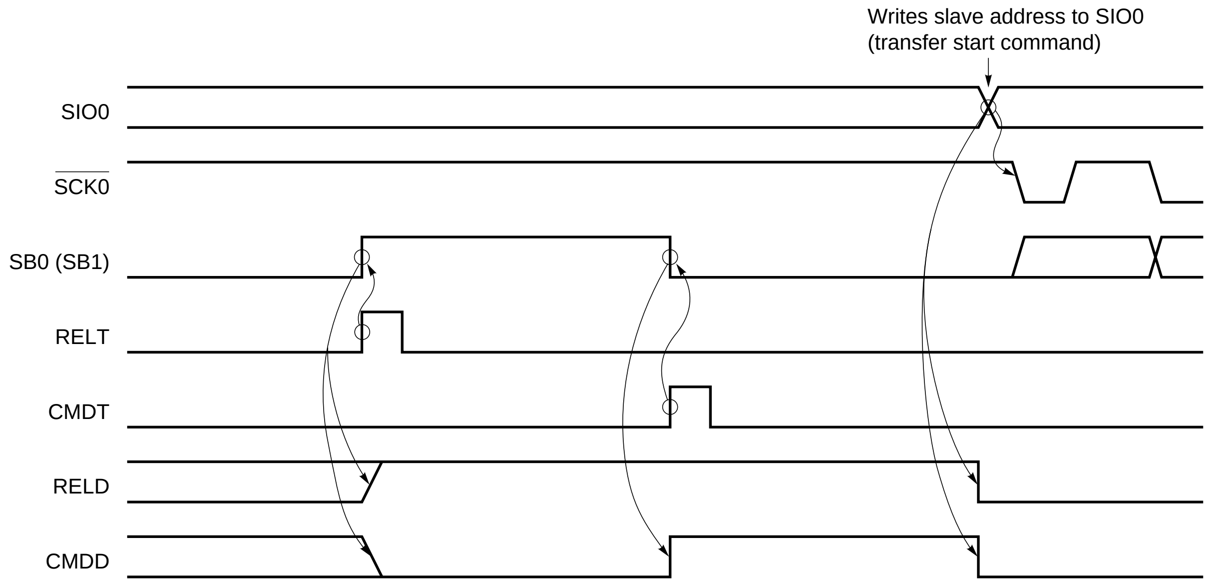


Figure 15-20. Operations of RELD and CMDD (Slave)

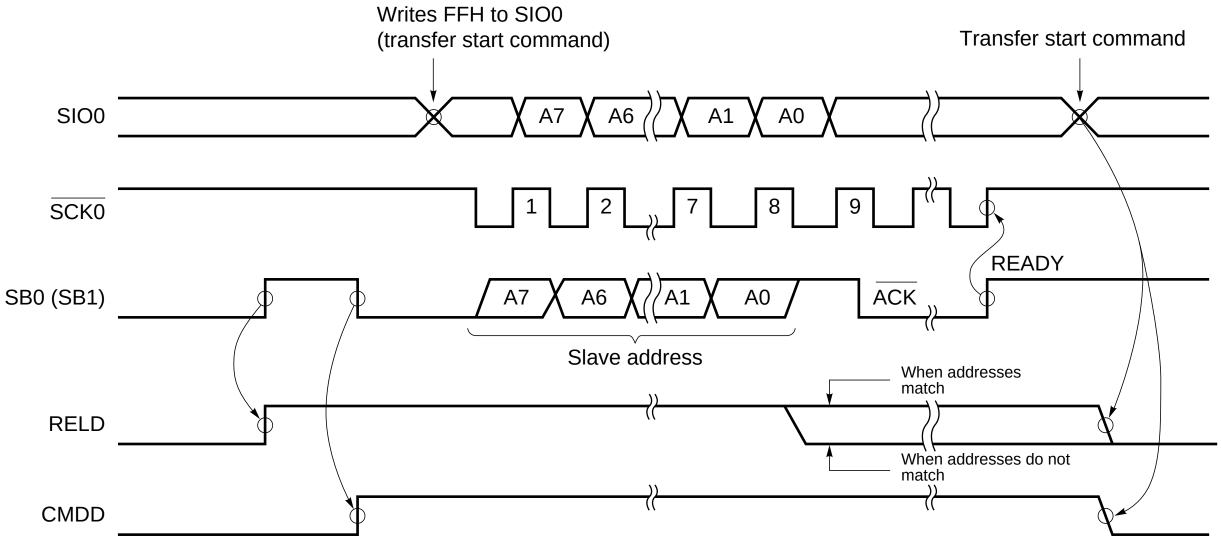
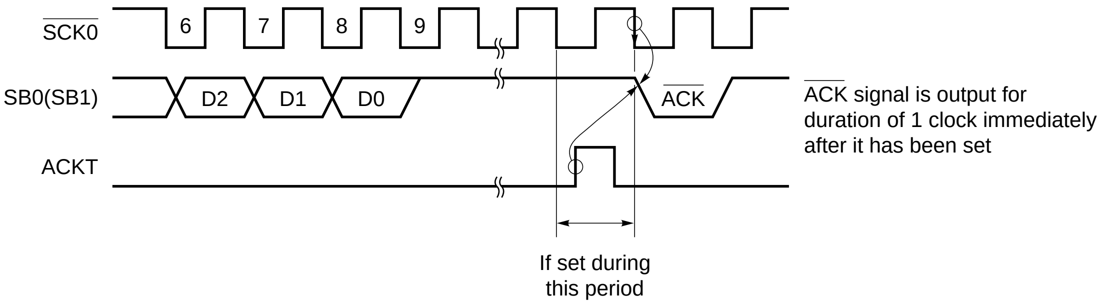
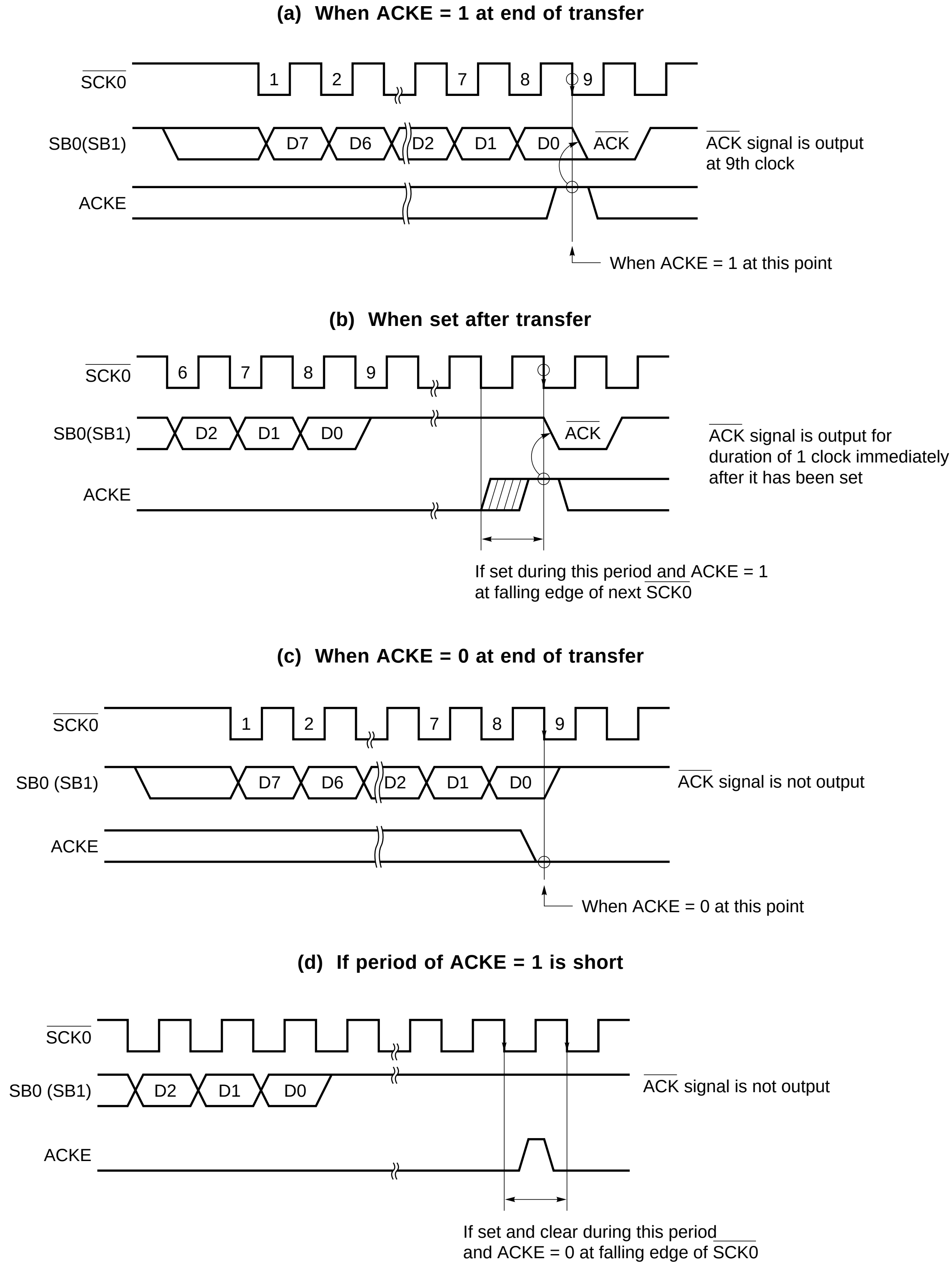


Figure 15-21. Operation of ACKT


SCK0
SB0(SB1)
ACKT
6
7
8
9
D2
D1
D0
ACK
ACK signal is output for duration of 1 clock immediately after it has been set
If set during this period

Caution Do not set ACKT before end of transfer.

Figure 15-22. Operation of ACKE



(a) When ACKE = 1 at end of transfer

ACK signal is output at 9th clock

When ACKE = 1 at this point

(b) When set after transfer

ACK signal is output for duration of 1 clock immediately after it has been set

If set during this period and ACKE = 1 at falling edge of next SCK0

(c) When ACKE = 0 at end of transfer

ACK signal is not output

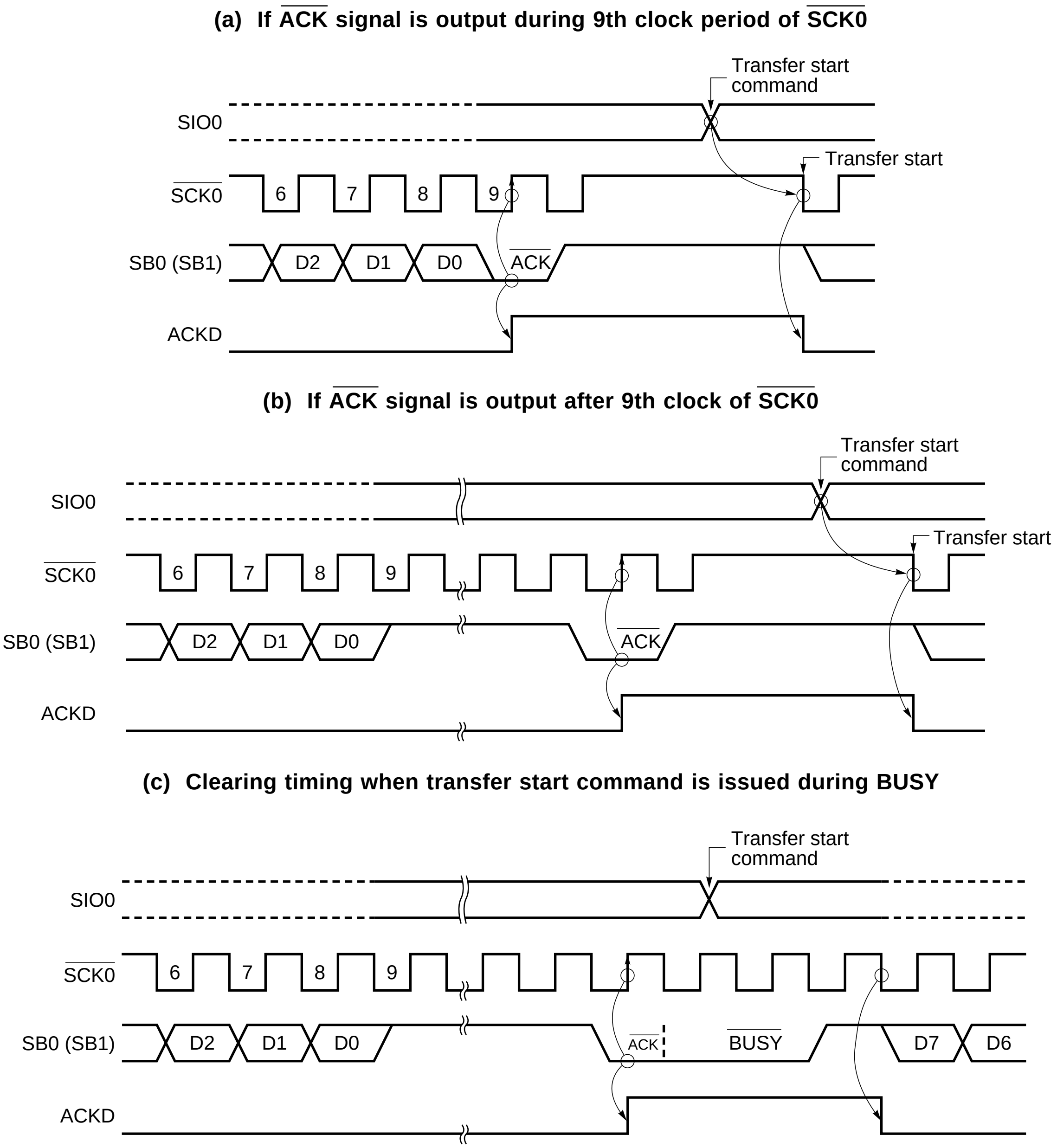
When ACKE = 0 at this point

(d) If period of ACKE = 1 is short

ACK signal is not output

If set and clear during this period and ACKE = 0 at falling edge of SCK0

Figure 15-23. Operation of ACKD

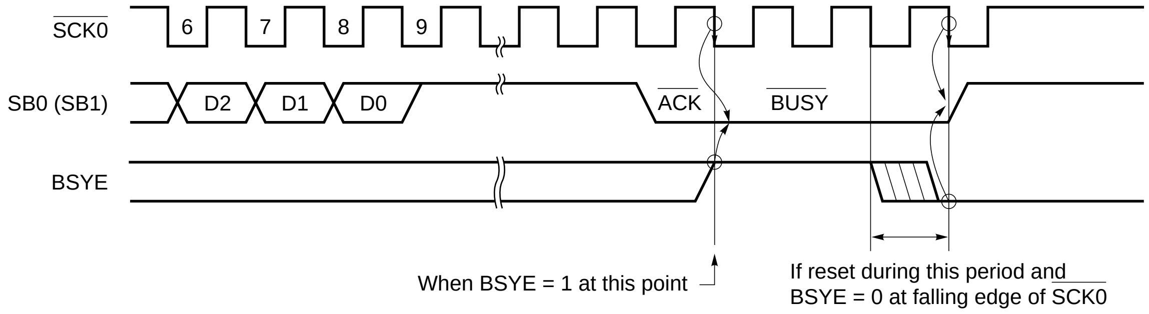


(a) If $\overline{\text{ACK}}$ signal is output during 9th clock period of $\overline{\text{SCK0}}$

(b) If $\overline{\text{ACK}}$ signal is output after 9th clock of $\overline{\text{SCK0}}$

(c) Clearing timing when transfer start command is issued during BUSY

Figure 15-24. Operation of BSYE



$\overline{\text{SCK0}}$
SB0 (SB1)
BSYE
6
7
8
9
D2
D1
D0
$\overline{\text{ACK}}$
$\overline{\text{BUSY}}$
When BSYE = 1 at this point
If reset during this period and
BSYE = 0 at falling edge of $\overline{\text{SCK0}}$

Table 15-4. Signals in SBI Mode (1/2)

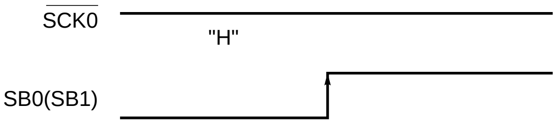
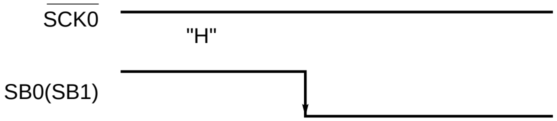
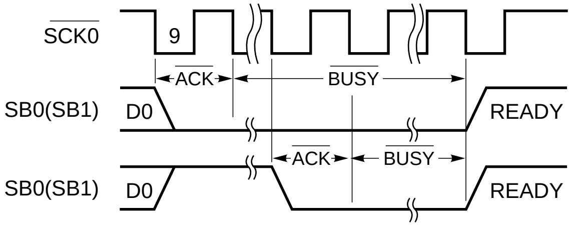
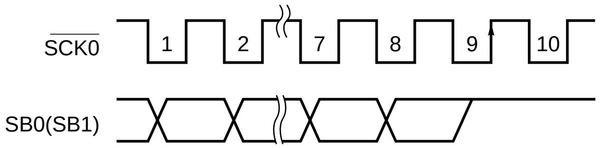
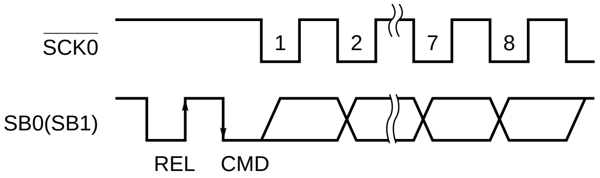
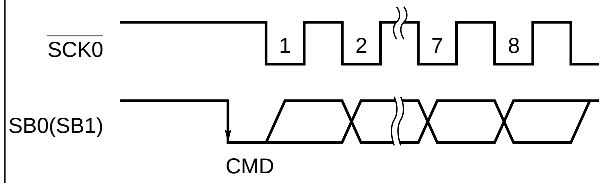
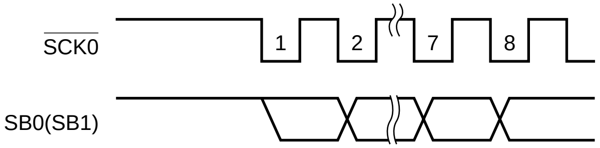
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Influence on Flag	Meaning of Signal
Bus release signal (REL)	Master	At rising edge of SB0 (SB1) when $\overline{\text{SCK0}}$ = 1		• Setting of RELT	• RELD is set. • CMDD is cleared.	Subsequently outputs CMD signal and indicates that transmit data is address
Command signal (CMD)	Master	At falling edge of SB0 (SB1) when $\overline{\text{SCK0}}$ = 1		• 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 ($\overline{\text{ACK}}$)	Master/ slave	Low-level signal output to SB0 (SB1) for duration of 1 clock of $\overline{\text{SCK0}}$ after serial reception has been completed	[Synchronous busy output]  $\overline{\text{SCK0}}$ 9 $\overline{\text{ACK}}$ $\overline{\text{BUSY}}$ SB0(SB1) D0 READY $\overline{\text{ACK}}$ $\overline{\text{BUSY}}$ SB0(SB1) D0 READY	<1> ACKE = 1 <2> Setting of ACKT	• ACKD is set	Reception completed
Busy signal ($\overline{\text{BUSY}}$)	Slave	[Synchronous busy signal] Low-level signal output to SB0 (SB1) following acknowledge signal		• 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		<1> BSYE = 0 <2> Instruction excution that writes data to SIO0 (transfer start command)	—	Serial reception enabled status

Table 15-4. Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Influence on Flag	Meaning of Signal
Serial clock ($\overline{\text{SCK0}}$)	Master	Synchronous clock for output of address/command/data, $\overline{\text{ACK}}$ signal, and synchronous $\overline{\text{BUSY}}$ signal. Address/command/data is transferred when first eight of this signal are output.	 $\overline{\text{SCK0}}$ SB0(SB1) 1 2 7 8 9 10	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 $\overline{\text{SCK0}}$)[Note 1]	Timing of signal output to serial data bus
Address (A7-A0)	Master	8-bit data transferred in synchronization with $\overline{\text{SCK0}}$ after REL and CMD signals have been output	 $\overline{\text{SCK0}}$ SB0(SB1) 1 2 7 8 REL CMD			Address value of slave device on serial bus
Command (C7-C0)	Master	8-bit data transferred in synchronization with $\overline{\text{SCK0}}$ after only CMD signal is output (REL signal is not output)	 $\overline{\text{SCK0}}$ SB0(SB1) 1 2 7 8 CMD			Command or message to slave device
Data (D7-D0)	Master/ slave	8-bit data transferred in synchronization with $\overline{\text{SCK0}}$ 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 $\overline{\text{SCK0}}$.

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 $\overline{\text{BUSY}}$ status, transfer is not started until the READY status is set.

(5) Pin configuration

The configuration of the serial clock pin $\overline{\text{SCK0}}$ and serial data bus pin SB0 (SB1) is as follows:

(a) $\overline{\text{SCK0}}$ .... Pin that inputs/outputs serial clock

<1> Master .... CMOS, push-pull output

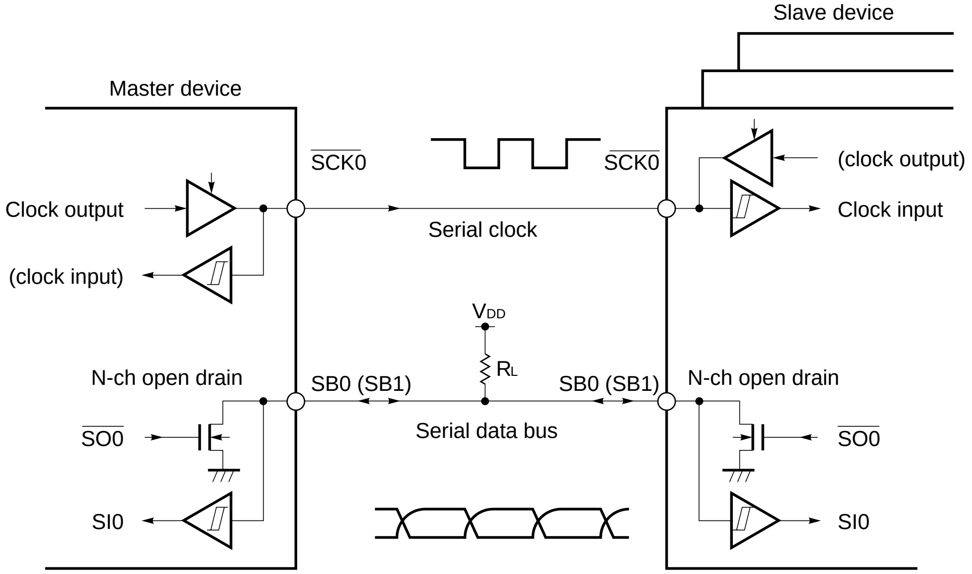
<2> Slave ...... Schmitt input

(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 15-25. Pin Configuration



Caution Because it is necessary to make the N-ch open drain high-impedance state when data is received, write FFH to serial I/O shift register 0 (SIO0) in advance. It can always be turned off during transfer. However, when the wake-up function specification bit (WUP) = 1, the N-ch open-drain output is always at 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 bit (WUP) 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 15-26 through 15-29 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 15-26. Address Transmit Operation from Master Device to Slave Device (WUP = 1)

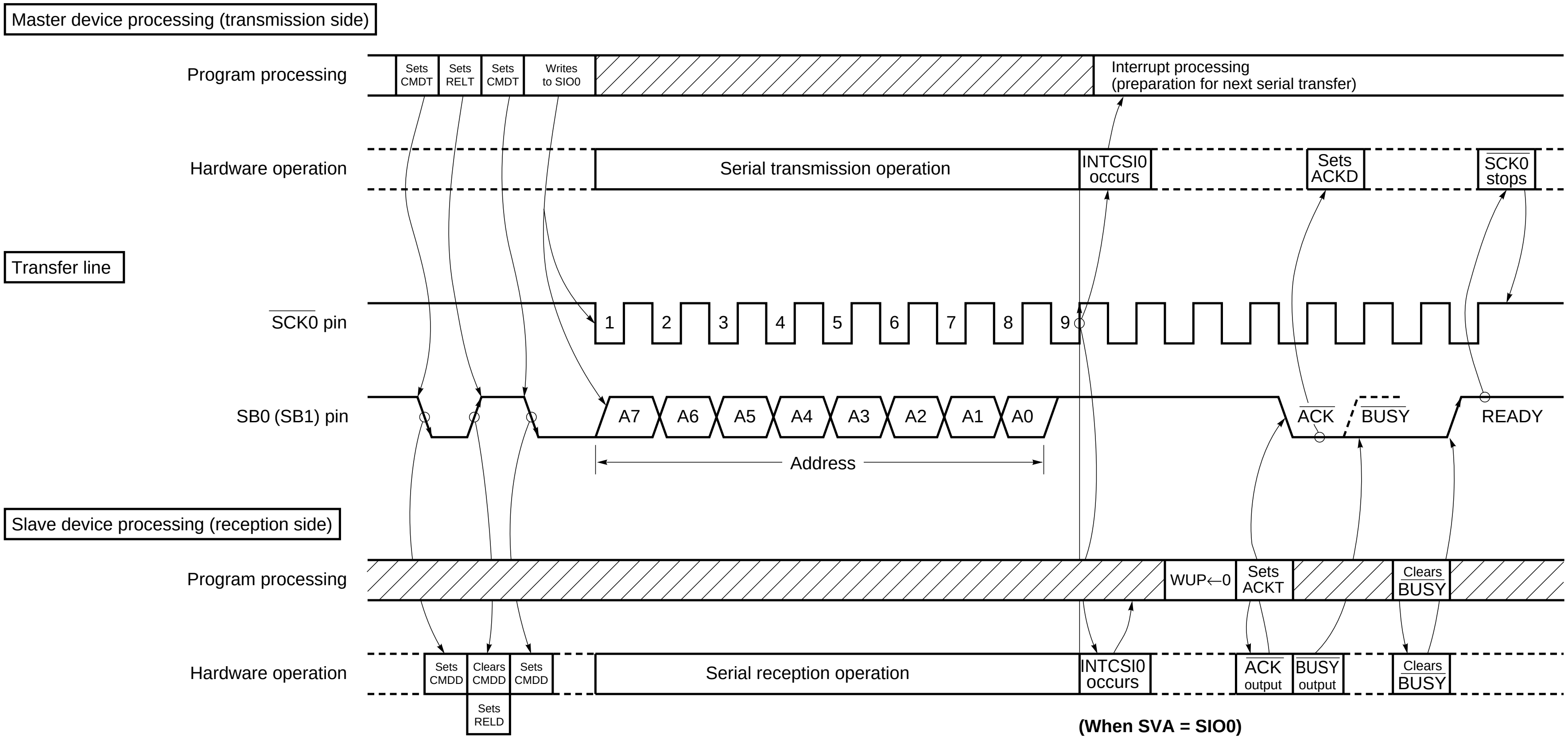


Figure 15-27. Command Transmit Operation from Master Device to Slave Device

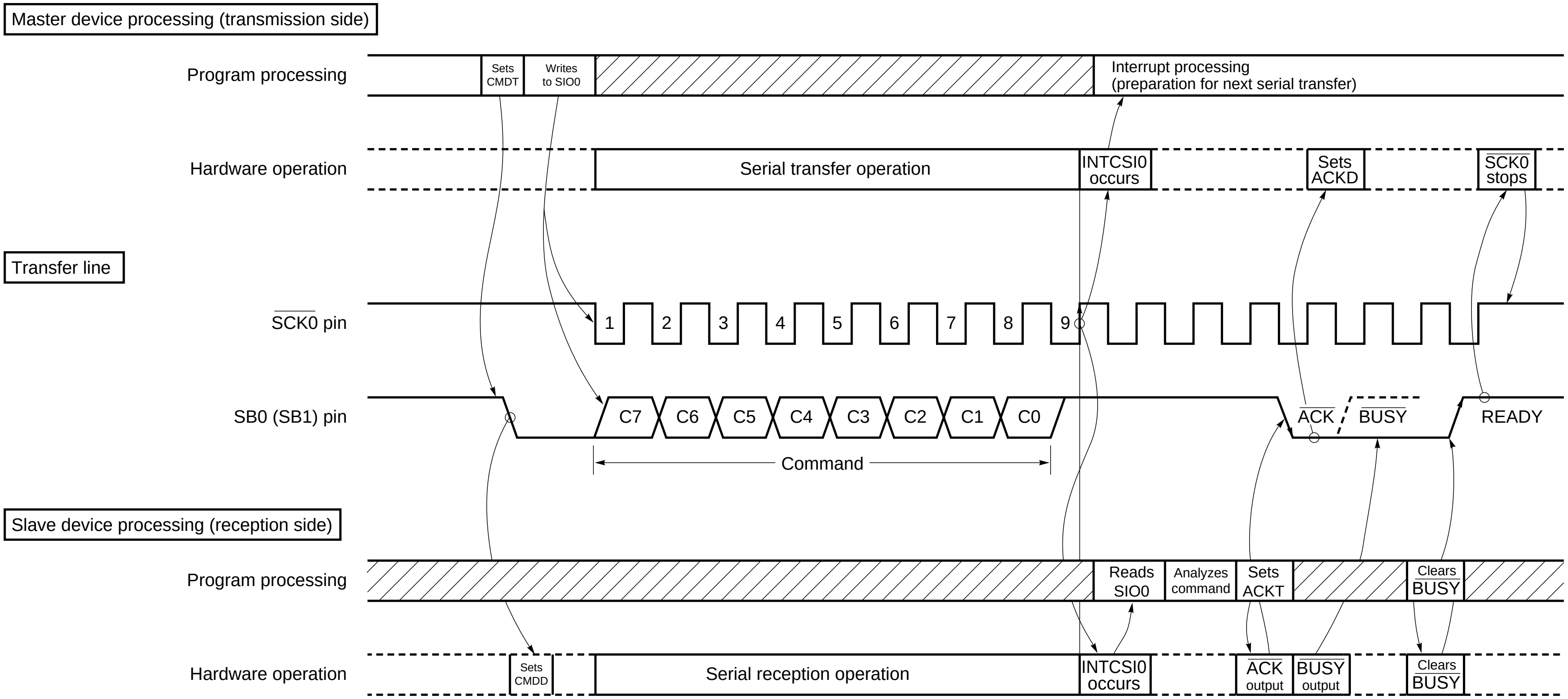


Figure 15-28. Data Transmit Operation from Master Device to Slave Device

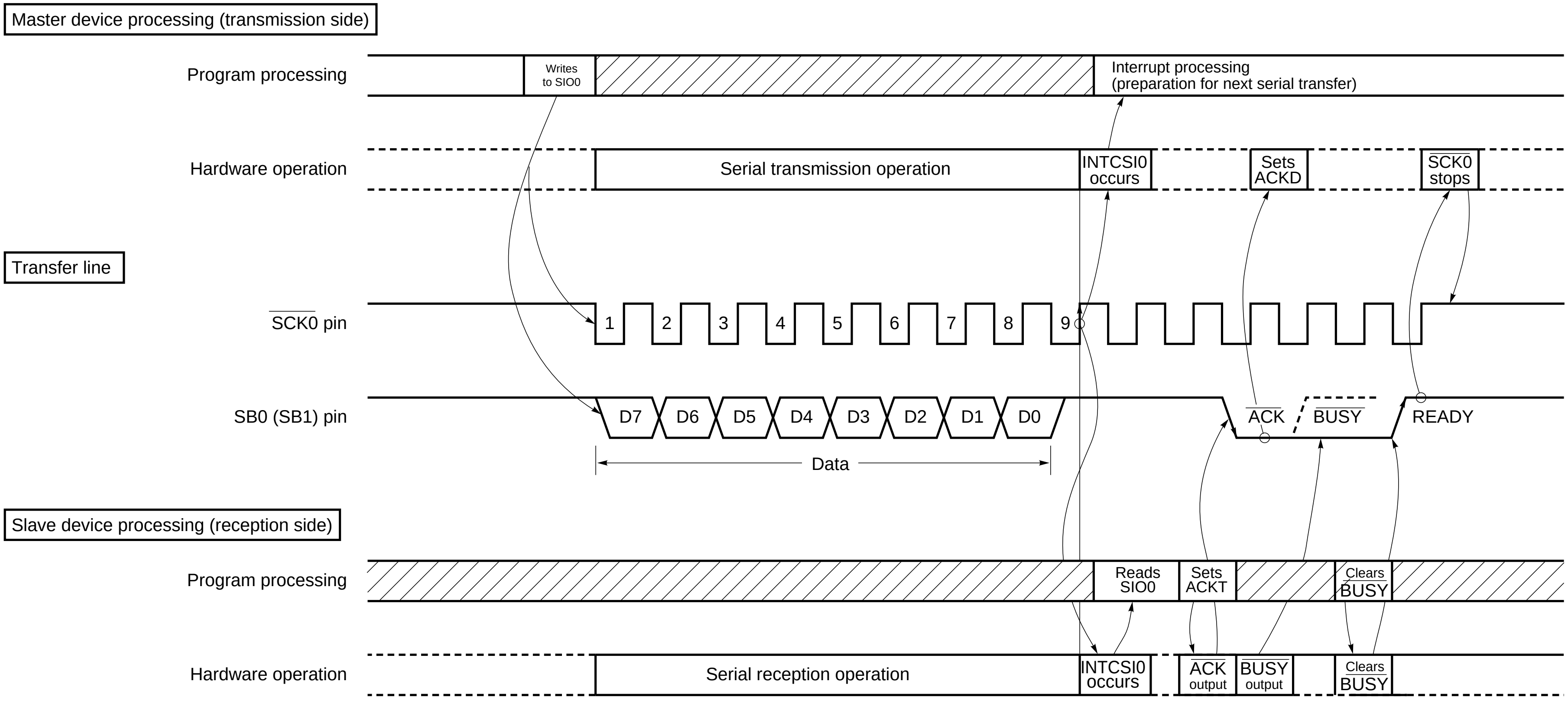
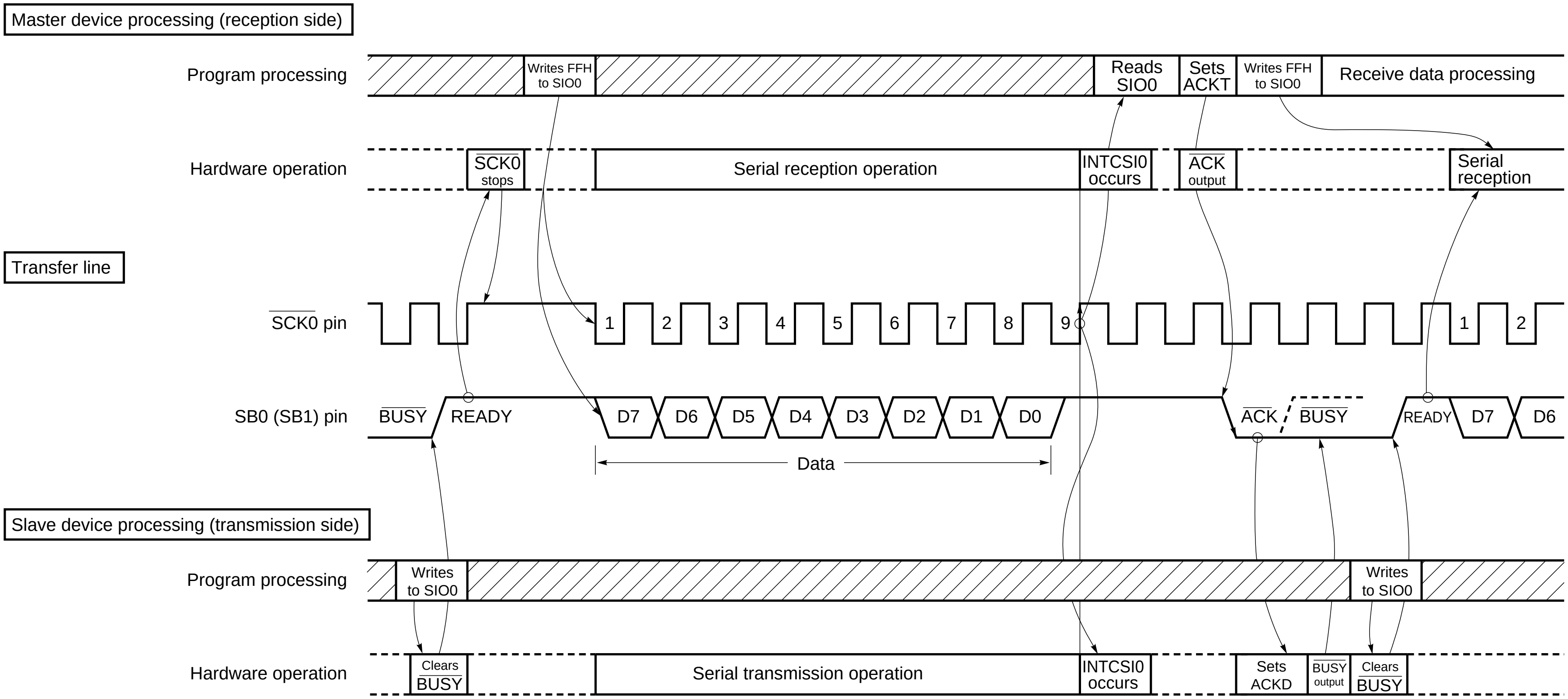


Figure 15-29. Data Transmit Operation from Slave Device to Master Device



(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 $\overline{\text{SCK0}}$ 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 output must be made 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 output is always at 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 that is used to input/output data (SB0 or SB1) 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.

(10) Method to judge busy state 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 ($\overline{\text{ACK}}$) 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 (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 = 1 during this period, $\overline{\text{BUSY}}$ cannot be released. Therefore, to set WUP to 1, be sure to release the $\overline{\text{BUSY}}$ status, and make sure that the SB0 (SB1) pin has gone high.

(d) Be sure to set the pin used to input or output data after the $\overline{\text{RESET}}$ signal has been input and before serial transfer of the first byte.

<1> Set 1 to the output latch 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 latch of P25 and P26 to which 1 has been set.

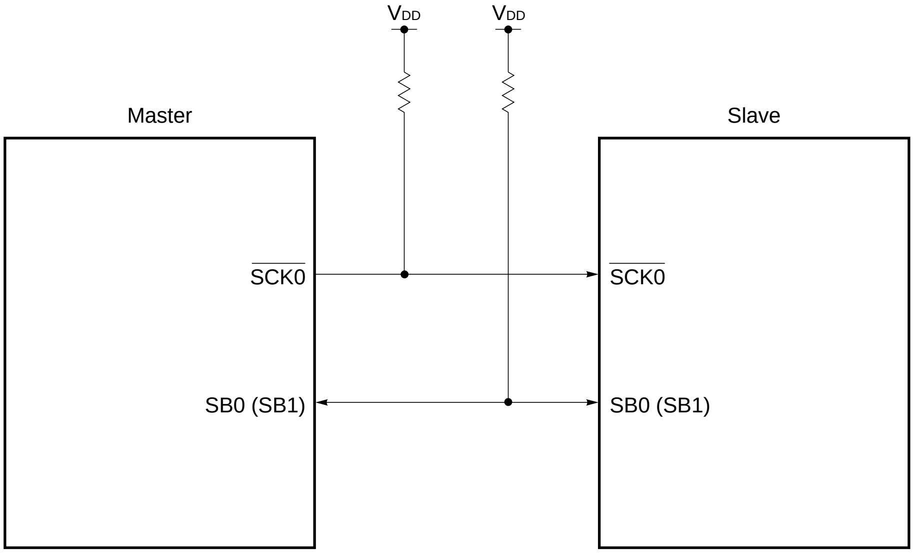
★ (e) A positive transition of the SB0 (SB1) pin from low to high or high to low is recognized as a bus release signal or a command signal when the $\overline{\text{SCK0}}$ line is high. If the change timing of the bus is shifted due to the influence of the board capacitance, data that is transmitted may be identified as bus release signal (or a command signal) by mistake. Exercise care in wiring.

15.4.4 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 ($\overline{\text{SCK0}}$) and serial data I/O (SB0 or SB1), are used to establish communication in this mode.

Figure 15-30. 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 $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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 0-3 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 **15.4.2 Operation in 3-wire serial I/O mode**.)											
1	0	SBI mode (Refer to **15.4.3 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

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 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	On 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 the 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	On 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 corresponding to INTCSI0

CSIE0 : Bit 7 of the serial operation mode register 0 (CSIM0)

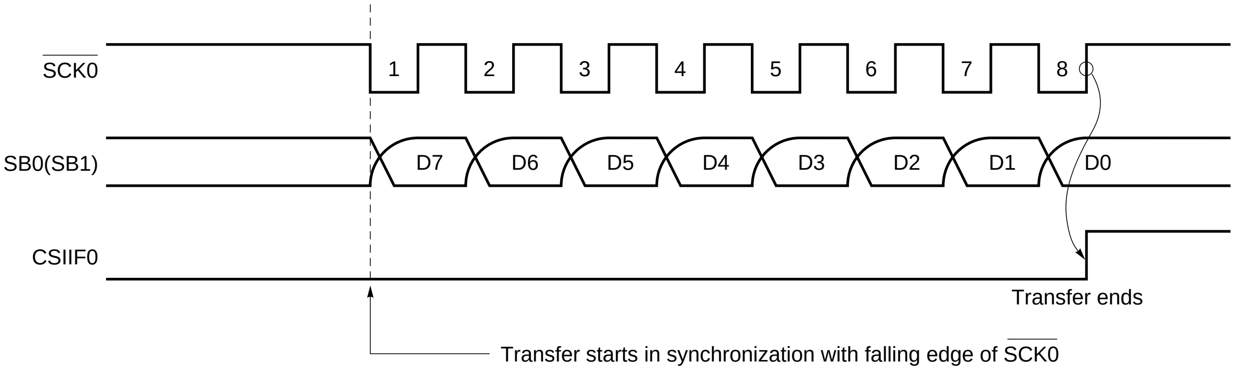
(2) **Communication operation**

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 ($\overline{\text{SCK0}}$). 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 the SIO0 at the rising edge of $\overline{\text{SCK0}}$.

When the 8-bit data has been completely transferred, the operation of the SIO0 is automatically stopped, and an interrupt request flag (CSIIF0) is set.

Figure 15-31. Timing of 2-Wire Serial I/O Mode


SCK0
SB0(SB1)
CSIIF0
1
2
3
4
5
6
7
8
D7
D6
D5
D4
D3
D2
D1
D0
Transfer ends
Transfer starts in synchronization with falling edge of SCK0

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 open-drain output must be made high-impedance state.

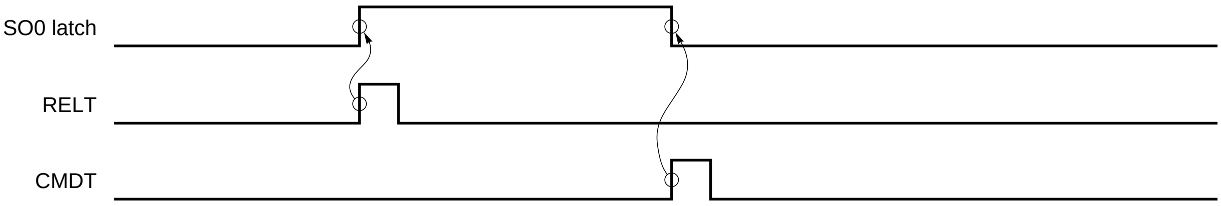
Because the SB0 (or SB1) pin outputs the status of the SO0 latch, the output status of the SB0 (or SB1) pin can be manipulated by setting the bit 0 (RELT) and bit 1 (CMDT) of serial bus interface control register (SBIC). However, do not manipulate the output status of the pin during serial transfer.

The output level of the $\overline{\text{SCK0}}$ pin is controlled by manipulating the P27 output latch in the output mode (mode of the internal system clock) (refer to **15.4.5 Manipulating $\overline{\text{SCK0}}$/P27 pin output**).

(3) Signals

Figure 15-32 shows the operations of RELT and CMDT.

Figure 15-32. 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 $\overline{\text{SCK0}}$ 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 output must be made high-impedance state 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.

15.4.5 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.

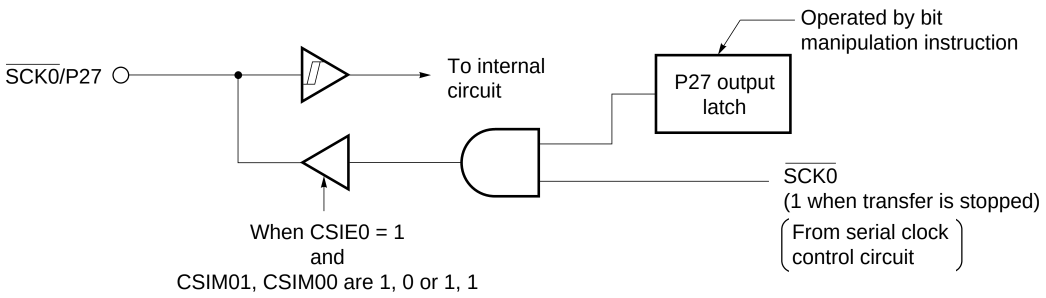
By manipulating the P27 output latch, the value of $\overline{\text{SCK0}}$ can be arbitrarily set by software (the SI0/SB0 and SO0/SB1 pins are controlled by the RELT and CMDT bits of serial bus interface control register (SBIC)).

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 15-33. Configuration of $\overline{\text{SCK0}}$/P27 Pin


$\overline{\text{SCK0}}$/P27
To internal circuit
Operated by bit manipulation instruction
P27 output latch
$\overline{\text{SCK0}}$
(1 when transfer is stopped)
From serial clock control circuit
When CSIE0 = 1
and
CSIM01, CSIM00 are 1, 0 or 1, 1

CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD78018FY SUBSERIES)

The μPD78018FY 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 17 SERIAL INTERFACE CHANNEL 1**).

Table 16-1. 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 transmit/receive 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)
2-wire serial I/O		Available	Not available
I^{2}C bus (Inter IC Bus)			

Note Can be set only when the main system clock oscillates at 4.19 MHz or less.

16.1 Functions of Serial Interface Channel 0

Serial interface channel 0 has the following four modes:

Table 16-2. 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 consumption can be reduced	–
3-wire serial I/O mode	$\overline{\text{SCK0}}$ (serial clock), SO0 (serial output), SI0 (serial input)	• Short data transfer processing time because independent input and output lines are used, allowing simultaneous tranmission 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
2-wire serial I/O mode	$\overline{\text{SCK0}}$ (serial clock), SB0 or SB1 (serial data bus)	• Any data transfer format can be supported by the program and lines for lhand-shake conventionally necessary for connecting multiple devices can be eliminated	
I^{2}C bus mode	SCL (serial clock), SDA0 or SDA1 (serial data bus)	• Supports I^{2}C bus format • 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 micro-computers are connected.	Useful for connecting CPU and peripheral IC containing serial interface conforming to I^{2}C bus

Caution Do not change the operation mode (3-wire serial I/O, 2-wire serial I/O, or I^{2}C bus) while the operation of serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.

16.2 Configuration of Serial Interface Channel 0

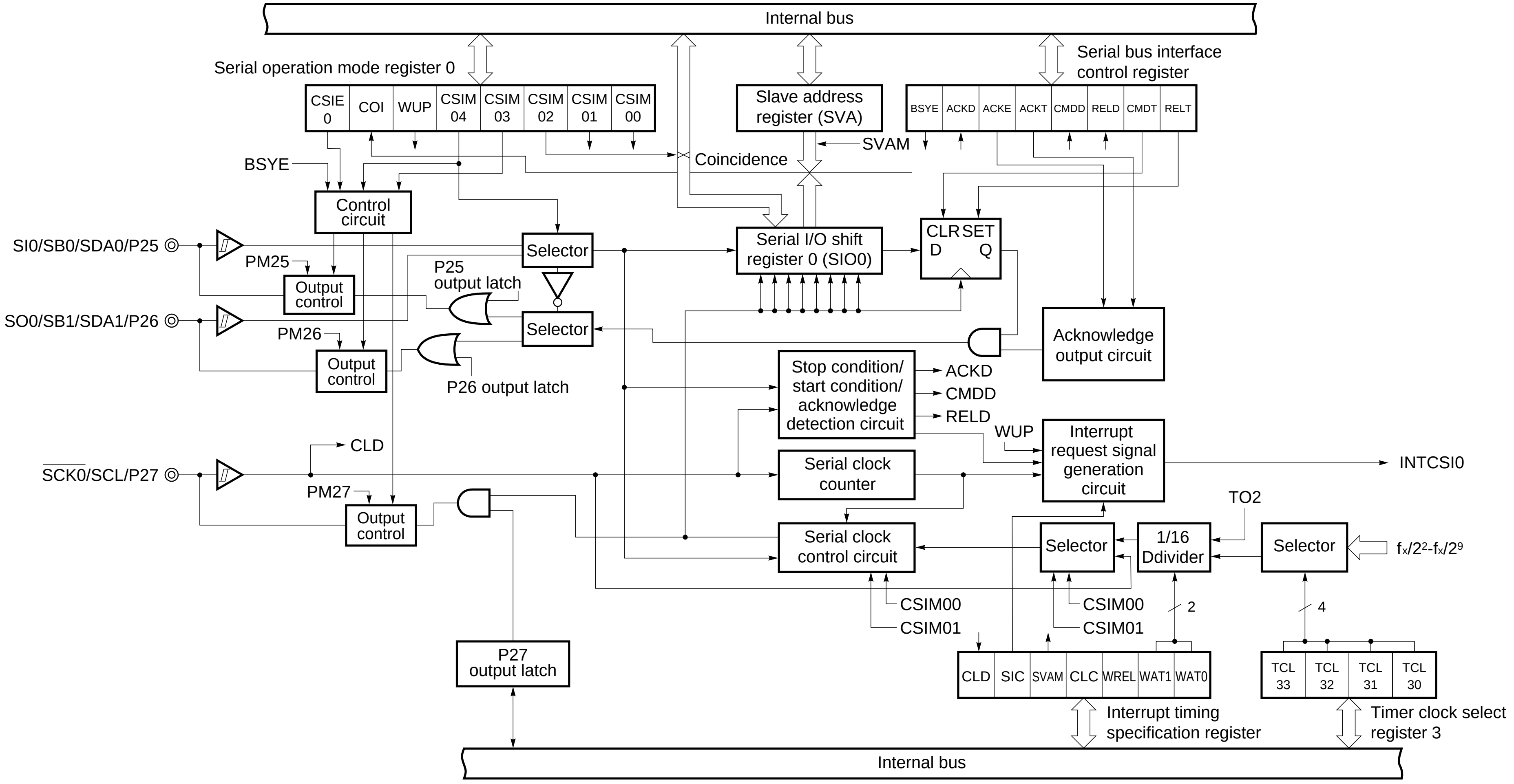
Serial interface channel 0 consists of the following hardware:

Table 16-3. 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)$^{\textbf{Note}}$

Note Refer to **Figure 6-8 Block Diagram of P20, P21, and P23-26 (μPD78018FY Subseries)**, and **Figure 6-9 Block Diagram of P22 and P27 (μPD78018FY Subseries)**.

Figure 16-1. 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 2-wire serial I/O mode and I^{2}C bus mode bus configuration, the input and output pins are shared. The device that is to receive data therefore must be made the N-ch open-drain output for transmission high-impedance state in advance. Therefore, write FFH to SIO0 in the 2-wire serial I/O mode. In the I^{2}C bus mode, set bit 7 (BSYE) of the serial bus interface control register (SBIC) to 1, and write FFH to SIO0.

SIO0 becomes undefined at $\overline{\text{RESET}}$.

Caution Do not execute an instruction that writes SIO0 in the I^{2}C bus mode while WUP (bit 5 of serial operation mode register 0 (CSIM0)) is 1 Even if such an instruction is not executed, data can be received while the wake-up function is being used (WUP = 1). For the details of the wake-up function, refer to 16.4.4 (1) (c) Wake-up function.

(2) Slave address register (SVA)

This 8-bit register sets the value of a slave address when the microcomputer is connected to the serial bus as a slave device. It 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.

The high-order 7 bits of data with its LSB masked by setting the bit 4 (SVAM) of the interrupt timing specification register (SINT) can compare with the slave address.

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. The wake-up function can be used by setting bit 5 (WUP) of CSIM0 to 1 in the I^{2}C bus mode. In this case, an interrupt request signal (INTCSI0) is generated when the slave address output by the master coincides with the value of SVA (the interrupt request signal is generated also when the stop condition is detected). This interrupt indicates that the master requests communication. When using the wake-up function, set SIC to 1.

When the microcontroller transmits data as the master or a slave in the 2-wire serial I/O mode or I^{2}C mode, errors can be detected by using SVA.

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/SDA0/P25 and SO0/SB1/SDA1/P26 pins. It can also be directly controlled by software.

(4) Serial clock counter

This counter counts the serial clocks output or input during transmit/receive 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 $\overline{\text{SCK0}}$/SCL/P27 pin.

(6) Interrupt request signal generation circuit

This circuit controls generation of an interrupt request signal. An interrupt request signal is generated as shown in Table 16-4 depending on the setting of bits 0 and 1 (WAT0 and WAT1) of the interrupt timing specification register (SINT) and bit 5 (WUP) of the serial operation mode register 0 (CSIM0).

(7) Acknowledge output circuit and stop condition/start condition/acknowledge detection circuit

These circuits output and detect various control signals in the I^{2}C bus mode.

They do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

Table 16-4. Generation of Interrupt Request Signal by Serial Interface Channel 0

Serial Transfer Mode	BSYE	WUP	WAT1	WAT0	ACKE	Description
3-wire serial I/O mode or 2-wire serial I/O mode	0	0	0	0	0	Generates interrupt request signal each time serial clock is counted eight times
	Others					Setting prohibited
I^2C bus mode (during transmission)	0	0	1	0	0	Generates interrupt request signal each time serial clock is counted eight times (8 clock wait). Normally, setting of WAT1, WAT0 = 1, 0 is not made during transmission. This setting is used only when reception and processing must be systematically arranged by software. Because ACK information is generated by reception side, ACKE is set to 0 (disabled).
			1	1	0	Generates interrupt request signal each time serial clock is counted nine times (9 clock wait). Because ACK information is generated by reception side, ACKE is set to 0 (disabled).
	Others					Setting prohibited
I^2C bus mode (during reception)	1	0	1	0	0	Generates interrupt request signal each time serial clock is counted eight times (8 clock wait). ACK information is output by manipulating ACKT by sofrware after interrupts request signal are generated.
			1	1	0/1	Generates interrupt request signal each time serial clock is counted nine times (9 clock wait). To generate ACK information automatically, ACKE is set to 1 (enable) before starting transfer. However, master sets ACKE to 0 (disable) before receiving last data.
	1	1	1	1	1	After receiving an address, generates an interrupt request signal when the values of the serial I/O shift register 0 (SIO0) and slave address register (SVA) coincide, and when the stop condition is detected. To generate ACK information automatically, ACKE is set to 1 (enable) before starting transfer.
	Others					Setting prohibited

Remark BSYE : Bit 7 of serial bus interface control register (SBIC)
ACKE : Bit 5 of serial bus interface control register (SBIC)

16.3 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 16-2)

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 16-3)

This register sets the serial clock and operation mode of serial interface channel 0, enables/disables the operation of the interface, sets the wakeup function, and indicates the coincidence signal of the address comparator.
CSIM0 is set by a 1-bit memory or 8-bit memory manipulation instruction.
This register is set to 00H when the $\overline{\text{RESET}}$ signal is input.

Caution Do not change the operation mode (3-wire serial I/O, 2-wire serial I/O, or I^{2}C bus) while the operation of serial interface channel 0 is enabled. To change the operation mode, stop the serial operation.

Figure 16-2. Format of Timer Clock Select Register 3

Symbol	7	6	5	4	3	2	1	0	Address	On 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	
				Serial clock in I²C bus mode	Serial clock in 3-wire serial I/O mode or 2-wire serial I/O mode
0	1	1	0	$f_X/2^6$ (156 kHz)	$f_X/2^2$ Note
0	1	1	1	$f_X/2^7$ (78.1 kHz)	$f_X/2^3$ (1.25 MHz)
1	0	0	0	$f_X/2^8$ (39.1 kHz)	$f_X/2^4$ (625 kHz)
1	0	0	1	$f_X/2^9$ (19.5 kHz)	$f_X/2^5$ (313 kHz)
1	0	1	0	$f_X/2^{10}$ (9.8 kHz)	$f_X/2^6$ (156 kHz)
1	0	1	1	$f_X/2^{11}$ (4.9 kHz)	$f_X/2^7$ (78.1 kHz)
1	1	0	0	$f_X/2^{12}$ (2.4 kHz)	$f_X/2^8$ (39.1 kHz)
1	1	0	1	$f_X/2^{13}$ (1.2 kHz)	$f_X/2^9$ (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 Before writing data other than that already written to TCL3, stop the serial transfer.

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

2. ( ) : At f_X = 10.0 MHz operation

Figure 16-3. Format of Serial Operation Mode Register 0

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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 $\overline{SCK0}$/SCL pin
1	0	Output of 8-bit timer register 2 (TM2) Note 2
1	1	Clock specified by bits 0-3 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 3 (input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS I/O)
		1								LSB			
1	1	0	Note 4 ×	Note 4 ×	0	0	0	1	2-wire serial I/O mode or I^2C bus mode	MSB	P25 (CMOS I/O)	SB1/SDA1 (N-ch open drain I/O)	$\overline{SCK0}$/SCL (N-ch open drain I/O)
		1	0	0	Note 4 ×	Note 4 ×	0	1			SB0/SDA0 (N-ch open drain I/O)	P26 (CMOS I/O)	

R/W

WUP	Controls wake-up function Note 5
0	Generates interrupt request signal in all modes each time serial transfer is executed
1	Generates interrupt request signal when address received after start condition is detected in I^2C bus mode (when CMDD = 1) coincides with data of slave address register (SVA)

R

COI	Slave address comparison result flag Note 6
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. The clock frequency is 1/16 of the frequency output by TO2 in the I^2C bus mode.

3. This pin can be used as P25 (CMOS input) when used only for transmission.

4. This pin can be freely used for port function.

5. Set bit 5 (SIC) of the interrupt timing specification register (SINT) to 1 when using the wake-up function. Do not execute an instruction that writes the serial I/O shift register 0 (SIO0) while WUP = 1.

6. COI is 0 when CSIE0 = 0.

Remark × : Don’t care
PM×× : Port mode register
P×× : Output latch of 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 16-4. Format of Serial Bus Interface Control Register (1/2)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	On reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W Note

R/W	RELT	Used to output stop condition in I^2C bus mode. 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 start condition in I^2C bus mode. 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	Stop condition 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 stop condition is detected in I^2C mode

R	CMDD	Start condition detection

Clearing conditions (CMDD = 0)	Setting condition (CMDD = 1)
• When transfer start instruction is executed • When stop condition is detected in I^2C mode • When CSIE0 = 0 • When $\overline{\text{RESET}}$ is input	• When start condition is detected in I^2C bus mode

Note Bits 2, 3, and 6 (RELD, CMDD, and ACKD) are read-only bits.

Remark CSIE0: Bit 7 of the serial operation mode register 0 (CSIM0)

Figure 16-4. Format of Serial Bus Interface Control Register (2/2)

R/W	ACKT	Makes SDA0 (SDA1) low immediately after execution of setting instruction (ACKT = 1) until falling edge of next SCL. Used to generate $\overline{\text{ACK}}$ signal by software when 8-clock wait is selected. This bit is cleared to 0 when transfer of serial interface is started or when CSIE0 = 0.

R/W	ACKE	Acknowledge signal automatic output control [Note 1]
	0	Disables automatic output of acknowledge signal (output by ACKT is enabled). Used for transmission, or reception with 8-clock wait selected.[Note 2]
	1	Enables automatic output of acknowledge signal. Outputs acknowledge signal in synchronization with falling edge of 9th clock of SCL (automatically outputs when $\overline{\text{ACKE}}$ = 1). This bit is not automatically cleared to 0 after acknowledge signal has been output. Used for reception with 9-clock wait selected.

R	ACKD	Acknowledge detection
	Clearing conditions (ACKD = 0)	Setting condition (ACKD = 1)
	• When transfer start instruction is executed • When $\overline{\text{CSIE0}}$ = 0 • When $\overline{\text{RESET}}$ is input	• When acknowledge signal is detected at rising edge of clock of SCL after completion of transfer

R/W	BSYE [Note 3]	Controls N-ch open-drain output for transmission in I^2C bus mode [Note 4]
	0	Enables output (transmission)
	1	Disables output (reception)

Notes **1.** Set this bit before starting transfer.

2. Output the acknowledge signal during reception by using ACKT when 8-clock wait is selected.

3. The wait mode can be released when transfer by the serial interface has been started when an address signal has been received. However, the BSYE flag is not cleared to 0.

4. Be sure to set BSYE to 1 when using the wake-up function.

Remark CSIE0: Bit 7 of the serial operation mode register 0 (CSIM0)

(4) Interrupt timing specification register (SINT)

This register controls the interrupt, wait and clock level, sets address mask function, and indicates the status of the level of the SCK0/SCL/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 16-5. Format of Interrupt Timing Specification Register (1/2)

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	0	Address	On reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W[Note 1]

R/W	WAT1	WAT0	Controls wait and interrupt
	0	0	Generates interrupt processing request at rising edge of 8th clock of $\overline{\text{SCK0}}$ (clock output goes to high-impedance state)
	0	1	Setting prohibited
	1	0	Used in I^{2}C bus mode (8-clock wait). Generates interrupt processing request at rising edge of 8th clock of SCL (master makes SCL output low and waits after outputting 8 clocks. Slave makes SCL pin low and requests for wait after inputting 8 clocks).
	1	1	Used in I^{2}C bus mode (9-clock wait). Generates interrupt processing request at rising edge of 9th clock of SCL (master makes SCL output low and waits after outputting 9 clocks. Slave makes SCL pin low land requests for wait after inputting 9 clocks).

R/W	WREL	Controls clearing wait
	0	Wait clear status
	1	Clears wait status. This bit is automatically cleared to 0 after wait status has been cleared (used to clear wait status set by WAT0, WAT1)

R/W	CLC	Controls clock level[Note 2]
	0	Used in I^{2}C bus mode. Makes output level of SCL pin low when serial transfer is not executed.
	1	Used in I^{2}C bus mode. Puts output level of SCL pin in high-impedance state when serial transfer is not executed (the clock line is high). Master uses this setting to make SCL high to generate start/stop condition.

Notes 1. Bit 6 (CLD) is read-only bit.

2. Set CLC to 0 when the I^{2}C bus mode is not used.

Figure 16-5. Format of Interrupt Timing Specification Register (2/2)

R/W	SVAM	Bits of SVA used as slave address
	0	Bits 0-7
	1	Bits 1-7

R/W	SIC	Selects interrupt source of INTCSI0 Note 1
	0	Sets CSIIF0 to 1 at end of transfer by serial interface channel 0
	1	Sets CSIIF0 to 1 at end of transfer by serial interface channel 0 or on detection of stop condition in I^2C bus mode.

R	CLD	Level of $\overline{\text{SCK0}}$/SCL pin Note 2
	0	Low level
	1	High level

Notes 1. Set SIC to 1 when using wake-up function in the I^2C bus mode.

2. CLD is 0 when CSIE0 = 0.

Remark SVA : Slave address register

CSIIF0: Interrupt request flag corresponding to INTCSI0

CSIE0 : Bit 7 of the serial operation mode register 0 (CSIM0)

16.4 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
- 2-wire serial I/O mode
- I^{2}C (Inter IC) bus mode

16.4.1 Operation stop mode

Serial transfer is not executed in this mode. Consequently, the power consumption 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/SDA0, P26/SO0/SB1/SDA1, and P27/$\overline{\text{SCK0}}$/SCL 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 $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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

16.4.2 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 ($\overline{\text{SCK0}}$), 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 $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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 0-3 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/SDA0/P25 pin function	SO0/SB1/SDA1/P26 pin function	$\overline{SCK0}$/SCL/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	1	2-wire serial I/O mode (Refer to **16.4.3 Operation in 2-wire serial I/O mode**.) I^{2}C bus mode (Refer to **16.4.4 Operation in I^{2}C bus 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 start condition has been detected in I^{2}C bus mode (when CMDD = 1) coincides with data of slave address register

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 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	On 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 the serial operation mode register 0 (CSIM0)

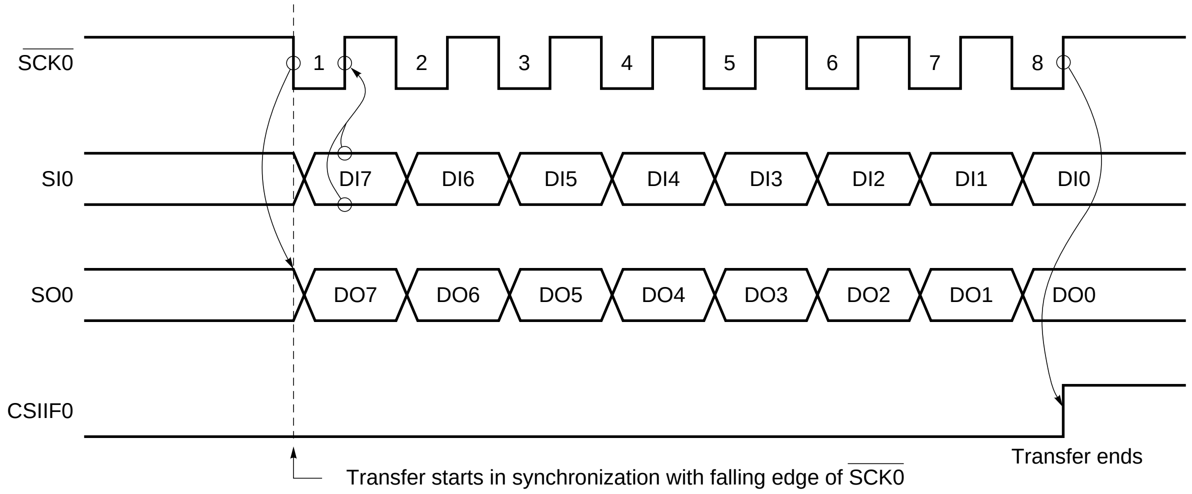
(2) Communication operation

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 ($\overline{\text{SCK0}}$). 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 $\overline{\text{SCK0}}$.

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 16-6. Timing of 3-Wire Serial I/O Mode



The SO0 pin serves as a CMOS output pin and outputs the status of the SO0 latch. The output status of this pin can be manipulated by setting bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC).

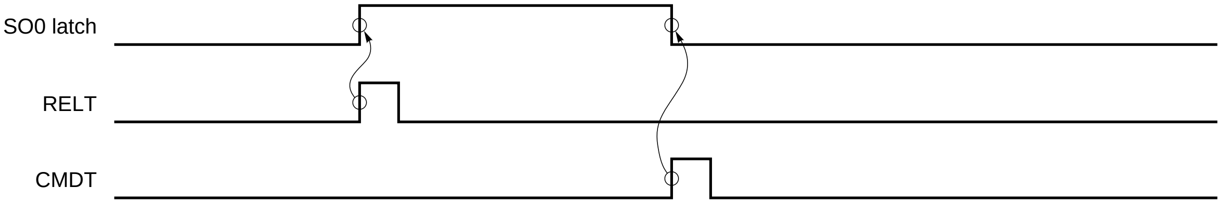
However, do not perform this manipulation during serial transfer.

The output level of the $\overline{\text{SCK0}}$ pin is controlled by manipulating the P27 output latch in the output mode (mode of the internal system clock) (refer to **16.4.7 Manipulating $\overline{\text{SCK0}}$/SCL/P27 pin output**).

(3) Signals

Figure 16-7 shows the operations of RELT and CMDT.

Figure 16-7. Operations of RELT and CMDT



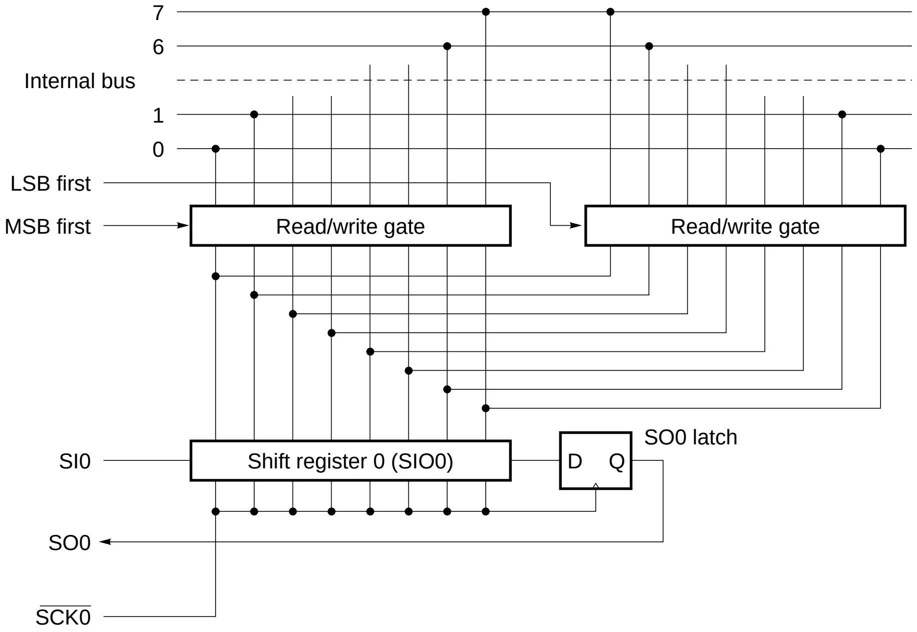
(4) MSB/LSB first selection

In the 3-wire serial I/O mode, whether data is transferred with the MSB or LSB first can be selected.

Figure 16-8 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 16-8. 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, specify whether the MSB or LSB is first 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 $\overline{\text{SCK0}}$ 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.

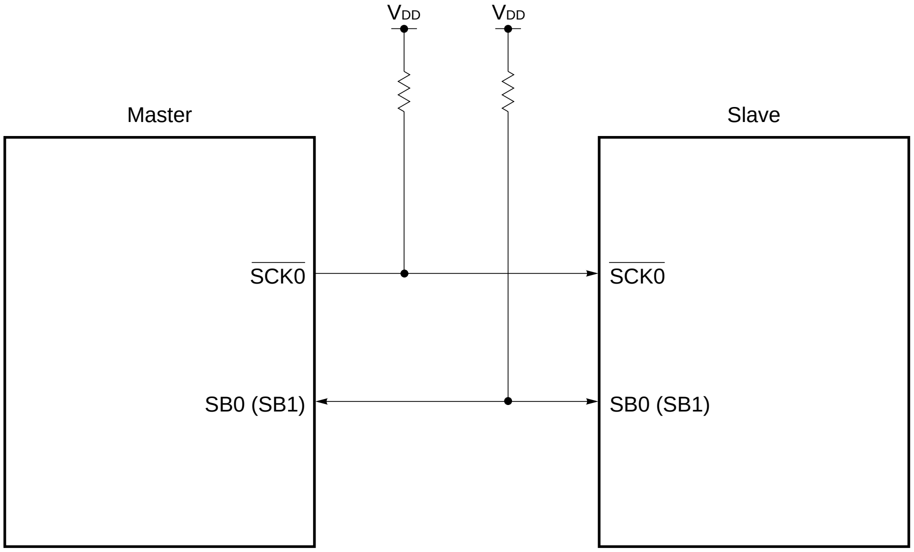
Serial transfer is automatically stopped at the end of 8-bit transfer, and an interrupt request flag (CSIIF0) is set.

16.4.3 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 ($\overline{\text{SCK0}}$) and serial data I/O (SB0 or SB1), are used to establish communication in this mode.

Figure 16-9. 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 $\overline{\text{RESET}}$ signal is input.

Symbol	<7>	<6>	<5>	4	3	2	1	0	Address	On 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 0-3 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/SDA0/P25 pin function	SO0/SB1/SDA1/P26 pin function	$\overline{\text{SCK0}}$/SCL/P27 pin function
0	×	3-wire serial I/O mode (Refer to **16.4.2 Operation in 3-wire serial I/O mode**.)											
1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode or I²C bus mode	MSB	P25 (CMOS I/O)	SB1/SDA1 (N-ch open drain I/O)	$\overline{\text{SCK0}}$/SCL (N-ch open drain I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0/SDA0 (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 start condition has been detected in I²C bus mode (when CMDD = 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 clear 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 port

Phase-out/Discontinued

(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	On 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 the 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	On reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	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 stop condition in I^2C mode

R

CLD	Level of $\overline{\text{SCK0}}$/SCL/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 When using 2-wire serial I/O mode, be sure to set bits 0 through 3 to 0.

Remark CSIIF0: Interrupt request flag corresponding to INTCSI0

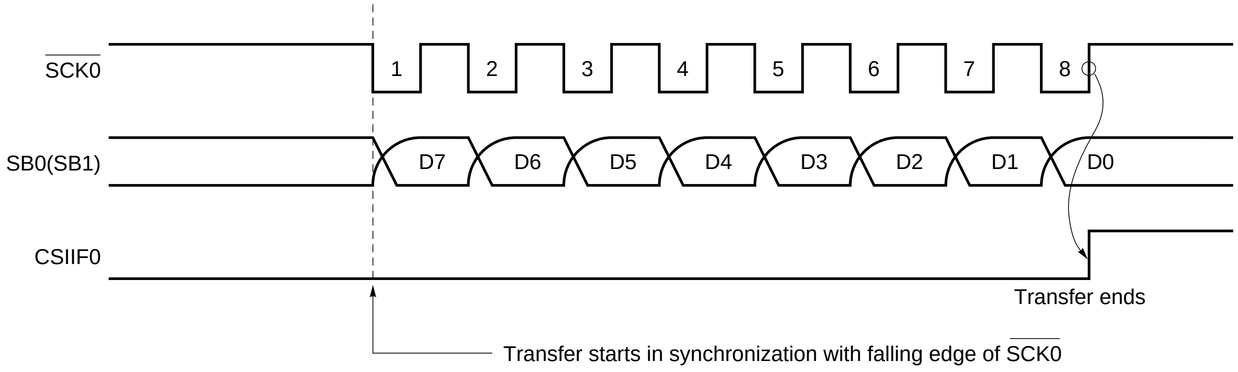
(2) Communication operation

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 ($\overline{\text{SCK0}}$). 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 the SIO0 at the rising edge of $\overline{\text{SCK0}}$.

When the 8-bit data has been completely transferred, the operation of the shift register is automatically stopped, and an interrupt request flag (CSIIF0) is set.

Figure 16-10. Timing of 2-Wire Serial I/O Mode


$\overline{\text{SCK0}}$
1
2
3
4
5
6
7
8
SB0(SB1)
D7
D6
D5
D4
D3
D2
D1
D0
CSIIF0
Transfer ends
Transfer starts in synchronization with falling edge of $\overline{\text{SCK0}}$

The pin specified as the serial data bus SB0 (or SB1) 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 open-drain output must be made high-impedance state.

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 bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not manipulate the output status of the pin during serial transfer.

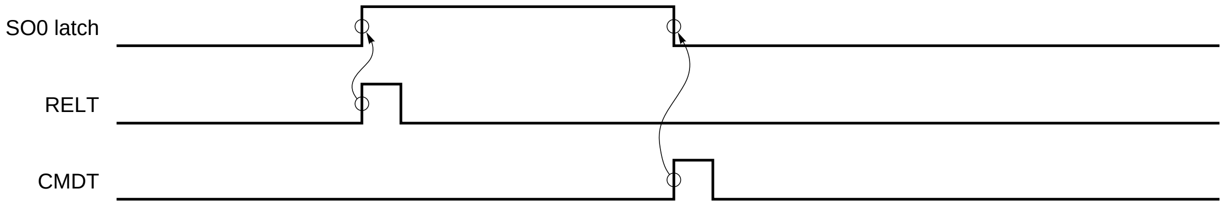
The output level of the $\overline{\text{SCK0}}$ pin is controlled by manipulating the P27 output latch in the output mode (mode of the internal system clock) (refer to **16.4.7 Manipulating $\overline{\text{SCK0}}$/SCL/P27 pin output**).

Phase-out/Discontinued

(3) Signals

Figure 16-11 shows the operations of RELT and CMDT.

Figure 16-11. 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 $\overline{\text{SCK0}}$ 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 output must be made high-impedance state 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 transmittion error has occurred.

16.4.4 Operation in I^{2}C bus mode

The I^{2}C bus mode is used to perform single-master and slave operations on the I^{2}C bus. In this mode, a single-master serial bus can communicate with two or more slave devices by using two signal lines: serial clock (SCL) and serial data bus (SDA0 or SDA1). The format of this mode is based on the clocked serial I/O mode with added functions to configure the bus. When organizing a serial bus with several microcomputers and peripheral ICs, therefore, the number of ports and wiring length on the printed wiring board can be reduced.

The master can output a “start condition”, “data”, and “stop condition” to slaves on the serial data bus.

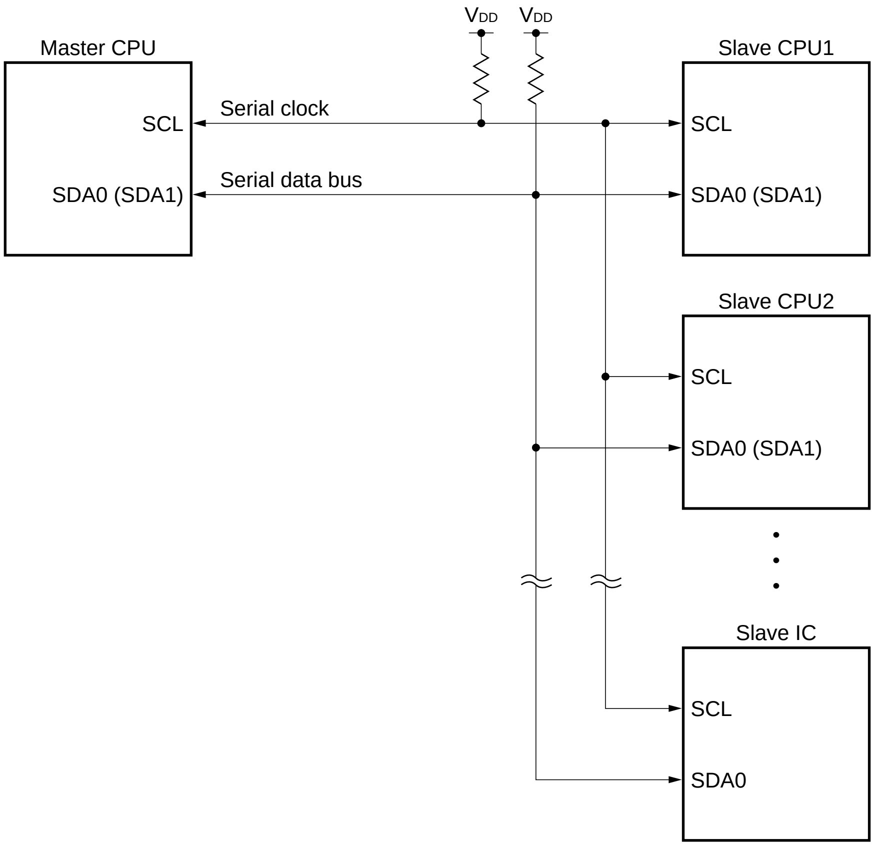
A slave receives and automatically detects these data by hardware. This function simplifies the application program that controls the I^{2}C bus.

Figure 16-12 shows an example of serial bus configuration by using CPUs or peripheral ICs having a serial interface conforming to the I^{2}C bus.

Because the serial clock pin (SCL) and serial data bus pin (SDA0 or SDA1) are N-ch open-drain output pins in the I^{2}C bus, pull-up resistors must be connected to the serial clock line and serial data bus line.

Table 16-5 describes the signals used in the I^{2}C bus mode.

Figure 16-12. Example of Serial Bus Configuration in I^{2}C Bus Mode



(1) Functions in the I^{2}C bus mode

The following functions are available in the I^{2}C bus mode:

(a) Automatic identification of serial data

The “start condition”, “data”, and “stop condition” on the serial data bus are automatically detected.

(b) Chip select by address

The master can select a specific slave device from those connected to the I^{2}C bus by transmitting a slave address and communicate with that slave.

(c) Wake-up function

When a slave operates, it generates an interrupt request when the address it has received from the master coincides with the value of the slave address register (SVA) (the interrupt request is generated also when the stop condition is detected). Therefore, the slaves on the I^{2}C bus other than the one selected by the master can operate independently of the serial communication.

(d) Acknowledge signal ($\overline{\text{ACK}}$) control function

The acknowledge signal that is used to check whether serial communication has been correctly executed can be controlled during the master and slave operations.

(e) Wait signal ($\overline{\text{WAIT}}$) control function

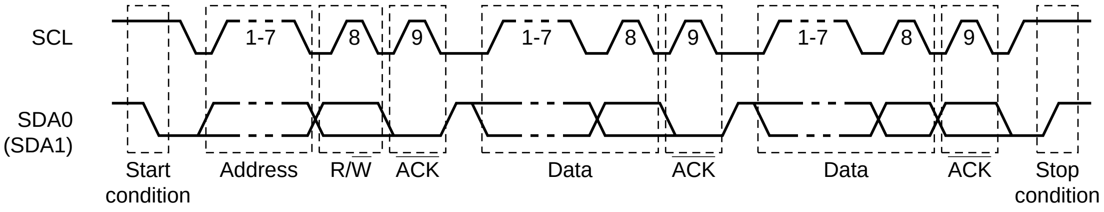
A slave device can control the wait signal that indicates the busy status of the slave.

(2) Definition of the I^{2}C bus

The following describes the serial data communication format of the I^{2}C bus and the meanings of the signals used.

Figure 16-13 shows the transfer timing of the “start condition”, “data”, and “stop condition”output to the I^{2}C serial data bus.

Figure 16-13. Serial Data Transfer Timing on I^{2}C Bus



The start condition, slave address, and stop condition are output by the master.

The acknowledge signal ($\overline{\text{ACK}}$) is output by either the master or slave (usually, this signal is output by the side that receives 8-bit data).

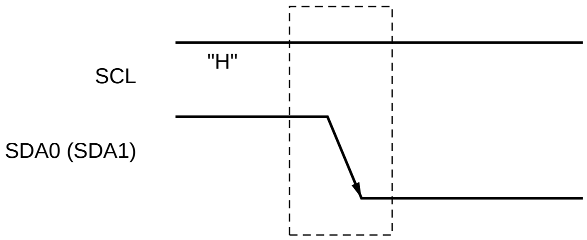
The serial clock (SCL) is continuously output by the master.

(a) Start condition

The start condition is output to the serial data bus when the SDA0 (SDA1) pin goes low while the SCL pin is high. Therefore, the start condition of the SCL and SDA0 (SDA1) pins is a signal output by the master when the master starts serial transfer to a slave. For the details about start condition output, refer to **16.4.5 Notes on using I^{2}C bus mode**.

The slave has hardware that detects the start condition.

Figure 16-14. Start Condition



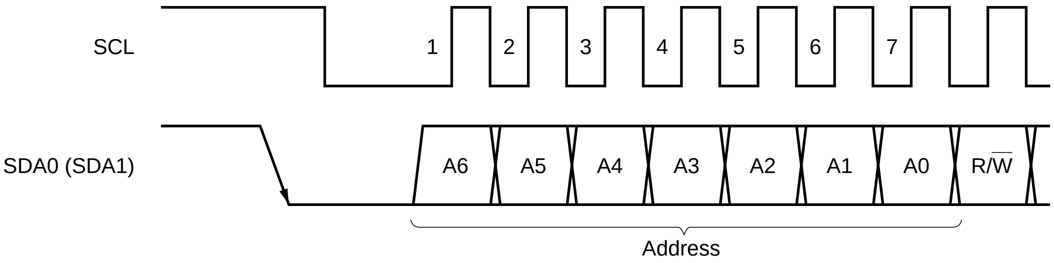
(b) Addresses

The 7-bit data following the start condition is defined to be an address.

An address is 7 bit of data output by the master to select a specific slave from those connected to the bus line. Therefore, all the slaves on the bus line must have a different address.

A slave detects the start condition by hardware and checks whether the 7-bit data output by the master coincides with the value of the slave address register (SVA) of the slave. If the 7-bit data coincides with the value of the slave address register of the slave, the slave is selected. After that, communication takes place between the master and this slave, until the master transmits a start or stop condition.

Figure 16-15. Addresses

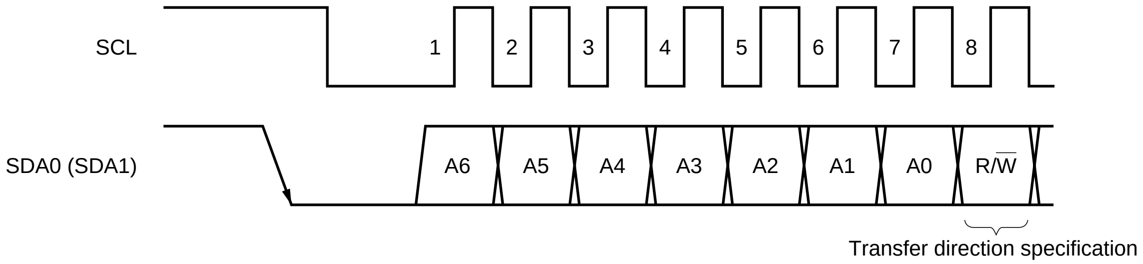


(c) Transfer direction specification

Following the 7-bit address, the master transmits 1-bit of data to specify the direction of transfer.

If this transfer direction specification bit is 0, the master transmits data to the slave. If the bit is 1, the master receives data from the slave.

Figure 16-16. Transfer Direction Specification



(d) Acknowledge signal ($\overline{ACK}$)

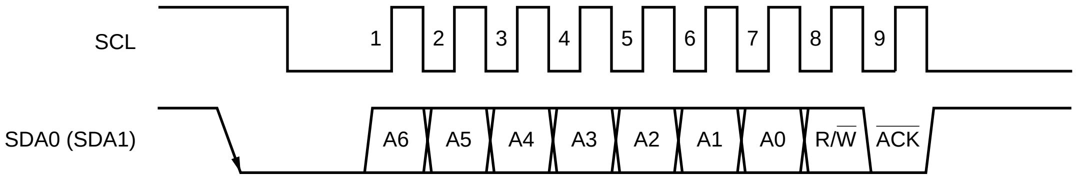
The acknowledge signal is used to confirm that serial data has been received at transmission and reception sides. The reception side returns the acknowledge signal each time it has received 8 bits of data.

The reception side outputs the acknowledge signal usually after it received 8 bits of data.

If the master is receiving data, however, it does not output the acknowledge signal after it has received the last data.

The transmission side checks whether the reception side has returned the acknowledge signal after it has transmitted 8 bits of data. When the acknowledge signal has been returned, it is assumed that the 8-bit data has been correctly received, and the next processing is performed. If a slave does not return the acknowledge signal, it has not received the data correctly. Consequently, the master outputs a stop condition to abort transmission.

Figure 16-17. Acknowledge Signal



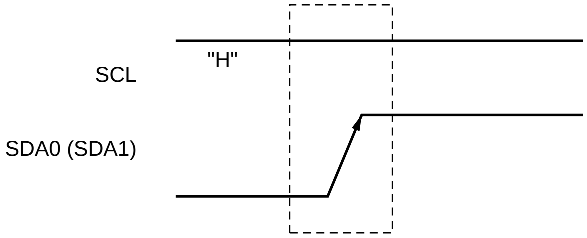
(e) Stop condition

The stop condition is output when the SDA0 (SDA1) pin goes high while the SCL pin is high.

The stop condition is output by the master to the slave when serial transfer has been completed.

The slave has hardware to detect the stop condition.

Figure 16-18. Stop Condition



(f) Wait signal (WAIT)

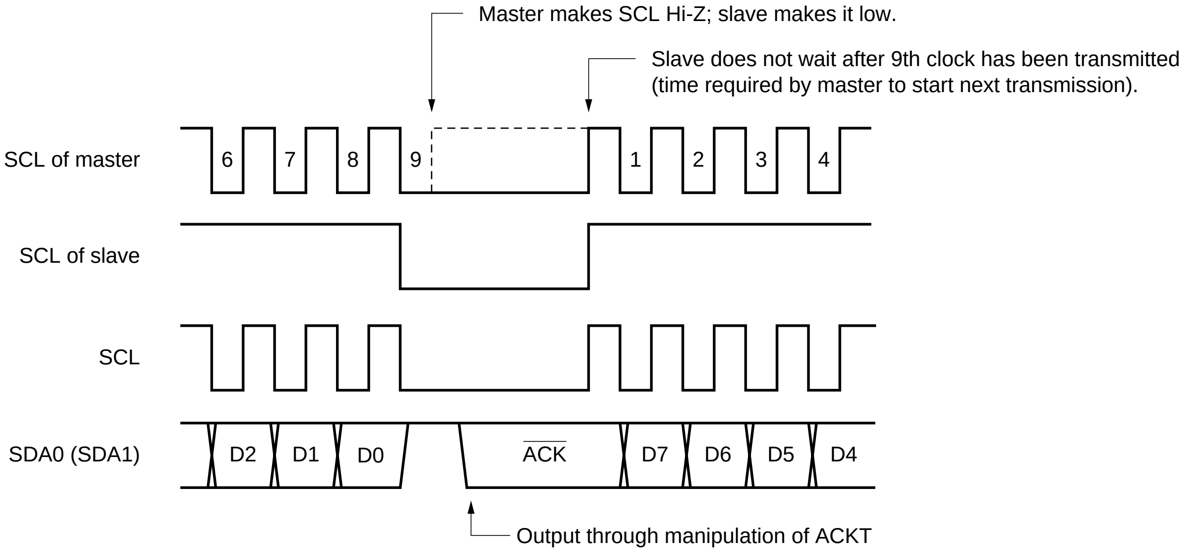
The wait signal is output by a slave to the master to indicate that the slave is getting ready for data transmission/reception (in wait status).

The slave informs the master that it is in the wait status by making the SCL pin low.

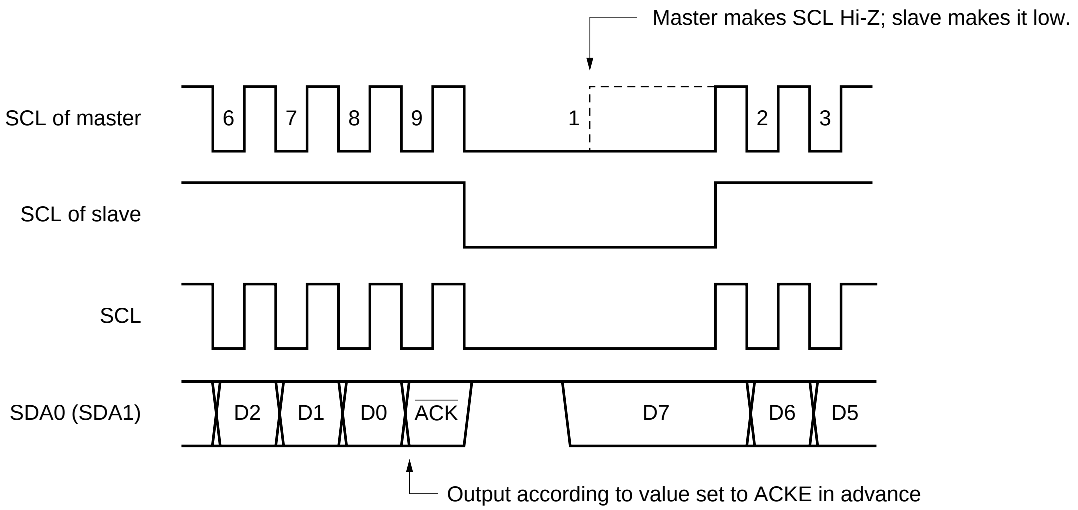
When the slave is released from the wait status, the master can start the next transfer. For how to release a slave from the wait status, refer to **16.4.5 Notes on using I^{2}C bus mode**.

Figure 16-19. Wait Signal

(a) 8-clock wait


Master makes SCL Hi-Z; slave makes it low.
Slave does not wait after 9th clock has been transmitted (time required by master to start next transmission).
SCL of master
6
7
8
9
1
2
3
4
SCL of slave
SCL
SDA0 (SDA1)
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5
D4
Output through manipulation of ACKT

(b) 9-clock wait


Master makes SCL Hi-Z; slave makes it low.
SCL of master
6
7
8
9
1
2
3
SCL of slave
SCL
SDA0 (SDA1)
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5
Output according to value set to ACKE in advance

(3) Register setting

The I²C bus 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 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	On reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM 04	CSIM 03	CSIM 02	CSIM 01	CSIM 00	FF60H	00H	R/W[Note 1]

R/W

CSIM01	CSIM00	Selects clock of serial interface channel 0
0	×	External clock input to $\overline{\text{SCK0}}$/SCL pin
1	0	Output of 8-bit timer register 2 (TM2) [Note 2]
1	1	Clock specified by bits 0-3 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/SDA0/P25 pin function	SO0/SB1/SDA1/P26 pin function	$\overline{\text{SCK0}}$/SCL/P27 pin function
0	×	3-wire serial I/O mode (Refer to **16.4.2 Operation in 3-wire serial I/O mode**.)											
1	1	0	Note 3 ×	Note 3 ×	0	0	0	1	2-wire serial I/O mode or I²C bus mode	MSB	P25 (CMOS I/O)	SB1/SDA1 (N-ch open-drain I/O)	$\overline{\text{SCK0}}$/SCK (N-ch open-drain I/O)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0/SDA0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

(Cont'd)

Notes 1. Bit 6 (COI) is read-only.

2. The clock frequency is 1/16 of the frequency output by TO2 in the I²C bus mode.

3. This pin can be used freely as a port pin.

Remark	
×	: Don't care
PM××	: Port mode register
P××	: Output latch of port

R/W

WUP	Controls wake-up function [Note 1]
0	An interrupt request signal is generated each time a serial transfer is executed in all modes
1	When in I^{2}C bus mode, after the start condition is detected (CMDD = 1), if the received address coincides with the slave address register (SVA) an interrupt request signal is generated.

R

COI	Slave address comparison result flag [Note 2]
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 I/O shift register 0 (SIO0).

R/W

CSIE0	Controls operation of serial interface channel 0
0	Stops operation
1	Enables operation

Notes 1. Set bit 5 (SIC) of the interrupt timing specification register (SINT) to 1 when using the wake-up function. Do not execute an instruction that writes the serial I/O shift register 0 (SIO0) while WUP = 1.

2. COI is 0 when CSIE = 0.

(b) Serial bus interface control register (SBIC)

The SBIC 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	On reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W [Note]

R/W

RELT	Used to output stop condition in I^{2}C bus mode. SO0 latch is set to 1 when RELT = 1. After SO0 latch has been set, this bit is automatically cleared to 0. It is also cleared to 0 when CSIE = 0.

R/W

CMDT	Used to output start condition in I^{2}C bus mode. SO0 latch is cleared to 0 when CMDT = 1. After SO0 latch has been cleared, this bit is automatically cleared to 0 It is also cleared to 0 when CSIE0 = 0.

R

RELD	Detects stop condition
Clearing conditions (RELD = 0)	Setting condition (RELD = 1)
• On execution of transfer start instruction • If values of SIO0 and SVA do not coincide on address reception • When CSIE0 = 0 • On $\overline{\text{RESET}}$	• When stop condition is detected in I^{2}C bus mode

(Cont'd)

Note Bits 2, 3, and 6 (RELD, CMDD, and ACKD) are read-only bits.

Remark CSIE0: Bit 7 of the serial operation mode register 0 (CSIM0)

R	CMDD	Detects start condition
	Clearing conditions (CMDD = 0)	Setting condition (CMDD = 1)
	• On execution of transfer start instruction • On detection of stop condition in I^{2}C bus mode • When CSIE0 = 0 • On $\overline{\text{RESET}}$	• On detection of start condition in I^{2}C bus mode

R/W	ACKT	Makes SDA0 (SDA1) low immediately after execution of setting instruction (ACKT = 1) until falling edge of next SCL. Used to generate $\overline{\text{ACK}}$ signal by software when 8-clock wait is selected. Cleared to 0 when serial interface starts transfer or when CSIE0 = 0.

R/W	ACKE	Controls automatic output of acknowledge signal Note 1
	0	Disables automatic output of acknowledge signal (output by ACKT is possible). Used for transmission, or reception with 8-clock wait selected Note 2.
	1	Enables automatic output of acknowledge signal. Outputs acknowledge signal in synchronization with falling edge of 9th clock of SCL (automatically output when $\overline{\text{ACKE}}$ = 1). This bit is not automatically cleared to 0 after acknowledge signal has been output. Used for reception with 9-clock wait selected.

R	ACKD	Detects acknowledge
	Clearing conditions (ACKD = 0)	Setting condition (ACKD = 1)
	• On execution of transfer start instruction • When CSIE0 = 0 • On $\overline{\text{RESET}}$	• On detection of acknowledge signal at rising edge of SCL clock after transfer has been completed

R/W	Note 3 BSYE	Controls N-ch open-drain output for transmission in I ^{2}C bus mode Note 4
	0	Enables output (transmission)
	1	Disables output (reception)

Notes 1. Set this bit before starting transfer.

2. Output the acknowledge signal during reception by using ACKT when 8-clock wait is selected.

3. The wait status can be released by starting the transfer of the serial interface or receiving an address signal. However, BSYE is not cleared to 0.

4. Be sure to set BSYE to 1 when using the wake-up function.

Remark CSIE0: Bit 7 of the serial operation mode register 0 (CSIM0)

(c) Interrupt timing specification register (SINT)

SINT 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	On reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W[Note 1]

R/W

WAT1	WAT0	Controls wait and interrupt[Note 2]
0	0	Generates interrupt processing request at rising edge of 8th clock of $\overline{\text{SCK0}}$ (clock output goes to high-impedance state)
0	1	Setting prohibited
1	0	Used in I^{2}C bus mode (8-clock wait). Generates interrupt processing request at rising edge of 8th clock of SCL. (After outputting 8 clocks, master makes SCL output low and waits. After inputting 8 clocks, slave makes SCL pin low and requests wait.)
1	1	Used in I^{2}C bus mode (9-clock wait). Generates interrupt processing request at rising edge of 9th clock of SCL. (After outputting 9 clocks, master makes SCL output low and waits. After inputting 9 clocks, slave makes SCL pin low and requests wait.)

R/W

WREL	Controls releasing wait
0	Wait released status
1	Released wait status. This bit is automatically cleared to 0 after wait status has been released (used to release wait status set by WAT0, WAT1).

R/W

CLC	Controls clock level
0	Used in I^{2}C bus mode. Makes output level of SCL pin low when serial transfer is not executed.
1	Used in I^{2}C bus mode. Puts output level of SCL pin in high-impedance state when serial transfer is not executed (the clock line is high). Master uses this setting to make SCL high to generate start/stop condition.

(Cont'd)

Notes 1. Bit 6 (CLD) is read-only bit.

2. Set WAT1 and WAT0 to 1, 0 or 1, 1 in the I^{2}C bus mode.

R/W	SVAM	SVA bits used as slave address
	0	Bits 0-7
	1	Bits 1-7

R/W	SIC	Selects INTCSI0 interrupt source Note 1
	0	Sets CSIIF0 to 1 at end of transfer on serial interface channel 0
	1	Sets CSIIF0 to 1 at end of transfer on serial interface channel 0 or on detection of stop condition in I^2C bus mode.

R/W	CLD	Level of $\overline{\text{SCK0}}$/SCL pin Note 2
	0	Low level
	1	High level

Notes **1.** Set SIC to 1 when using the wake-up function in the I^2C bus mode.

2. CLD is 0 when CSIE0 = 0.

Remark SVA : Slave address register
CSIIF0: Interrupt request flag corresponding to INTCSI0
CSIE0 : Bit 7 of the serial operation mode register 0 (CSIM0)

(4) Signals

Table 16-5 lists the signals used in the I^{2}C bus mode.

Table 16-5. Signals in I^{2}C Bus Mode

Signal Name	Output Device	Definition	Output Condition	Influence on Flag	Meaning of Signal
Start condition	Master	Falling edge of SDA0 (SDA1) when SCL is high [Note 1]	Setting of CMDT	Sets CMDD	Indicates that address is transmitted next and that serial communication is started
Stop condition	Master	Rising edge of SDA0 (SDA1) when SCL is high [Note 1]	Setting of RELT	Sets RELD Clears CMDD	Indicates end of serial transmission
Acknowledge signal ($\overline{ACK}$)	Master/slave	Low-level signal of SDA0 (SDA1) output during 1-clock period of SCL after completion of serial reception	• ACKE = 1 • Setting of ACKT	Sets ACKD	Indicates that 1 byte has been completely received
Wait ($\overline{WAIT}$)	Slave	Low-level signal output to SCL	WAT1, WAT0 = 1 X	–	Indicates that serial reception cannot be executed
Serial clock (SCL)	Master	Synchronization clock for outputting signals	Execution of instruction that writes data to SIO0 when CSIE0 = 1 (serial transfer start instruction) [Note 2]	Sets CSIIF0 [Note 3]	Synchronization signal for serial communication
Address (A6-A0)	Master	7-bit data output in synchronization with SCL after start condition has been output			Indicates address value on serial bus that specifies slave
Transfer direction (R/$\overline{W}$)	Master	1-bit data output in synchronization with SCL after address has been output			Indicates whether data is transmitted or received
Data (D7-D0)	Master/slave	8-bit data output in synchronization with SCL not immediately after start condition			Indicates data actually communicated

Notes 1. The level of the serial clock can be controlled by the CLC bit in interrupt timing specification register (SINT).

2. Serial transfer is started in the wait status after the wait status has been released.

3. If 8-clock wait is selected with WUP = 0, CSIIF0 is set at the rising edge of the eighth clock of SCL. When 9-clock wait is selected with WUP = 0, CSIIF0 is set at the rising edge of the ninth clock of SCL. An address is received when WUP = 1 and CSIIF0 is set if that address coincides with the value of the slave address register (SVA), and if the stop condition is detected

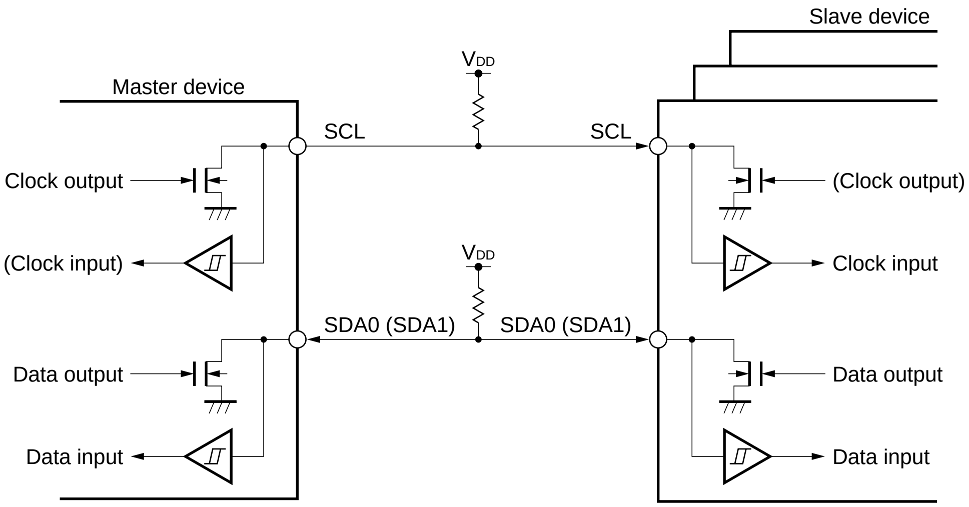
(5) Pin configuration

The serial clock pin (SCL) and serial data bus pin (SDA0 or SDA1) are configured as follows:

(a) SCL .... Pin that inputs/outputs serial clock
<1> Master ...... N-ch open-drain output
<2> Slave .... Schmitt input
(b) SDA0 (SDA1) .... Serial data input/output dual pin.
N-ch open-drain output and Schmitt input for both master and slave

Because both the serial clock and serial data bus are N-ch open-drain output, they must be connected to external pull-up resistors.

Figure 16-20. Pin Configuration



Caution Because the N-ch open drain output must be made high-impedance state when data is received, set bit 7 (BSYE) of the serial bus interface control register (SBIC) to 1 in advance and write FFH to the serial I/O shift register 0 (SIO0).
However, when the wake-up function is used (when bit 5 (WUP) of the serial operation mode register 0 (CSIM0) is set), the N-ch open drain output is always at high-impedance state even if FFH is not written to SIO0.

(6) Detecting address coincidence

In the I^2C bus mode, the master can select a specific slave device by transmitting a slave address to it.
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) is 1 and only if the slave address transmitted by the master coincides with the address set to the SVA, CSIIF0 Is set (CSIIF0 is also set when the stop condition is detected).
Set SIC to 1 when the wake-up function is used.

Caution Whether a slave is selected or not is detected by coincidence of the data (address) received after the start condition.
To detect this coincidence, an address coincidence detection interrupt (INTCSI0) that occurs when WUP = 1, is usually used. Therefore, to enable detection of whether a slave is selected or not, be sure that WUP = 1.

(7) Error detection

Because the status of serial bus SDA0 (SDA1) during transmission is also loaded to the serial I/O shift register 0 (SIO0) in the I^2C bus mode, a transmission error can be detected in the following ways:

(a) By comparing SIO0 data before and after transmission

If the two data are different, it is assumed that a transmission error has occurred.

(b) By using slave address register (SVA)

The transmission data is placed in SIO0 and SVA, and transmission is executed. After transmission has been completed, the COI bit (that indicates the coincidence signal from the address comparator) of serial operation mode register 0 (CSIM0) is tested. If this bit is “1”, transmission has been completed normally. If it is “0”, a transmission error has occurred.

(8) Communication operation

In the I^2C bus mode, the master outputs an address onto the serial bus to select one of the slave devices to be communicated.
Following the slave address, the master transmits an $R/\overline{W}$ bit that indicates the transfer direction of data, and starts serial communication with the slave.
Timing charts for data communication are shown in Figures 16-21 to 16-22.
The serial I/O shift register 0 (SIO0) performs a shift operation in synchronization with the falling edge of the serial clock (SCL), and the transmitted data is transferred to the SO0 latch and is output from the SDA0 or SDA1 pin, with MSB first.
The data input to the SDA0 or SDA1 pin is loaded to the shift register (SIO0) at the rising edge of SCL.

Figure 16-21. Example of Communication from Master to Slave (with 9-clock wait selected for both master and slave) (1/3)

(a) Start condition - address

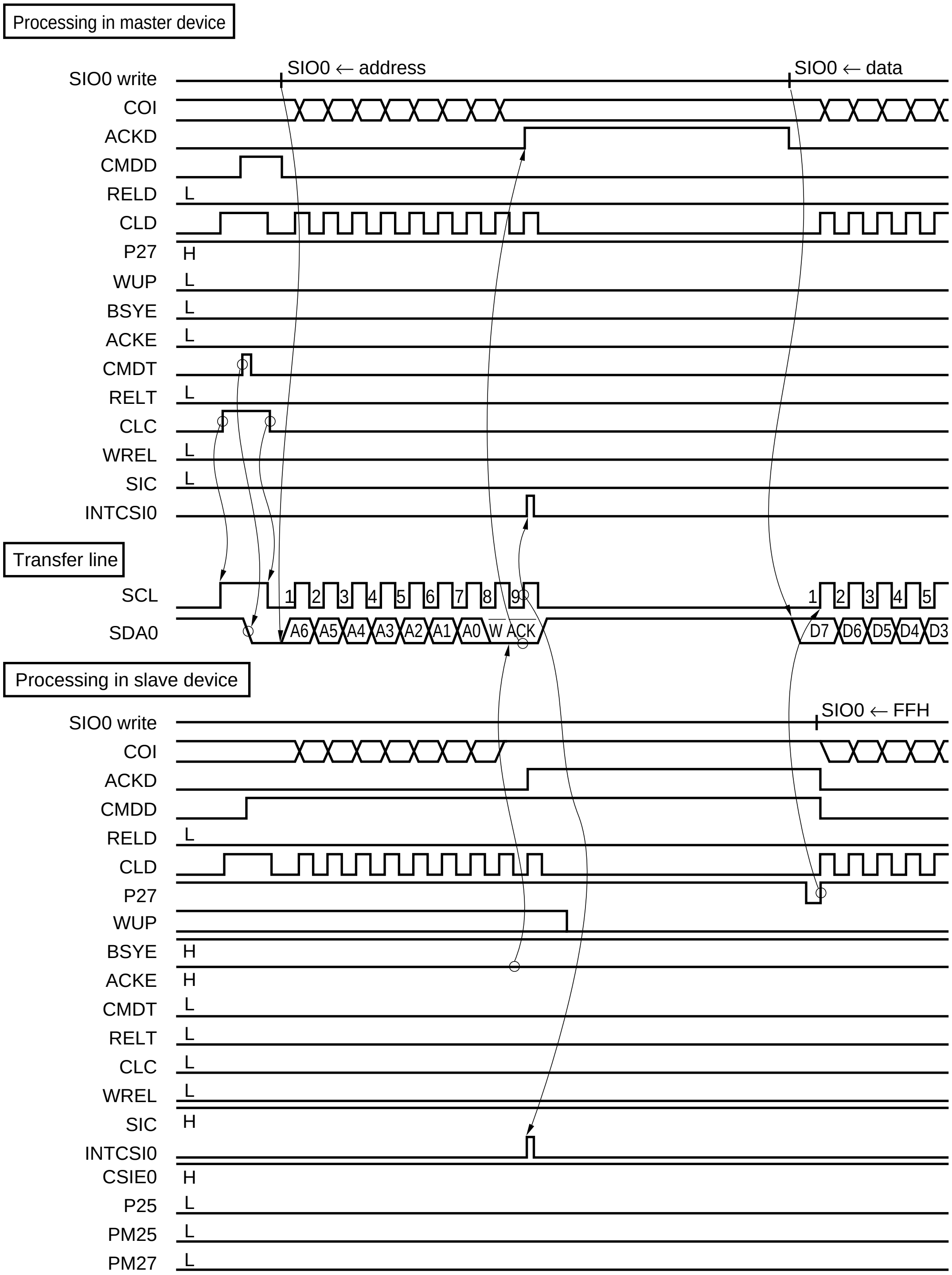


Figure 16-21. Example of Communication from Master to Slave (with 9-clock wait selected for both master and slave) (2/3)

(b) Data

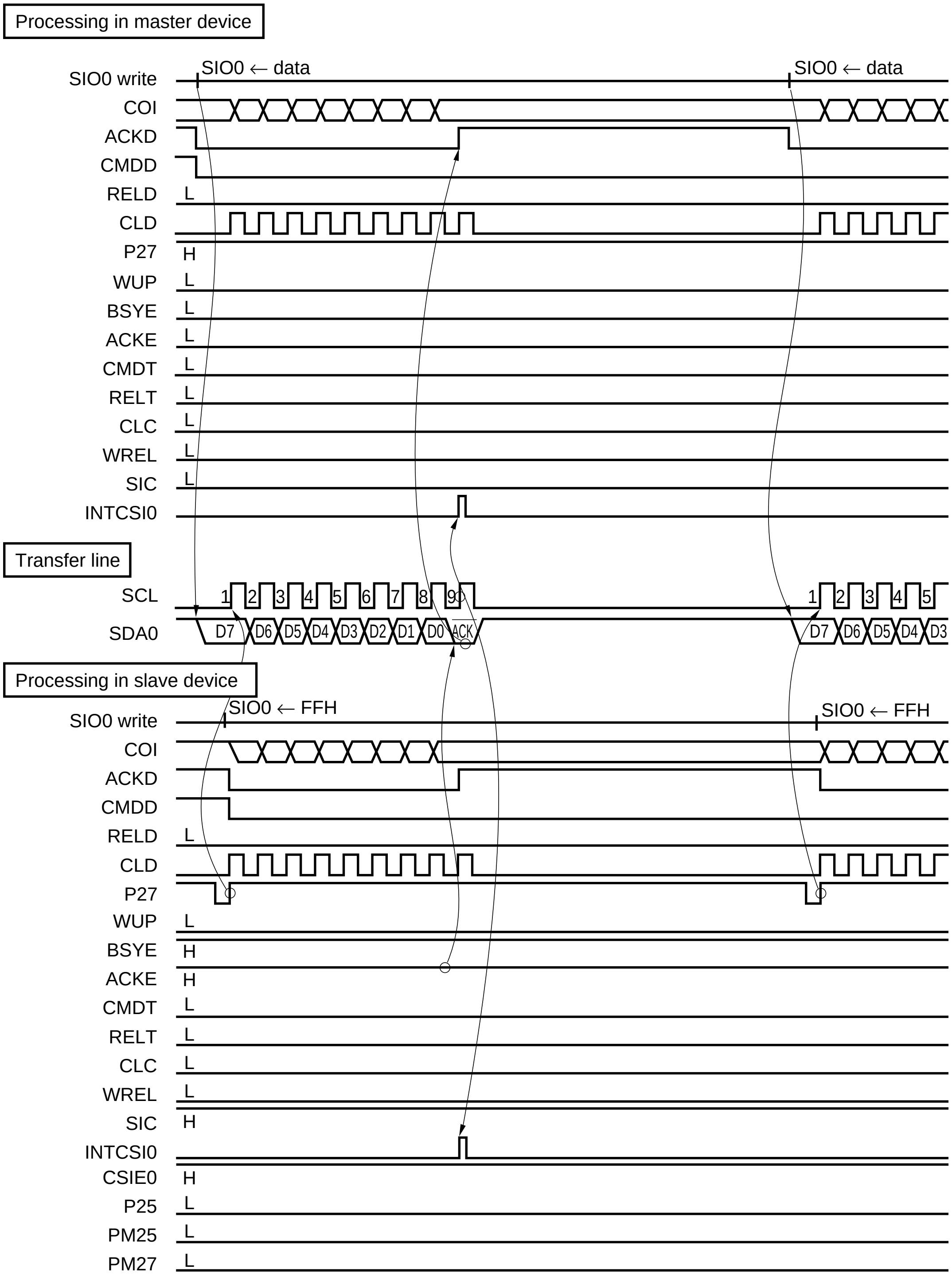


Figure 16-21. Example of Communication from Master to Slave (with 9-clock wait selected for both master and slave) (3/3)

(c) Stop condition

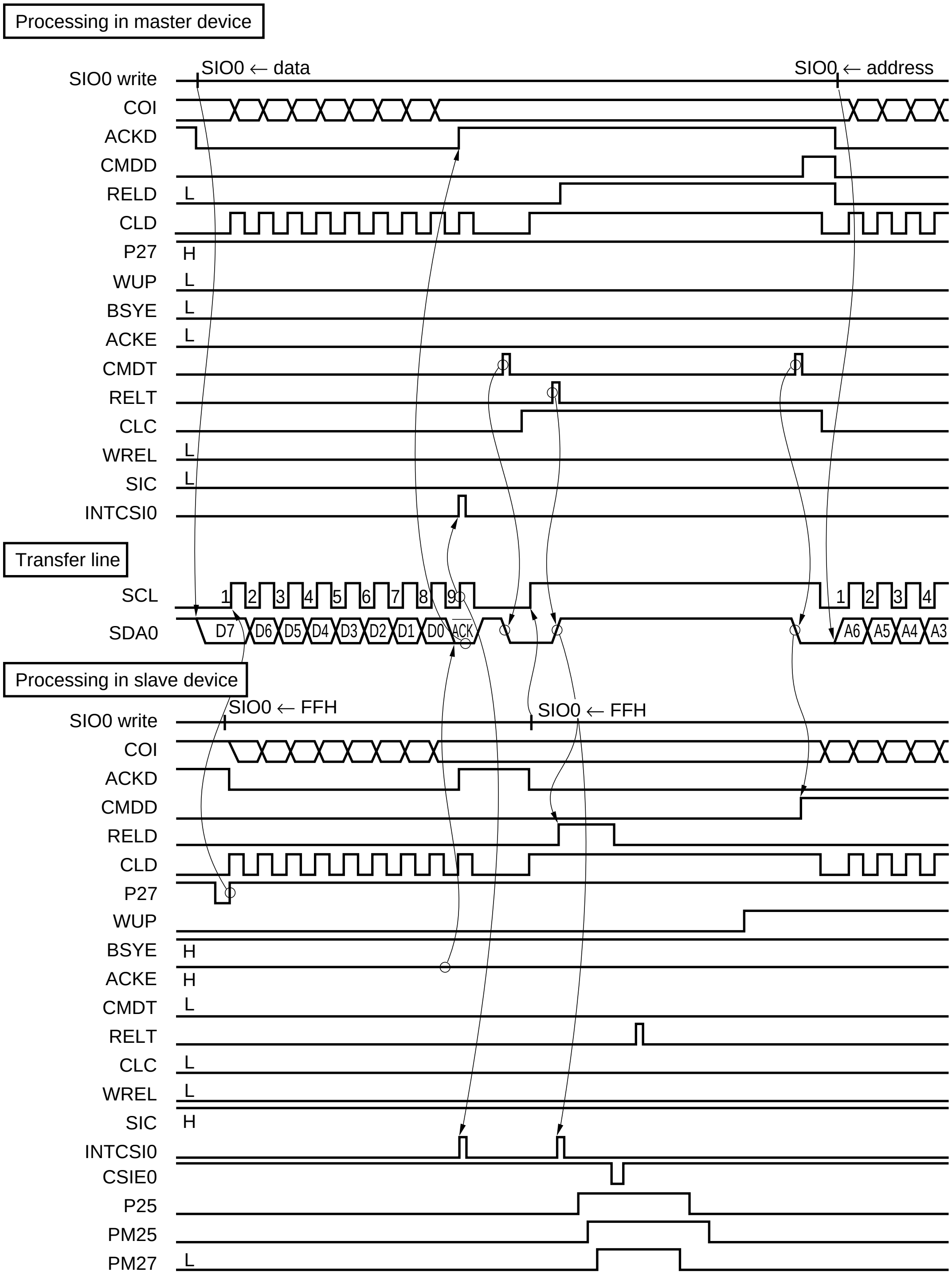


Figure 16-22. Example of Communication from Slave to Master (with 9-clock wait selected for both master and slave) (1/3)

(a) Start condition - address

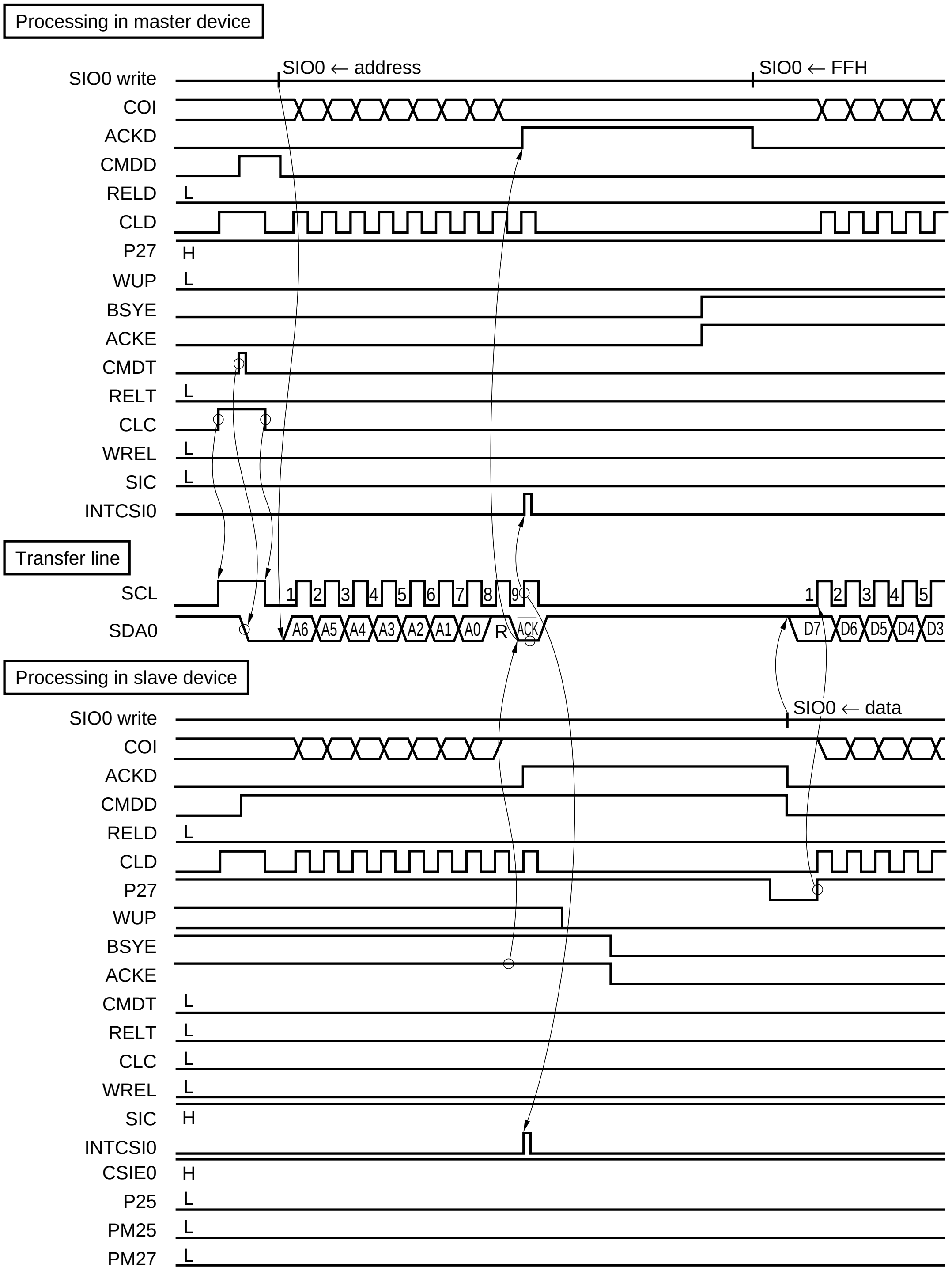


Figure 16-22. Example of Communication from Slave to Master (with 9-clock wait selected for both master and slave) (2/3)

(b) Data

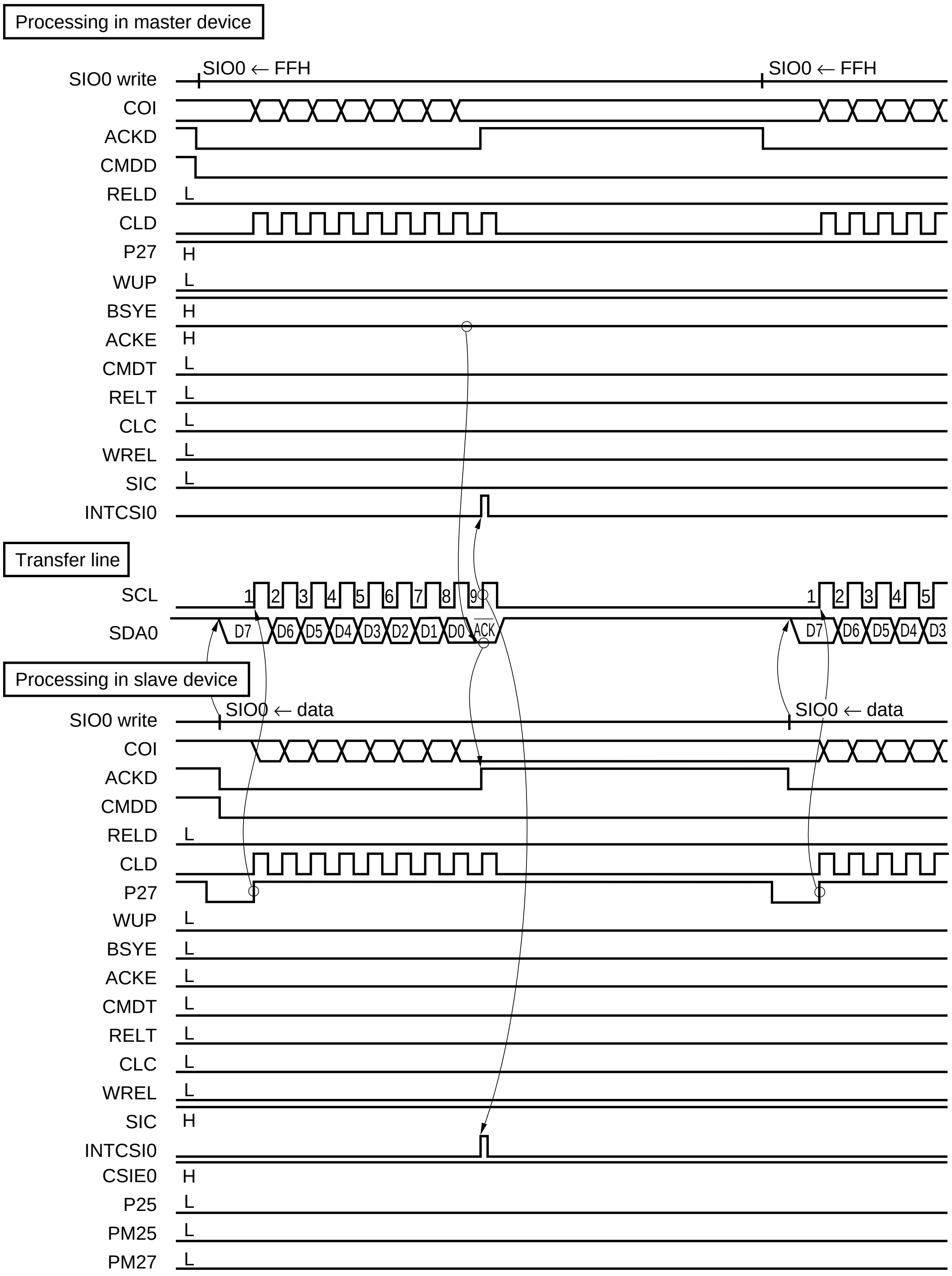
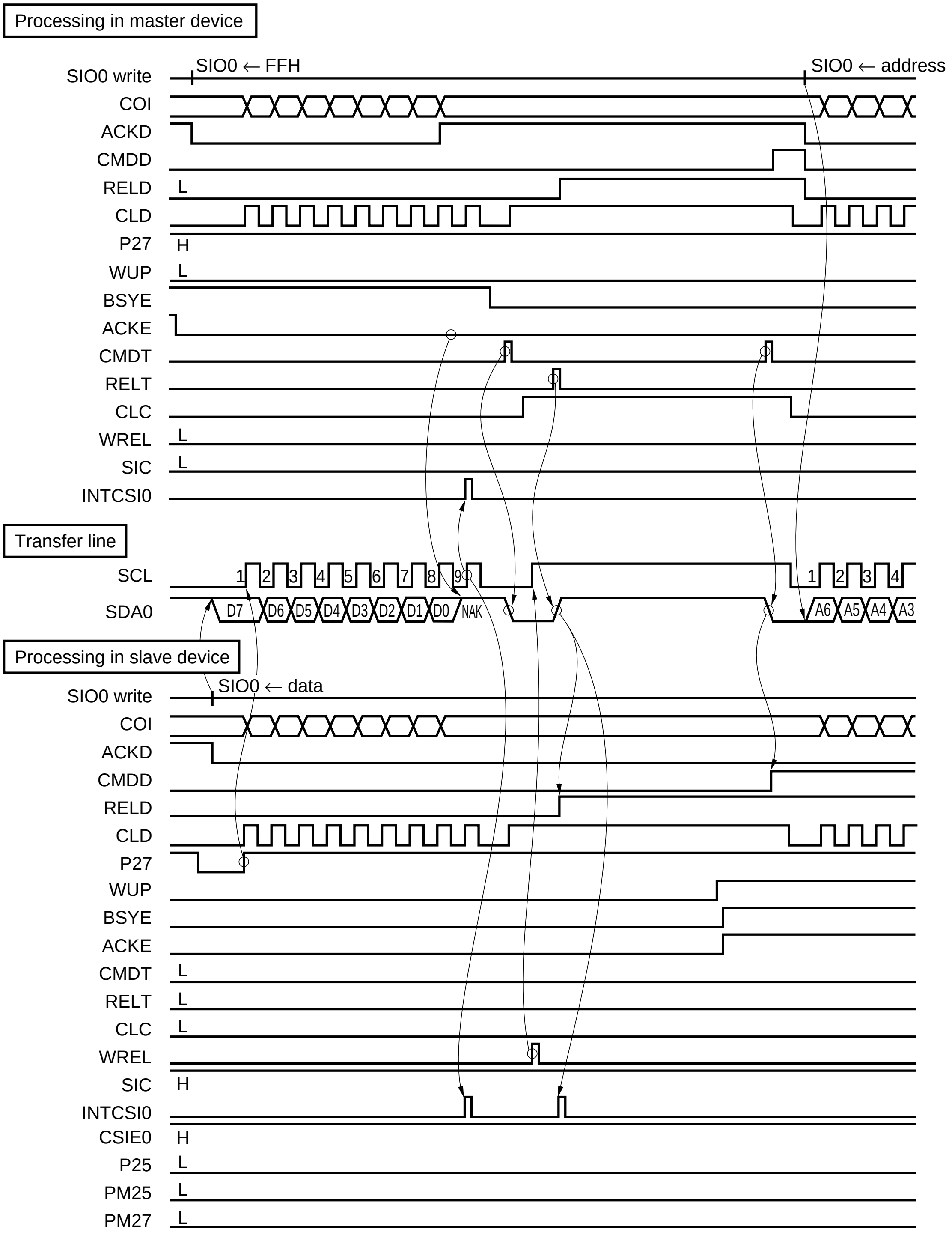


Figure 16-22. Example of Communication from Slave to Master (with 9-clock wait selected for both master and slave) (3/3)

(c) Stop condition



(9) Transfer start

Serial transfer is started when transfer data is placed in SIO0 if the following two conditions are satisfied:

- The serial interface channel 0 operation control bit (CSIE0) = 1
- The internal serial clock is stopped or SCL is low after 8-bit serial data has been transferred

Cautions 1. If CSIE0 is set to “1” after the data has been written to SIO0, the transfer is not started.

2. Because the N-ch open-drain output must be made high-impedance state when data is to be received, set bit 7 (BSYE) of serial bus interface control register (SBIC) to 1 and write FFH to SIO0 in advance.
However, when the wake-up function is used (when bit 5 (WUP) of the serial operation mode register 0 (CSIM0) is set), do not write FFH to SIO0 before reception. The N-ch open-drain output is always at high-impedance state even if FFH is not written to SIO0.

3. If data is written to SIO0 with the slave in the wait status, the data is not lost. Transfer is started when SCL is output after the wait status has been released.

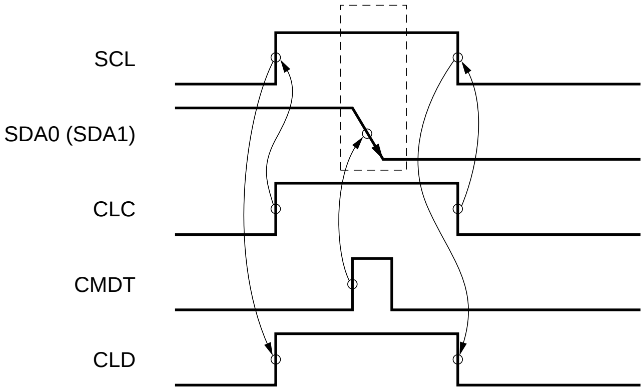
Serial transfer is automatically stopped when 8 bits of data have been completely transferred, and an interrupt request flag (CSIIF0) is set.

16.4.5 Notes on using I^{2}C bus mode

(1) Output of start condition (master)

The SCL pin usually outputs low level when the serial clock is not output. To output the start condition, the SCL pin must be made high once. To make the SCL pin high, set the CLC bit of interrupt timing specification register (SINT) to 1.
After setting CLC, clear CLC to 0 and make the SCL pin low. The serial clock is not output if CLC remains 1.
When the master outputs the start condition or stop condition, make sure that CLD is 1 after CLC has been set to 1. This is because a slave may make SCL low (wait status).

Figure 16-23. Output of Start Condition



(2) Releasing slave from wait status (slave transmission)

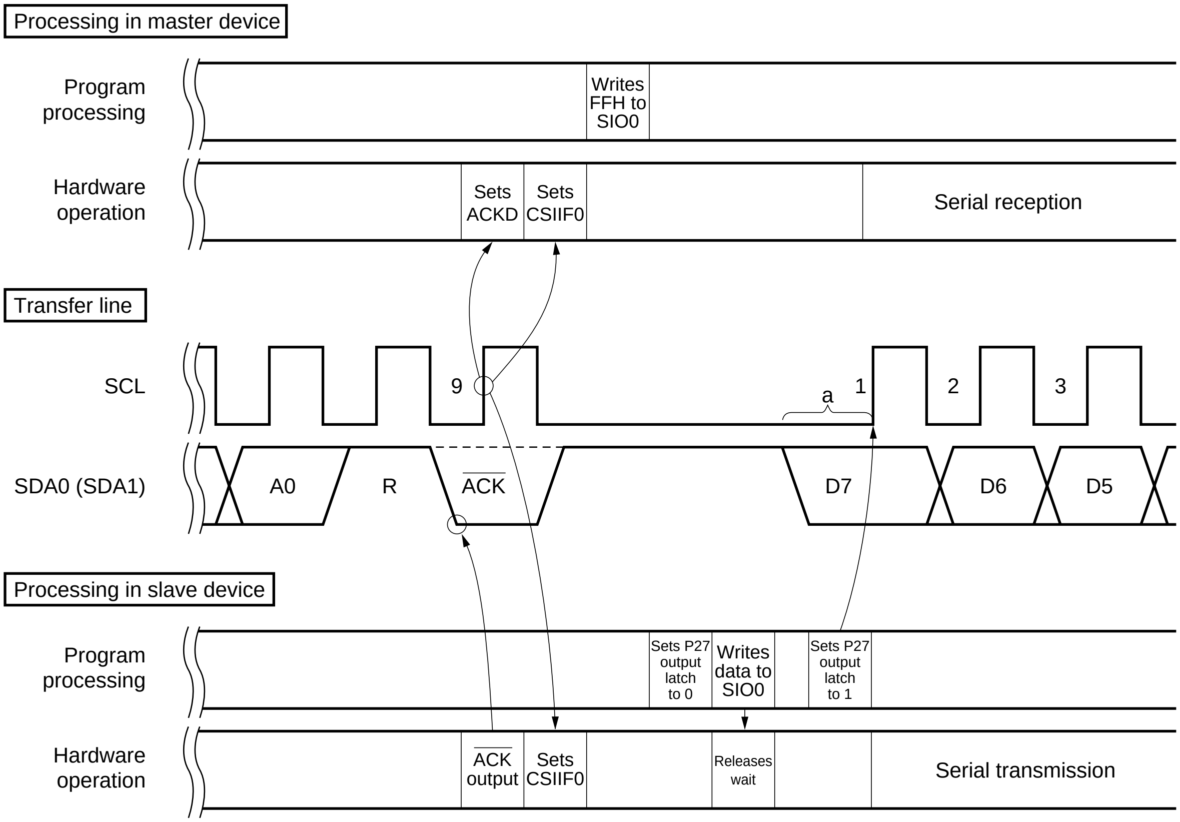
A slave is released from the wait status when the WREL flag (bit 2 of the interrupt timing specification register (SINT)) is set or when an instruction that writes data to serial I/O shift register 0 (SIO0) is executed.

When the slave transmits data, it is immediately released from the wait status when the instruction that writes data to SIO0 is executed, and the clock rises without the first transmit bit output to the data line. It is therefore necessary to transmit the data by manipulating the output latch of P27 by the program as shown in Figure 16-24. At this time, control the low-level width (portion a in Figure 16-24) of the first serial clock with the timing when P27's output latch is set to 1 after the instruction that writes data to SIO0 has been executed.

If the master does not output an acknowledge signal (when data has been transmitted from a slave), set the WREL flag of SINT to 1 to release the slave from the wait status.

For the timing of these operations, refer to Figure 16-22.

Figure 16-24. Releasing Slave from Wait Status (during transmission)



(3) Releasing slave from wait status (slave reception)

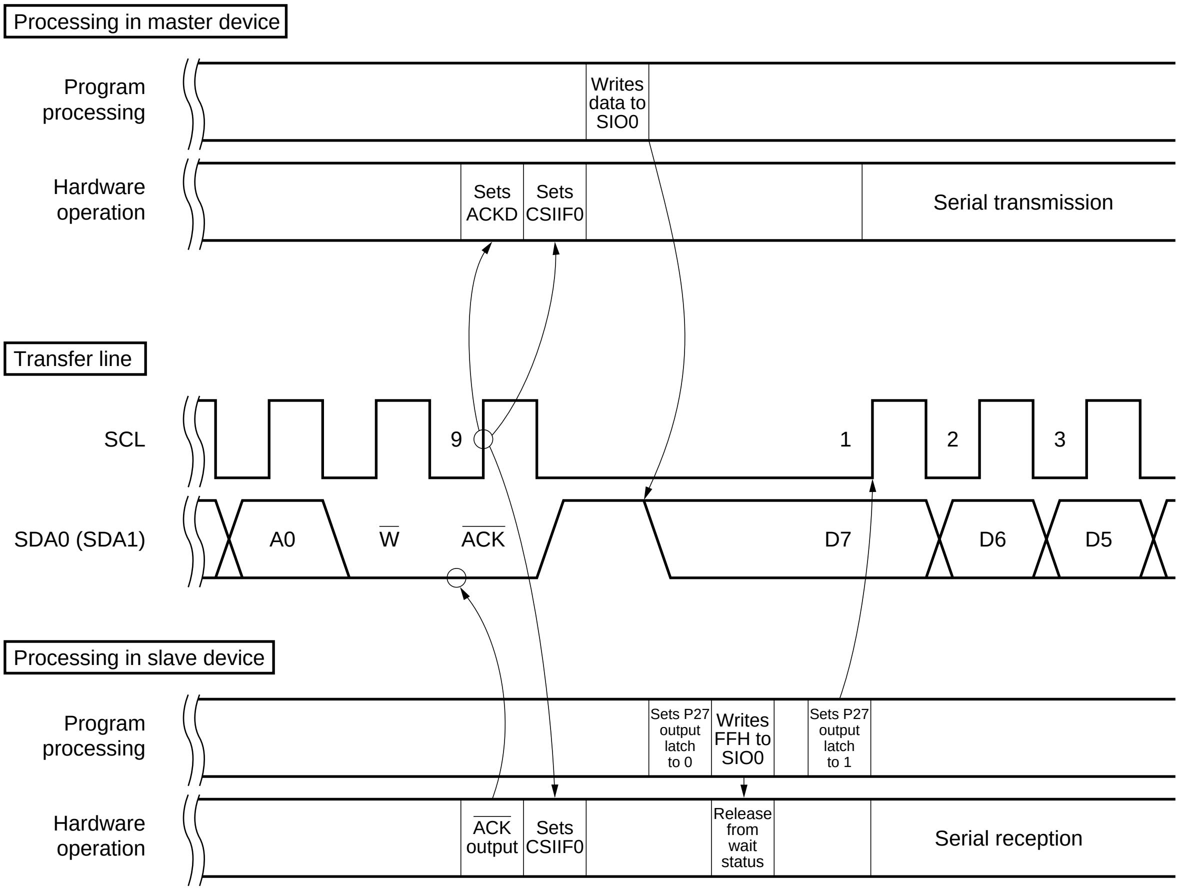
A slave is released from the wait status when the WREL flag (bit 2 of the interrupt timing specification register (SINT)) is set or when an instruction that writes data to serial I/O shift register 0 (SIO0) is executed.

When a slave receives data, if the SCL line goes into a high-impedance state immediately after the instruction that writes data to SIO0 has been executed, the data of the first bit from the master may not be received. This is because SIO does not start its operation if the SCL line is in the high-impedance state while the instruction that writes data to SIO0 is being executed (or, until the next instruction is executed).

Therefore, receive the data by manipulating the output latch of P27 by the program, as shown in Figure 16-25.

For the timing of these operations, refer to Figure 16-21.

Figure 16-25. Releasing Slave from Wait Status (during reception)



(4) Completion processing of reception by slave

Make sure that bit 3 (CMDD) of the serial bus interface control register (SBIC) and bit 6 (COI) of the serial operation mode register 0 (CSIM0) (when CMDD = 1) are checked in the reception completion processing of the slave (interrupt processing). This is to prevent the slave from being unable to identify whether the start condition or data comes first and therefore to prevent the wake-up function from being unusable when a non-specific amount of data is received from the master.

16.4.6 Restrictions in I^{2}C bus mode

The following restrictions apply to the μPD78018FY subseries.

- **Restrictions when used as slave device in I^{2}C bus ode**

Applicable models μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, 78018FY, 78P018FY, and IE-78014-R-EM-A

Description When the wake-up function is executed (by setting the WUP flag (bit 5 of the serial operation mode register 0 (CSIM0)) in the serial transfer status[Note], data between the other slaves and master will be judged as an address. If this data happens to coincide with the slave address of the μPD78018FY subseries, the μPD78018FY subseries will initiate communication, destroying the communication data.

Note The serial transfer status is the status in which the interrupt request flag (CSIIF0) is set because of the end of serial transfer after the serial I/O shift register 0 (SIO0) has been written.

Preventive measure This restriction can be avoided by modifying the program.

Before executing the wake-up function, execute the following program that releases serial transfer status. To execute the wake-up function, do not execute an instruction that writes SIO0. Even if such an instruction is not executed, data can be received when the wake-up function is executed.

This program releases the serial transfer status. To release the serial transfer status, the serial interface channel 0 must be set once in the operation stop status (by clearing the CSIE0 flag (bit 7 of the serial operation mode register (CSIM0) to 0). However, if the serial interface channel 0 is set in the operation stop status in the I^{2}C bus mode, the SCL pin output a high level and the SDA0 (SDA1) pin outputs a low level, affecting communication of the I^{2}C bus. Therefore, this program places the SCL and SDA0 (SDA1) pins in the high-impedance state to prevent the I^{2}C bus from being affected.

In the example below, SDA0 (/P25) is used as a serial data input/output pin. When SDA1 (/P26) is used as the serial data input/output pin, take P2.5 and PM2.5 in the program below as P2.6 and PM2.6, respectively

For the timing of each signal when this program is executed, refer to **Figure 16-21**.

- **Example of program releasing serial transfer status**

```
SET1 P2.5      ; <1>
SET1 PM2.5     ; <2>
SET1 PM2.7     ; <3>
CLR1 CSIE0     ; <4>
SET1 CSIE0     ; <5>
SET1 RELT      ; <6>
CLR1 PM2.7     ; <7>
CLR1 P2.5      ; <8>
CLR1 PM2.5     ; <9>
```

<1> Prevents the SDA0 pin from outputting a low level when the I^{2}C bus mode is restored by the instruction in <5>. The output of the SDA0 pin goes into a high-impedance state.

<2> Sets the P25 (/SDA0) pin in the input mode to prevent the SDA0 line from being affected when the port mode is set by the instruction in <4>. The P25 pin is set in the input mode when the instruction in <2> is executed.

<3> Sets the P27 (/SCL) pin in the input mode to prevent the SCL line from being affected when the port mode is set by the instruction in <4>. The P27 pin is set in the input mode when the instruction in <3> is executed.

<4> Changes the mode from the I^{2}C bus mode to port mode.

<5> Restores the mode from the port mode to the I^{2}C bus mode.

<6> Prevents the instruction in <8> from causing the SDA0 pin to output a low level.

<7> Sets the P27 pin in the output mode because the P27 pin must be in the output mode in the I^{2}C bus mode.

<8> Clears the output latch of the P25 pin to 0 because the output latch of the P25 pin must be cleared to 0 in the I^{2}C bus mode.

<9> Sets the P25 pin in the output mode because the P25 pin must be in the output mode in the I^{2}C bus mode.

Remark RELT: Bit 0 of serial bus interface control register (SBIC)

16.4.7 Manipulating $\overline{\text{SCK0}}$/SCL/P27 pin output

The $\overline{\text{SCK0}}$/SCL/P27 pin can perform static output through software manipulation, in addition to the normal serial clock output.

The value of serial clocks can be set by software (the SIO/SB0/SDA0 and SO0/SB1/SDA1 pins are controlled by the RELT and CMDT bits of the serial bus interface control register (SBIC)).

The following describes how to manipulate the $\overline{\text{SCK0}}$/SCL/P27 pin output.

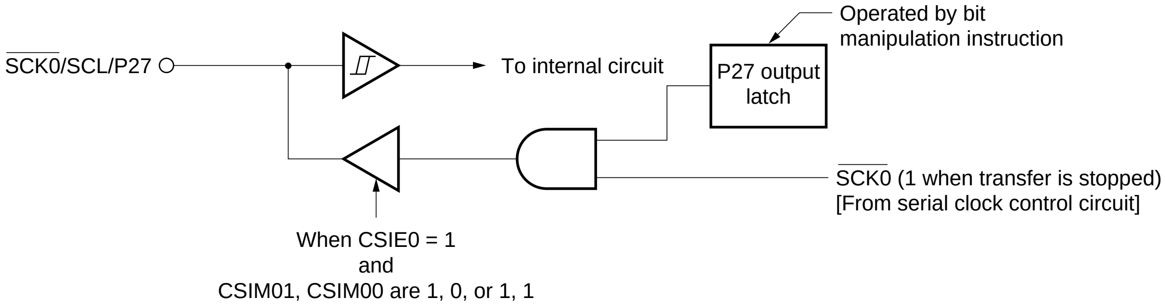
(1) In 3-wire serial I/O mode or 2-wire serial I/O mode

The output level of the $\overline{\text{SCK0}}$/SCL/P27 pin is manipulated by the P27 output latch.

<1> Set serial operation mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin: output mode, serial operation: enabled). $\overline{\text{SCK0}}$ = 1 when serial transfer is stopped.

<2> Manipulate the P27 output latch by using a bit manipulation instruction.

Figure 16-26. Configuration of $\overline{\text{SCK0}}$/SCL/P27 Pin


$\overline{\text{SCK0}}$/SCL/P27
To internal circuit
Operated by bit manipulation instruction
P27 output latch
$\overline{\text{SCK0}}$ (1 when transfer is stopped)
[From serial clock control circuit]
When CSIE0 = 1
and
CSIM01, CSIM00 are 1, 0, or 1, 1

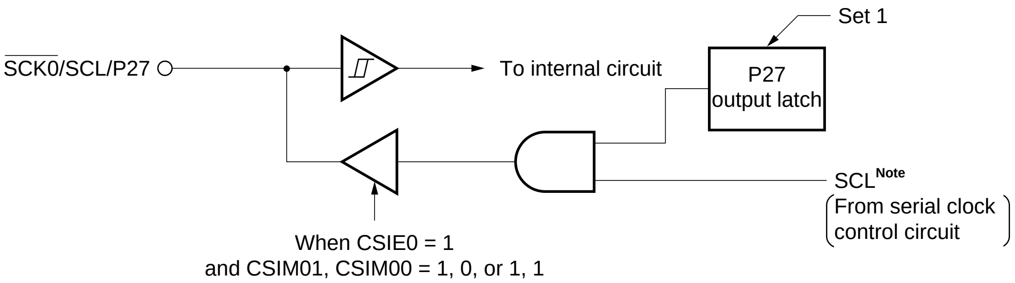
(2) In I^{2}C bus mode

Manipulate the output level of the $\overline{\text{SCK0}}$/SCL/P27 pin by using the CLC bit of the interrupt timing specification register (SINT).

<1> Set serial operation mode register 0 (CSIM0) (SCL pin: output mode, serial operation: enabled).
Put 1 on the P27 output latch. SCL = 0 when serial transfer is stopped.

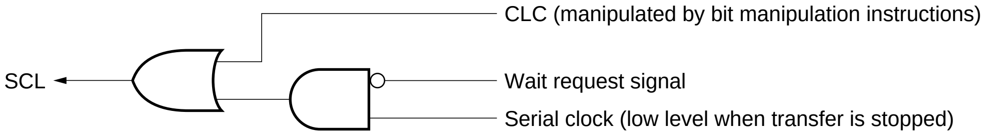
<2> Manipulate the CLC bit of SINT by using a bit manipulation instruction.

Figure 16-27. Configuration of $\overline{\text{SCK0}}$/SCL/P27 Pin



Note The level of the SCL signal is in accordance with the contents of the logic circuit shown in Figure 16-28.

Figure 16-28. Logic Circuit for SCL Signal


CLC (manipulated by bit manipulation instructions)
SCL
Wait request signal
Serial clock (low level when transfer is stopped)

Remarks 1. This figure shows the relations of the signals and does not indicate the internal circuit.

2. CLC: bit 3 of interrupt timing specification register (SINT)

CHAPTER 17 SERIAL INTERFACE CHANNEL 1

17.1 Function of Serial Interface Channel 1

Serial interface channel 1 has the following three modes:

Table 17-1. 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 consumption	–
3-wire serial I/O mode	$\overline{\text{SCK1}}$ (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 transmit/receive function (MSB/LSB first selectable)	$\overline{\text{SCK1}}$ (serial clock) SO1 (serial output) SI1 (serial input)	• This mode has an automatic transmit/receive function as well as the same functions as 3-wire serial I/O mode. • 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.	

17.2 Configuration of Serial Interface Channel 1

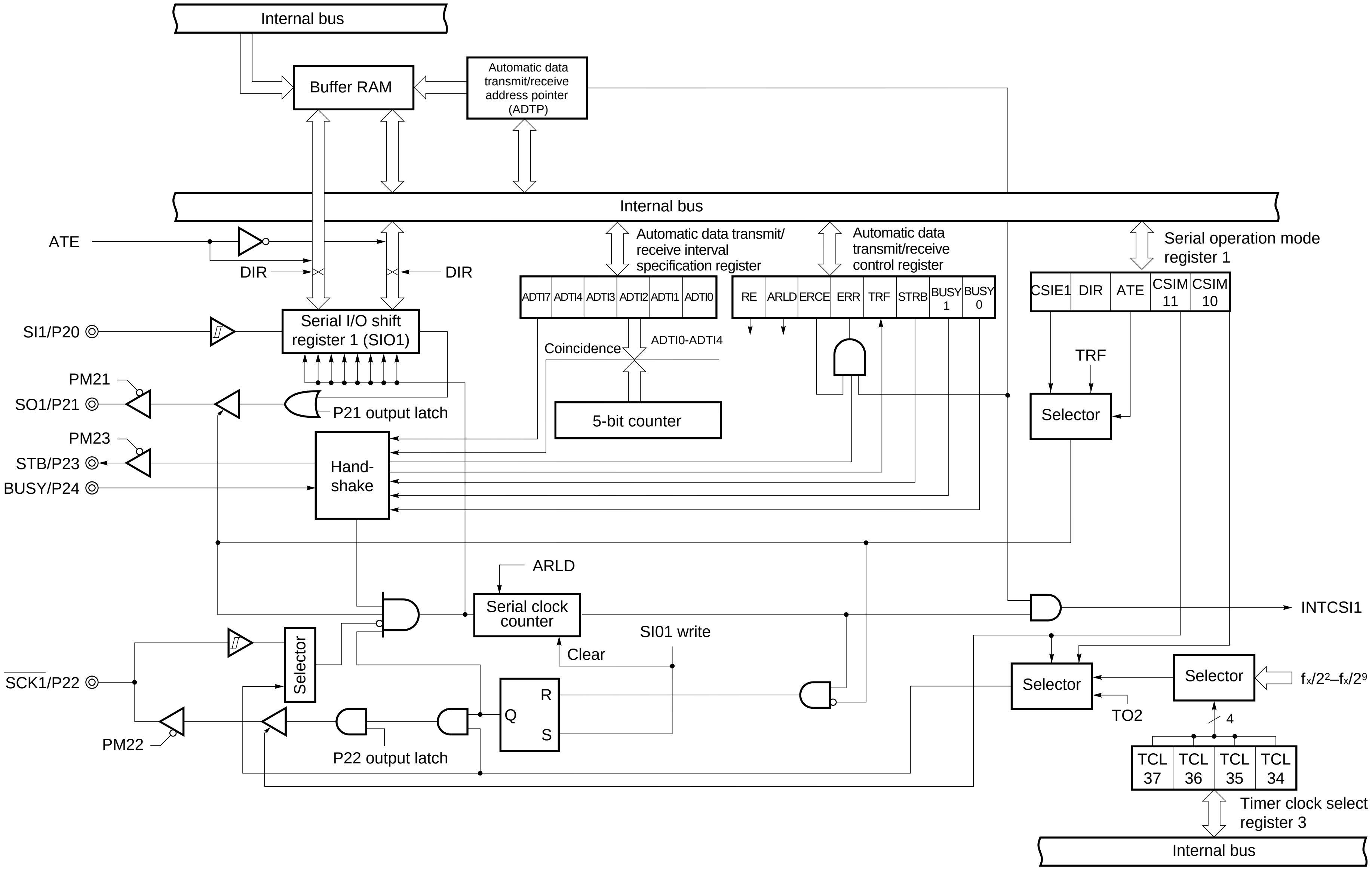
Serial interface channel 1 consists of the following hardware:

Table 17-2. Configuration of Serial Interface Channel 1

Item	Configuration
Register	Serial I/O shift register 1 (SIO1) Automatic data transmit/receive address pointer (ADTP)
Control register	Timer clock select register 3 (TCL3) Serial operation mode register 1 (CSIM1) Automatic data transmit/receive control register (ADTC) Automatic data transmit/receive interval specification register (ADTI) Port mode register 2 (PM2)**Note**

Note Refer to **Figure 6-6, 6-8 Block Diagram of P20, P21, P23-P26** and **Figure 6-7, 6-9 Block Diagram of P22 and P27**.

Figure 17-1. 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 transmit/receive operation is performed.

(2) Automatic data transmit/receive address pointer (ADTP)

This register stores a value of (number of transmission data bytes – 1) when the automatic transmit/receive 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 high-order 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 transmit/receive operation is performed.

(3) Serial clock counter

This counter counts the serial clocks output or input during transmit/receive operation, and checks whether 8-bit serial data has been transmitted/received.

17.3 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 transmit/receive control register (ADTC)
- Automatic data transmit/receive 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 17-2. Format of Timer Clock Select Register 3

Symbol	7	6	5	4	3	2	1	0	Address	On 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 Before writing data other than that already written to TCL3, stop the serial transfer.

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 transmit/receive 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 17-3. Format of Serial Operation Mode Register 1

Symbol	〈7〉	6	〈5〉	4	3	2	1	0	Address	On 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$^{\text{Note 1}}$
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 4-7 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 transmit/receive function

DIR	First bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

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 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$^{\text{Note 3}}$ (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 transmit/receive 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 port

(3) Automatic data transmit/receive control register (ADTC)

This register enables/disables reception of automatic transmission/reception, operation mode, strobe output, busy input, and error check and indicates execution of automatic transmission/reception and error detection.

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 17-4. Format of Automatic Data Transmit/Receive Control Register

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	On 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 transmit/receive 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 transfer/reception is in progress (set to 1 when this bit is written to SIO1)

R

ERR	Error detection of automatic transmit/receive 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 transmit/receive 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 transmit/ receive function
0	Single mode
1	Repeated mode

R/W

RE	Controls reception of automatic transmit/receive 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 determined with TRF instead of CSIIF1 (interrupt request flag).

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 transmit/receive interval specification register (ADTI)

This register sets the interval time at which data is transferred by the automatic transmit/receive 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 17-5. Format of Automatic Data Transmit/Receive Interval Specification Register (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	On 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-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 transmit/receive function is in use.

2. Be sure to set bits 5 and 6 to 0.

3. To control the interval time for data transfer of automatic transmission/reception by using ADTI, busy control becomes invalid (refer to 17.4.3 (4) (a) Busy control option).

Remark f_X : Main system clock oscillation frequency

f_{SCK} : Serial clock frequency

Figure 17-5. Format of Automatic Data Transmit/Receive Interval Specification Register (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	On 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 transmit/receive function is in use.

2. Be sure to set bits 5 and 6 to 0.

3. To control the interval time for data transfer of automatic transmission/reception by using ADTI, busy control becomes invalid (refer to 17.4.3 (4) (a) Busy control option).

Remark f_X : Main system clock oscillation frequency
f_{SCK} : Serial clock frequency

17.4 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 transmit/receive function

17.4.1 Operation stop mode

Serial transfer is not executed in this mode. Consequently, the power consumption 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	On 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	SI1Note 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 transmit/receive control register (ADCT) to 0.

Remark	
×	: Don't care
PM××	: Port mode register
P××	: Output latch of port

17.4.2 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 ($\overline{\text{SCK1}}$), 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 $\overline{\text{RESET}}$ signal is input.

Symbol	⟨7⟩	6	⟨5⟩	4	3	2	1	0	Address	On 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$^{\text{Note 1}}$
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 4-7 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 transmit/receive function

DIR	First bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

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 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$^{\text{Note 3}}$ (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 transmit/receive 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 port

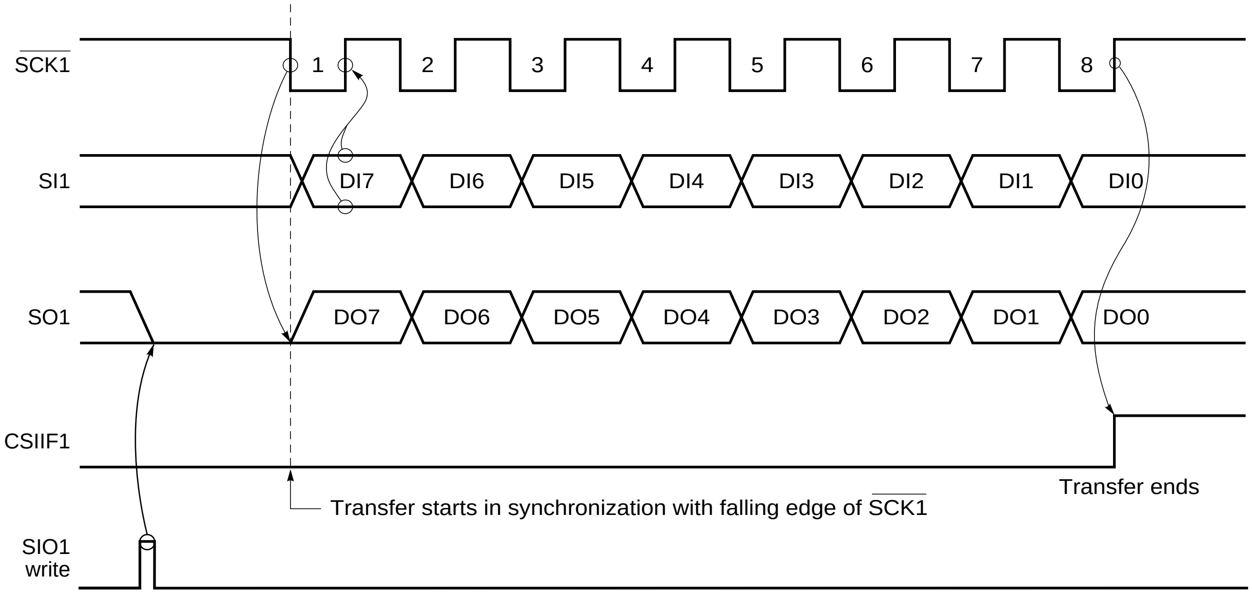
(2) Communication operation

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 ($\overline{\text{SCK1}}$). 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 $\overline{\text{SCK1}}$.

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 17-6. Timing of 3-Wire Serial I/O Mode



Caution The SIO1 pin goes low when SO1 is written.

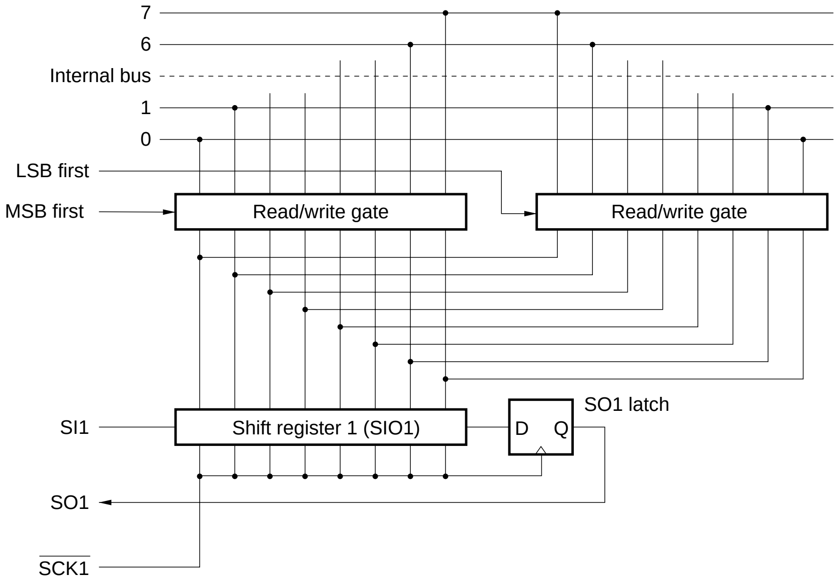
(3) Selecting MSB/LSB first

In the 3-wire serial I/O mode, a function to select whether data is transferred with its MSB or LSB first can be used.

Figure 17-7 shows the configuration of the serial I/O shift register 1 (SIO1) and internal bus. As shown in the figure, data can be read or written by inverting the MSB or LSB.

Whether data is transferred with the MSB or LSB first can be specified by using bit 6 (DIR) of the serial operation mode register 1 (CSIM1).

Figure 17-7. Transfer Bit Sequence Select Circuit



The first bit to be transferred is selected by changing the bit sequence in which data is written to SIO1. The shift sequence of SIO1 is unchanged.

Therefore, select the first bit to be transferred (MSB or LSB) before writing data to the shift register.

(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.

17.4.3 Operation in 3-wire serial I/O mode with automatic transmit/receive 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 transmit/receive function is set by using the serial operation mode register 1 (CSIM1) and automatic data transmit/receive control register (ADTC) and automatic data transmit/receive 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	On 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$^{\text{Note 1}}$
1	0	Output of 8-bit timer register 2 (TM2)
1	1	Clock specified by bits 4-7 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 transmit/receive function

DIR	First bit	SI1 pin function	SO1 pin function
0	MSB	SI1/P20 (input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM 11	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$^{\text{Note 3}}$ (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 transmit/receive 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 port

(b) Automatic data transmit/receive 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	On 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 transmit/receive 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 transmit/receive 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 transmit/receive 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 transmit/receive function
0	Single mode
1	Repeat mode

R/W

RE	Controls reception of automatic transmit/receive 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 determined with TRF, instead of CSIIF1 (interrupt request flag).

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 transmit/receive interval specification register (ADTI)

This register sets the interval time at which data is transferred by the automatic transmit/receive 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	On 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-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 transmit/receive function is in use.

2. Be sure to set bits 5 and 6 to 0.

3. To control the interval time for data transfer of automatic transmission/reception by using ADTI, busy control becomes invalid (refer to 17.4.3 (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	On 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 transmit/receive function is in use.

2. Be sure to set bits 5 and 6 to 0.

3. To control the interval time for data transfer of automatic transmission/reception by using ADTI, busy control becomes invalid (refer to 17.4.3 (4) (a) Busy control option).

Remark f_X : Main system clock oscillation frequency
f_{SCK} : Serial clock frequency

(2) Setting of automatic transmit/receive data

(a) Setting of transmit data

<1> Write the transmit data from the lowest address FAC0H of the buffer RAM (up to FADFH). However, the data must be transmitted from the high-order address to the low-order address.

<2> Set the value of the number of transmit data bytes minus 1 to the automatic data transmit/receive address pointer (ADTP).

(b) Setting of automatic transmit/receive mode

<1> Set CSIE1 and ATE of the serial operation mode register 1 (CSIM1) to 1.

<2> Set RE of the automatic transmit/receive control register (ADTC) to 1.

<3> Set a data transmit/receive interval to the automatic data transmit/receive 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 transmit/receive operation, and the written value has no meaning.

The following operation is automatically executed by performing (a) and (b) above.

- After the data in the buffer RAM specified by ADTP has been transferred to SIO1, transfer is executed (start of the automatic transmit/receive 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 transmit/receive operation).
- When the automatic transmit/receive oeration is completed, TRF is cleared to 0.

(3) Communication operation

(a) Basic transmit/receive 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.

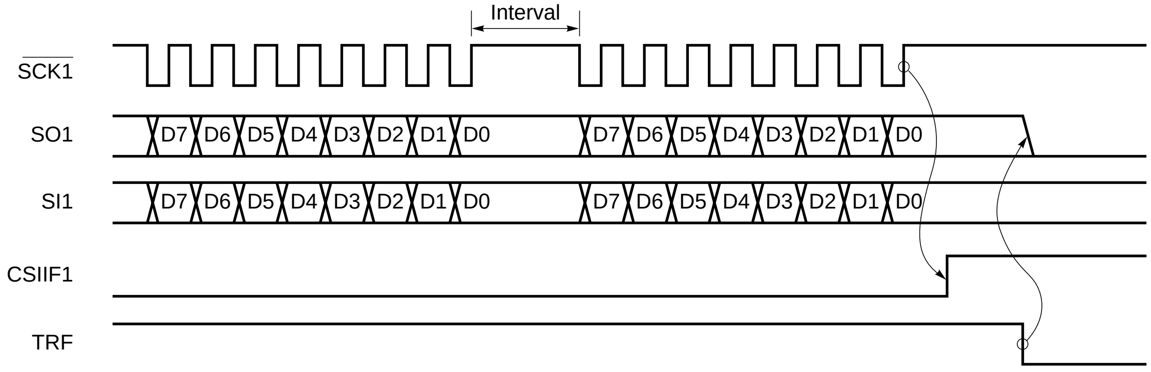
Serial transfer is started by writing any data to the serial I/O shift register 1 (SIO1) when the bit 7 (CSIE1) of the serial operation mode register 1 (CSIM1) is set to 1.

When the last byte has been completely transmitted, an interrupt request flag (CSIIF1) is set. Note, however, that the completion of automatic transmission/reception should be determined with the bit 3 (TRF) of the automatic transmit/receive control register (ADTC), instead of CSIIF1.

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 17-8 shows the operation timing of the basic transmit/receive mode, and Figure 17-9 shows an operation flowchart. The buffer RAM operation when six bytes are transmitted/received is shown in Figure 17-10.

Figure 17-8. Operation Timing of Basic Transmit/Receive Mode

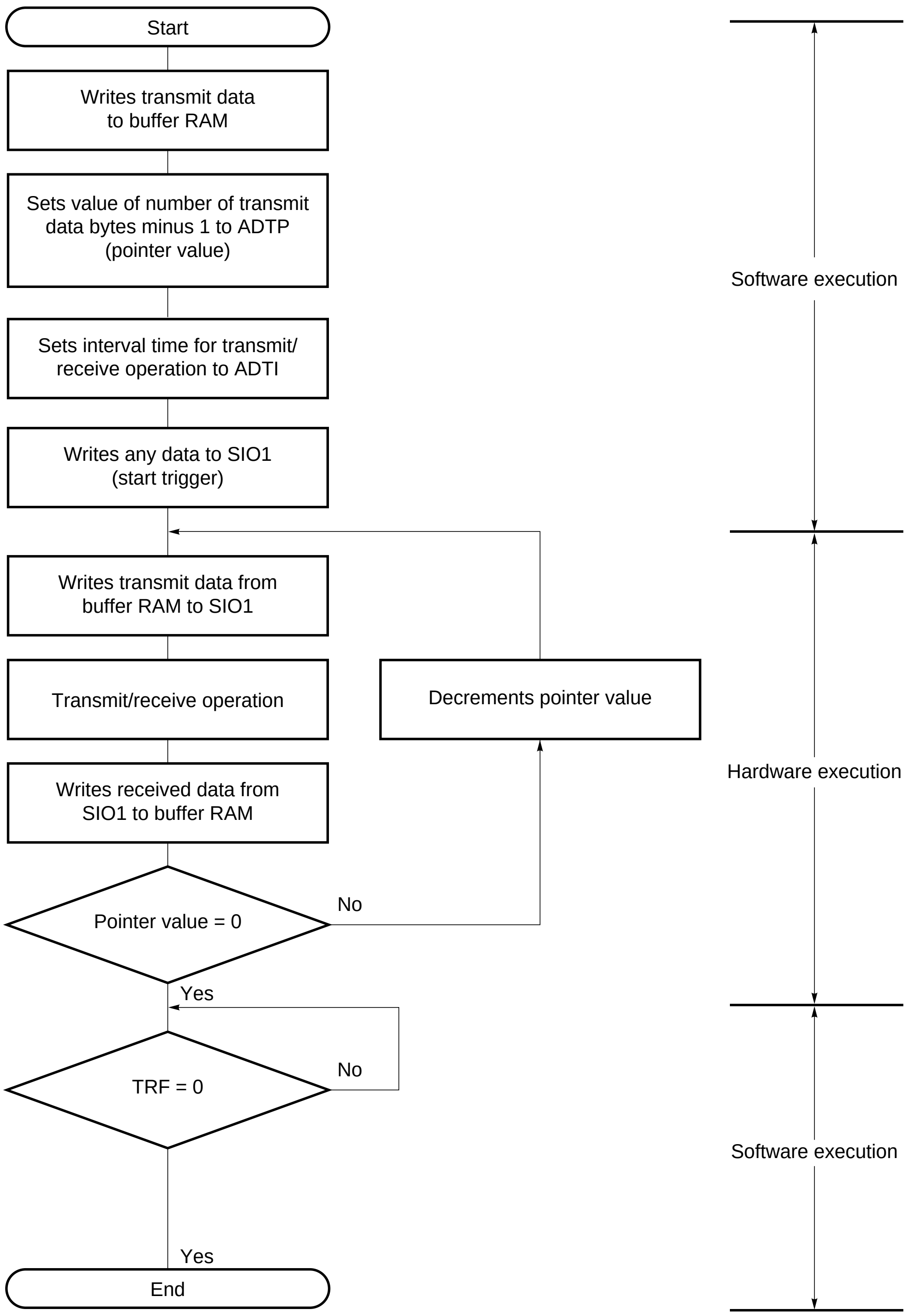


Cautions 1. In the basic transmit/receive mode, the buffer RAM is written/read after 1-byte data has been transmitted/received. Therefore, there is interval time until the next transmission/reception is executed. Because the buffer RAM is written/read simultaneously with the CPU processing, the maximum interval time depends on the CPU processing and a value of the automatic data transmit/receive 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 the automatic transmit/receive control register (ADTC)

Figure 17-9. Flowchart of Basic Transmit/Receive Mode


Start
Writes transmit data to buffer RAM
Sets value of number of transmit data bytes minus 1 to ADTP (pointer value)
Sets interval time for transmit/ receive operation to ADTI
Writes any data to SIO1 (start trigger)
Writes transmit data from buffer RAM to SIO1
Transmit/receive operation
Decrements pointer value
Writes received data from SIO1 to buffer RAM
Pointer value = 0
No
Yes
TRF = 0
No
Yes
End
Software execution
Hardware execution
Software execution

ADTP : Automatic data transmit/receive address pointer
ADTI : Automatic data transmit/receive interval specification register
SIO1 : Serial I/O shift register 1
TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

The buffer RAM operates as follows when 6 bytes are transmitted/received in the basic transmit/receive mode (ARLD = 0, RE = 1).

(i) Before transmission/reception (Refer to Figure 17-10 (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 transmit/receive address pointer (ADTP) is decremented. Subsequently, transmit data 2 (T2) is transferred from buffer RAM to SIO1.

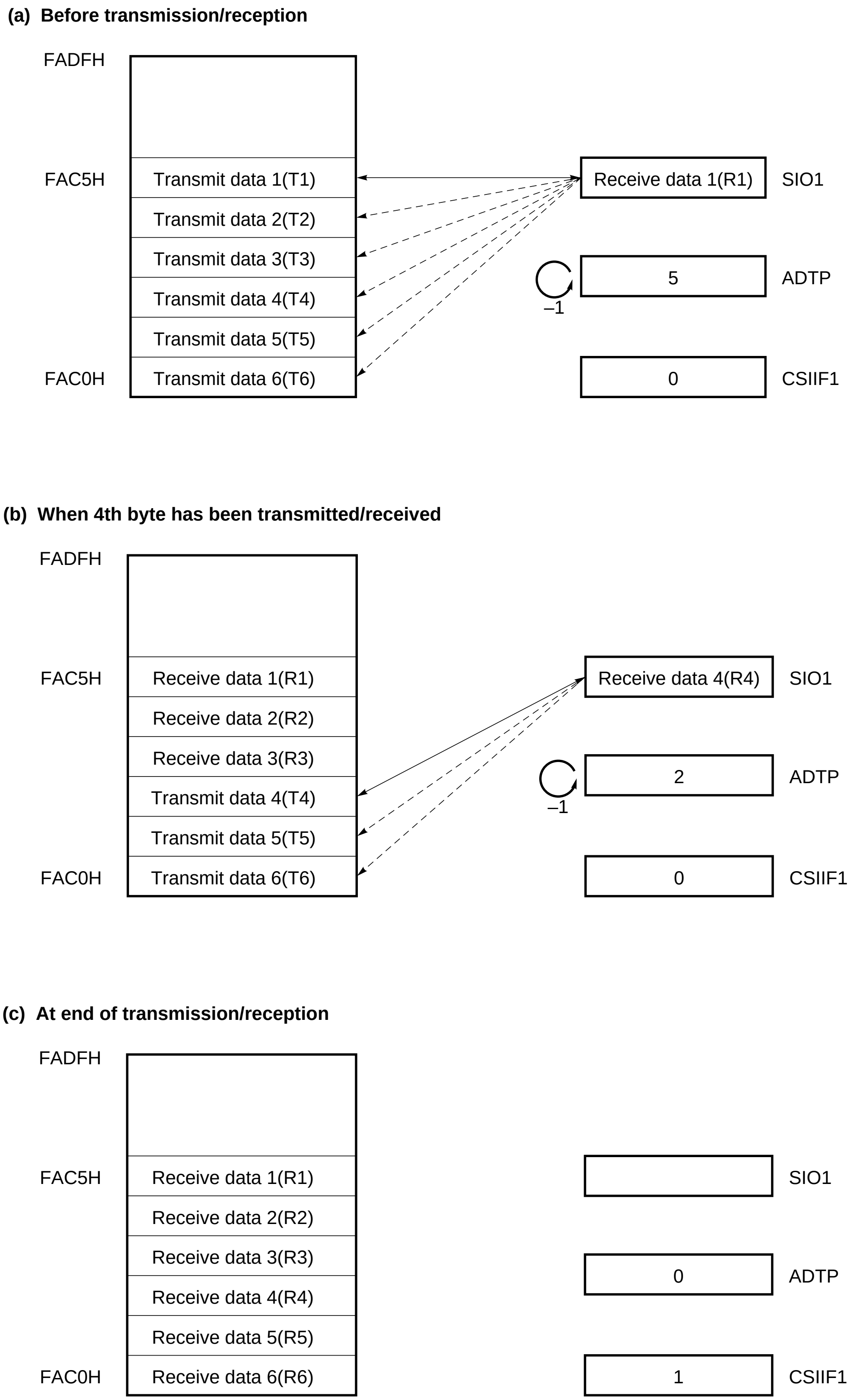
(ii) When 4th byte is transmitted/received (Refer to Figure 17-10 (b))

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 17-10 (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 17-10. Buffer RAM Operation when 6 Bytes Are Transmitted/Received (in basic transmit/receive mode)



(a) Before transmission/reception

(b) When 4th byte has been transmitted/received

(c) At end of transmission/reception

(b) Basic transmit 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.

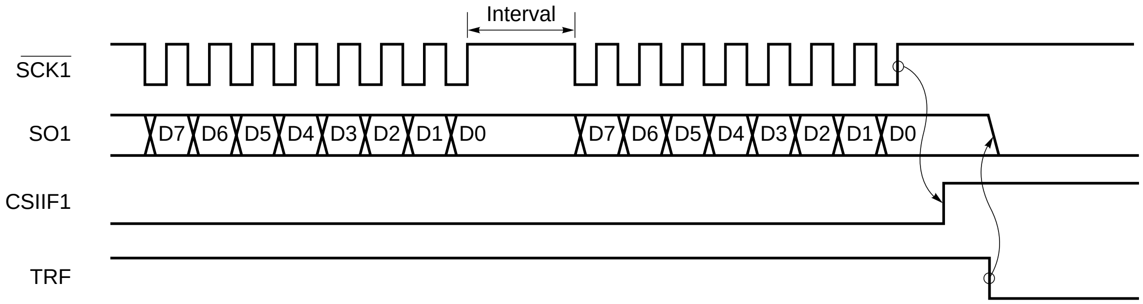
The interrupt request flag (CSIIF1) is set when the last byte has been completely transmitted. Note, however, that the completion of automatic transmission/reception should be determined with the bit 3 (TRF) of the automatic transmit/receive control register (ADTC), instead of CSIIF1.

When receive 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 17-11 shows the operation timing of the basic transmit mode, and Figure 17-12 shows an operation flowchart.

Figure 17-13 shows the operation of the buffer RAM when 6 bytes are transmitted.

Figure 17-11. Operation Timing of Basic Transmit Mode

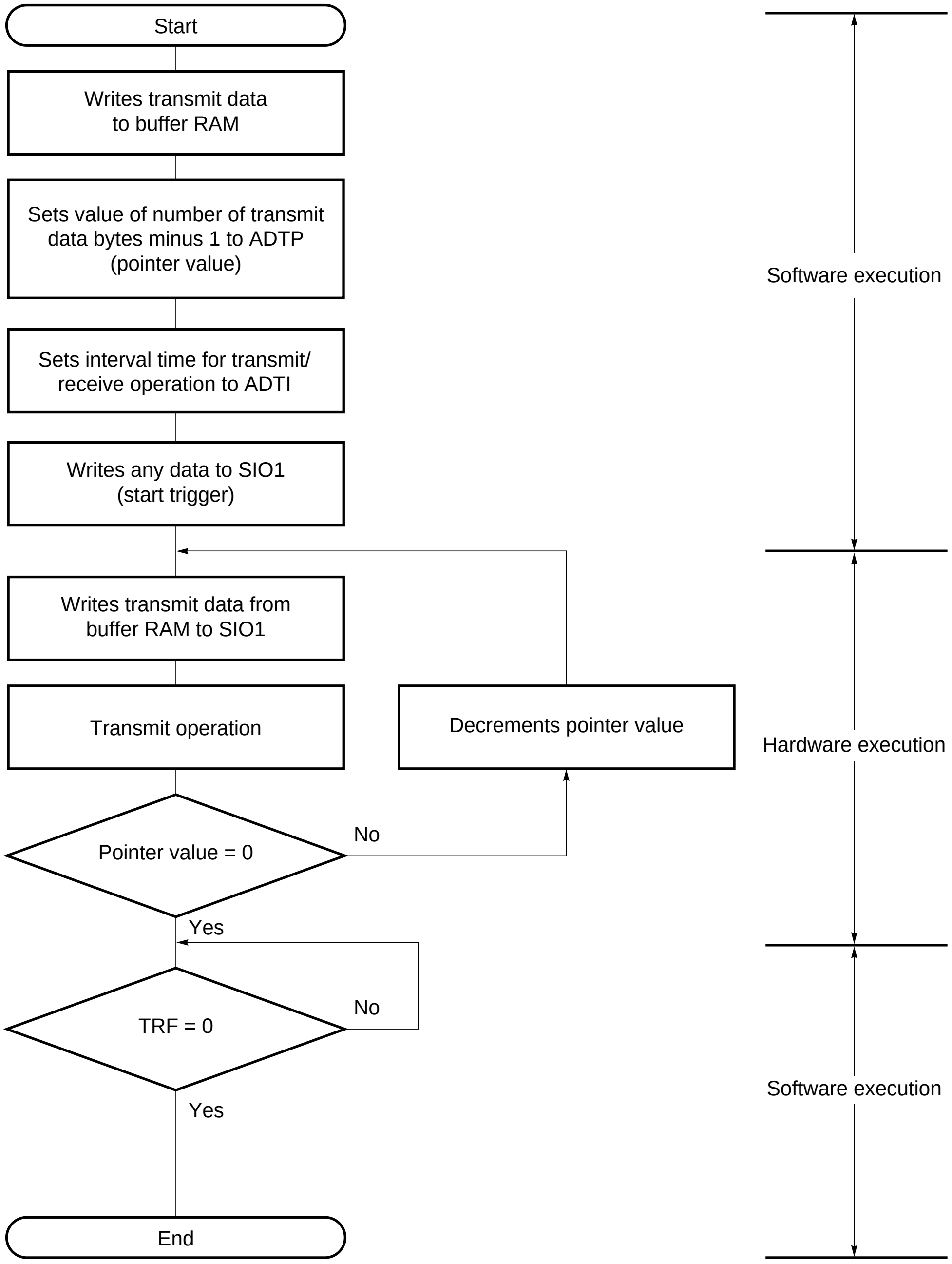


Cautions 1. In the basic transmit mode, the buffer RAM is read after 1-byte data has been transmitted. Therefore, there is an interval time until the next transmission is executed. Because the buffer RAM is read simultaneously with the CPU processing, the maximum interval time depends on the CPU processing 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 the automatic transmit/receive control register (ADTC)

Figure 17-12. Flowchart of Basic Transmit Mode


Start
Writes transmit data to buffer RAM
Sets value of number of transmit data bytes minus 1 to ADTP (pointer value)
Sets interval time for transmit/ receive operation to ADTI
Writes any data to SIO1 (start trigger)
Writes transmit data from buffer RAM to SIO1
Transmit operation
Decrements pointer value
Pointer value = 0
No
Yes
TRF = 0
No
Yes
End
Software execution
Hardware execution
Software execution

ADTP	:	Automatic data transmit/receive address pointer
ADTI	:	Automatic data transmit/receive interval specification register
SIO1	:	Serial I/O shift register 1
TRF	:	Bit 3 of automatic data transmit/receive control register (ADTC)

The buffer RAM operates as follows when 6 bytes are transmitted in the basic transmit mode (ARLD = 0, RE = 0).

(i) Before transmission (Refer to Figure 17-13 (a))

Transfer 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 transferred, automatic data transmit/receive address pointer (ADTP) is decremented. Subsequently, transfer data 2 (T2) is transferred from buffer RAM to SIO1.

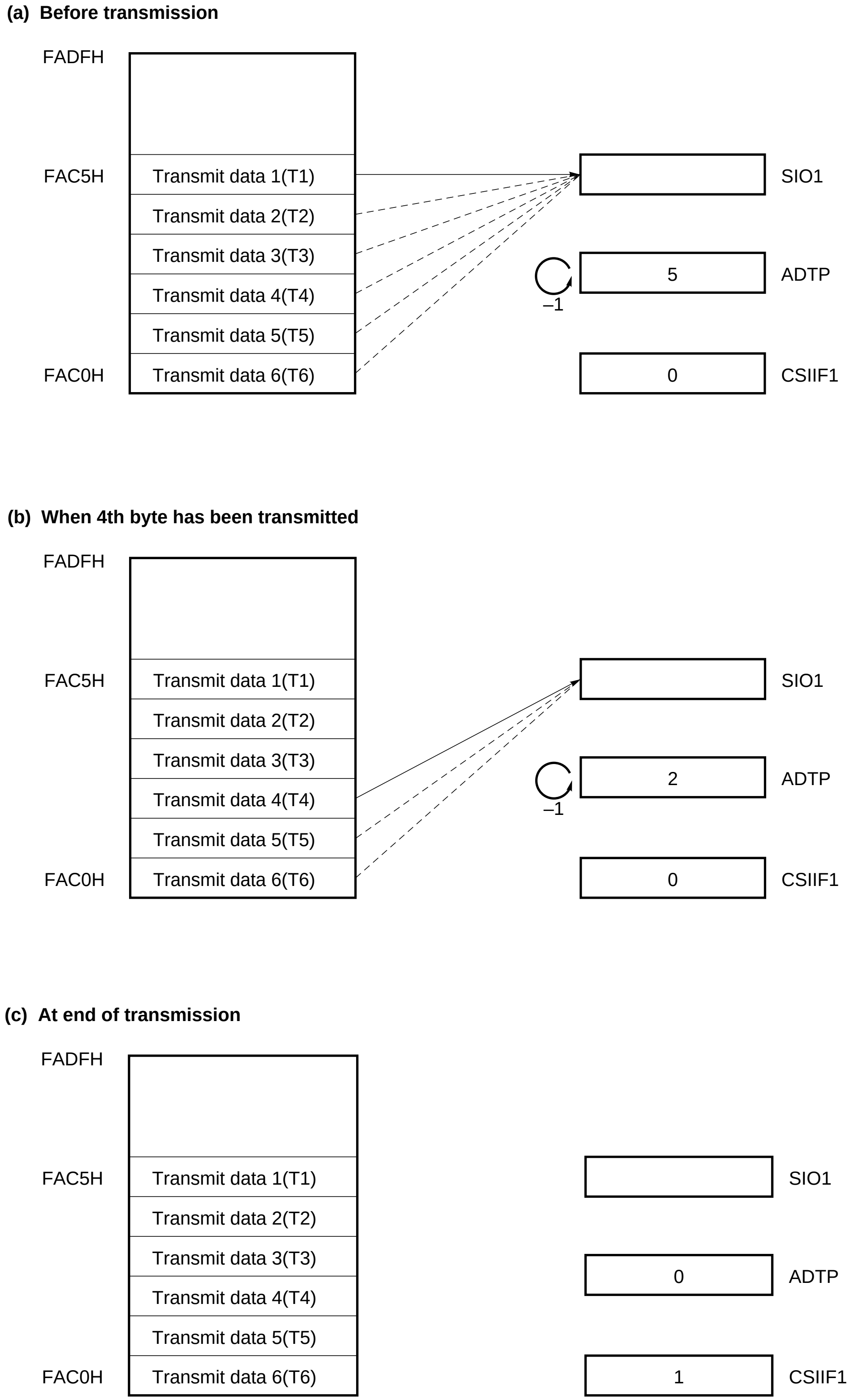
(ii) When 4th byte is transmitted (Refer to Figure 17-13 (b))

When the third byte has been transmitted completely, transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When the fourth byte has been transmitted, and ADTP is decremented.

(iii) End of transmission (Refer to Figure 17-13 (c))

When the sixth byte has been transmitted, an interrupt request flag (CSIIF1) is set (INTCSI1 occurs).

Figure 17-13. Buffer RAM Operation when 6 Bytes Are Transmitted (in basic transmit mode)


(a) Before transmission

(b) When 4th byte has been transmitted

(c) At end of transmission

(c) Repetitive transmit mode

This mode is to repeatedly transmit the data stored in the 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.

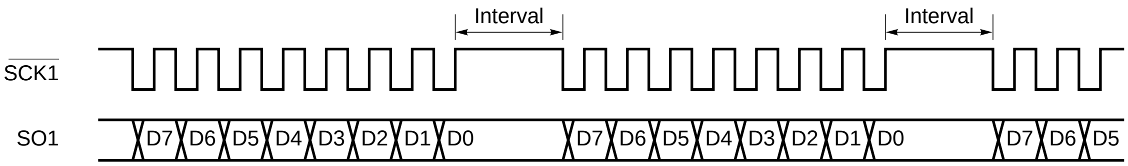
Unlike the basic transmit mode, the interrupt request flag (CSIIF1) is not set after the last byte (data at address FAC0H) has been transmitted, the value at which the transmission/reception has been started is set again to the automatic data transmit/receive address pointer (ADTP), and the contents of the buffer RAM are transmitted again.

When receive 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 17-14 shows the operation timing of the repetitive transmit mode, and Figure 17-15 shows an operation flowchart.

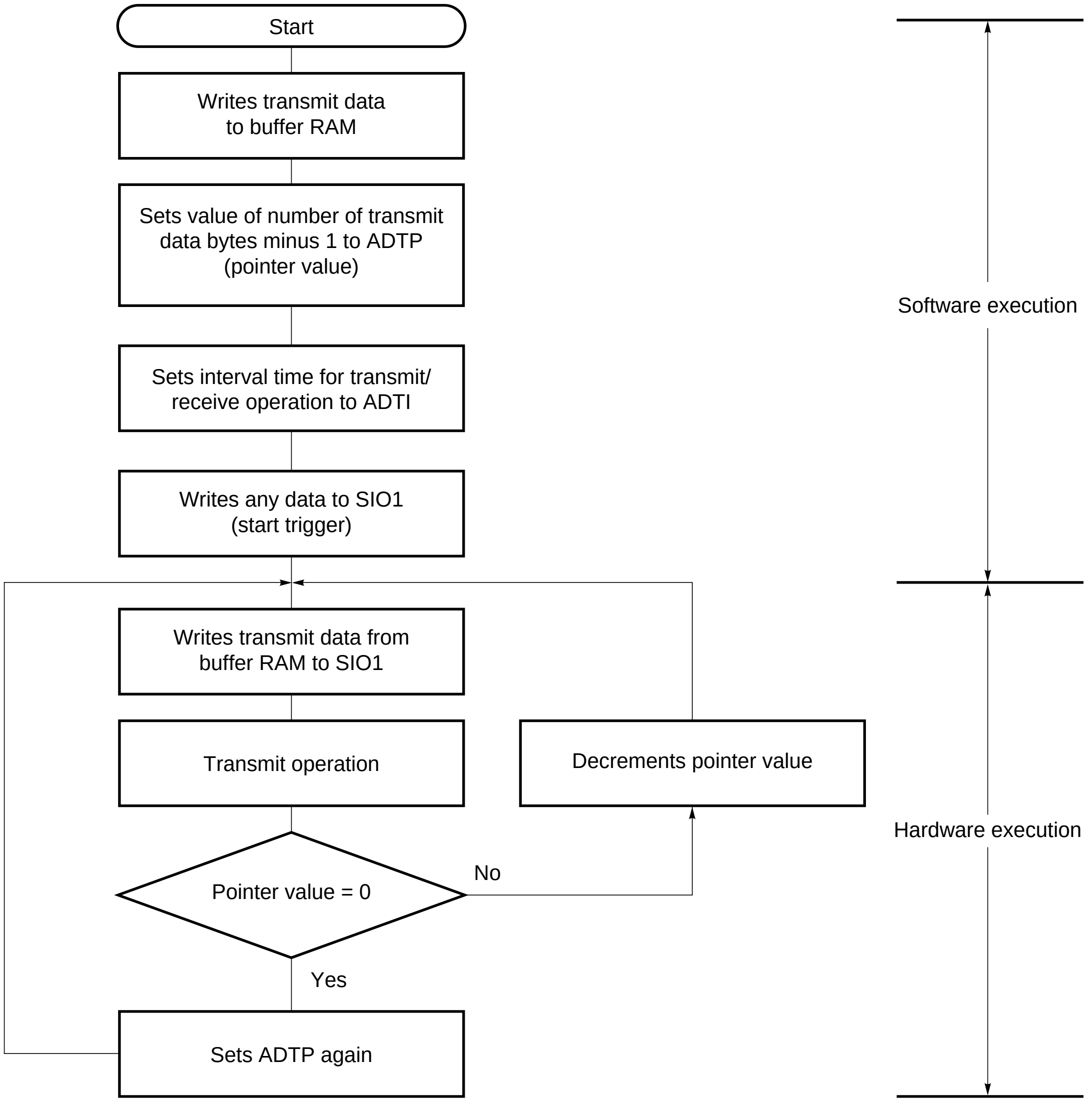
And, Figure 17-16 shows the operation of the buffer RAM when 6 bytes are transmitted in the repetitive transmit mode.

Figure 17-14. Operation Timing of Repetitive Transmit Mode



Caution In the repetitive transmit mode, the buffer RAM is read after 1-byte data has been transmitted. Therefore, there is interval time until the next transmission is executed. Because the 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 transmit/receive time interval specification register (ADTI) (refer to (5) Interval time of automatic transmission/reception).

Figure 17-15. Flowchart of Repetitive Transmit Mode


Start
Writes transmit data to buffer RAM
Sets value of number of transmit data bytes minus 1 to ADTP (pointer value)
Sets interval time for transmit/ receive operation to ADTI
Writes any data to SIO1 (start trigger)
Writes transmit data from buffer RAM to SIO1
Transmit operation
Decrements pointer value
Pointer value = 0
No
Yes
Sets ADTP again
Software execution
Hardware execution

ADTP	:	Automatic data transmit/receive address pointer
ADTI	:	Automatic data transmit/receive interval specification register
SIO1	:	Serial I/O shift register 1

The buffer RAM operates as follows when 6 bytes are transmitted in the repetitive transmit mode (ARLD = 1, RE = 0).

(i) Before transmission (Refer to Figure 17-16 (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, automatic data transmit/receive address pointer (ADTP) is decremented. Subsequently, transmit data 2 (T2) is transferred from buffer RAM to SIO1.

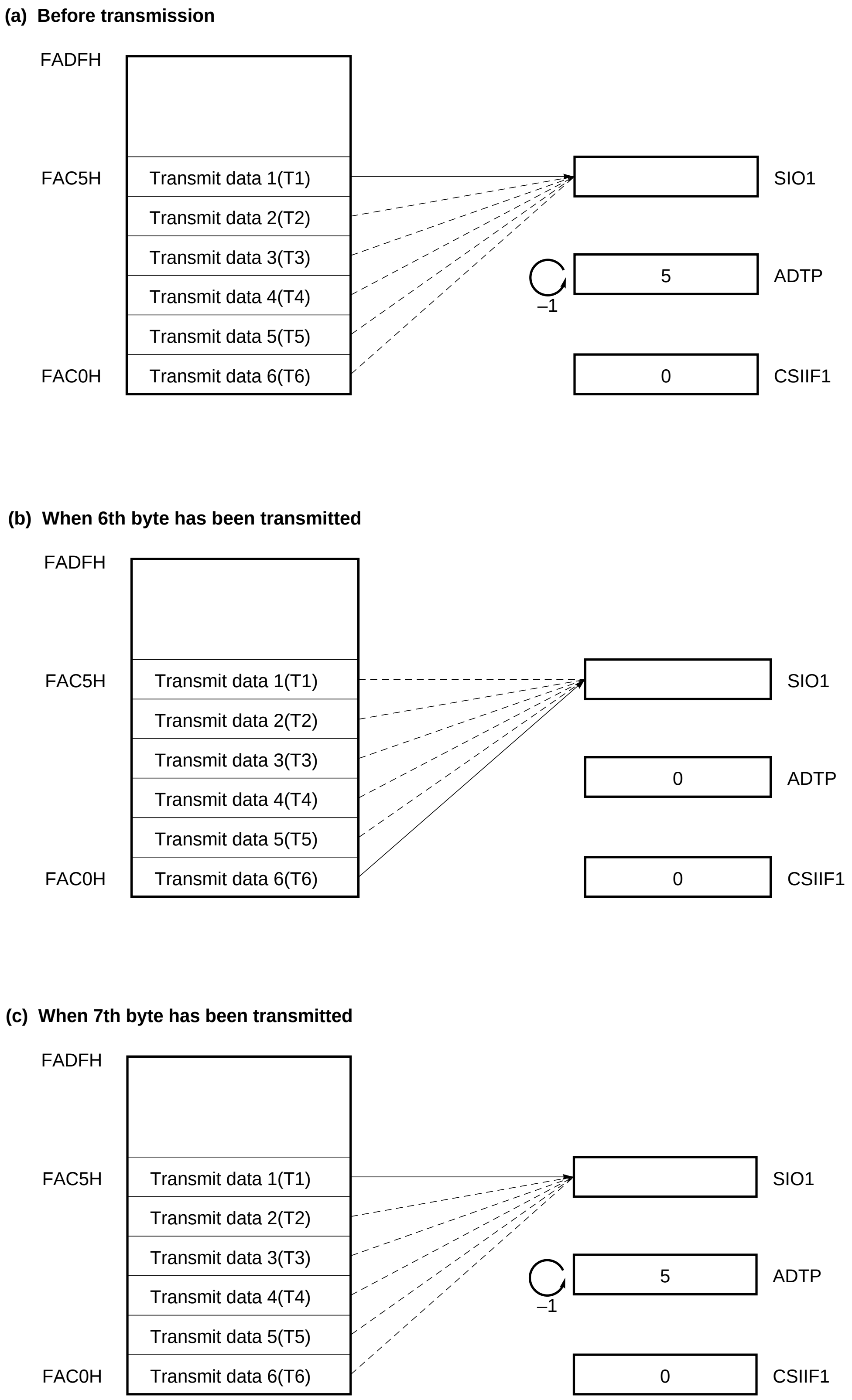
(ii) When 6th byte has been transmitted (Refer to Figure 17-16 (b))

When the sixth byte has been transmitted, the interrupt request flag (CSIIF1) is not set.
The previous pointer value is assigned to the ADTP.

(iii) When 7th byte is transmitted (Refer to Figure 17-16 (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 buffer RAM to SIO1.

Figure 17-16. Buffer RAM Operation when 6 Bytes Are Transmitted (in repetitive transmit 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 automatic transmission or reception under execution, reset bit 7 (CSIE1) of the serial operation mode register 1 (CSIM1) to 0.

At this time, transmission or reception is not stopped until transfer of 8-bit data has been completed.

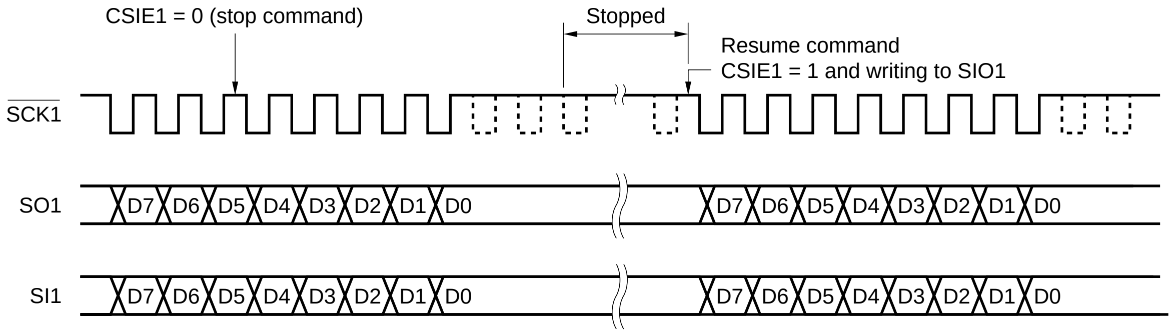
When transmission or reception has been stopped, bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) is reset to 0 after the data of the eighth bit has been transferred, and all the port pins multiplexed with serial interface pins (P20/SI1, P21/SO1, P22/SCK1, P23/$\overline{\text{STB}}$, and P24/BUSY) are set in the port mode.

To resume automatic transmission/reception, set CSIE1 to 1, and write any value to the serial I/O shift register 1 (SIO1). This allows the rest of the data to be transferred.

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer of 8-bit data, even in progress, is stopped, and the HALT mode is set. When the HALT mode is released, automatic transmission/reception is resumed from where it was stopped.

2. When automatic transmission/reception was stopped, do not change the operation mode to the 3-wire serial I/O mode while TRF = 1.

Figure 17-17. Stopping and Resuming Automatic Transmission/Reception



CSIE1: Bit 7 of serial operation mode register 1 (CSIM1)

(4) Synchronization control

Busy control and strobe control are functions to synchronize transmission/reception between the master device and a slave device.

By using these functions, a shift in bits being transmitted or received can be detected.

(a) Busy control option

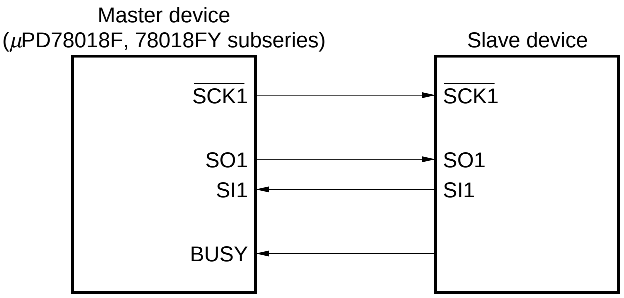
Busy control is a function to keep the serial transmission/reception by the master device waiting while the busy signal output by a slave device to the master is active.

When using this busy control option, the following conditions must be satisfied.

- Bit 5 (ATE) of the serial operation mode register 1 (CSIM1) is set to 1.
- Bit 1 (BUSY1) of the automatic data transmit/receive control register (ADTC) is set to 1.

Figure 17-18 shows the system configuration of the master device and a slave device when the busy control option is used.

Figure 17-18. System Configuration when Busy Control Option Is Used



The master device inputs the busy signal output by the slave device to the BUSY/P24 pin. The master device samples the input busy signal in synchronization with the falling of the serial clock. Even if the busy signal becomes active while 8-bit data is being transmitted or received, transmission/reception by the master is not kept waiting. If the busy signal is active at the rising edge of the serial clock 2 clocks after completion of transmission/reception of the 8-bit data, the busy input becomes valid. After that, the master transmission/reception is kept waiting while the busy signal is active.

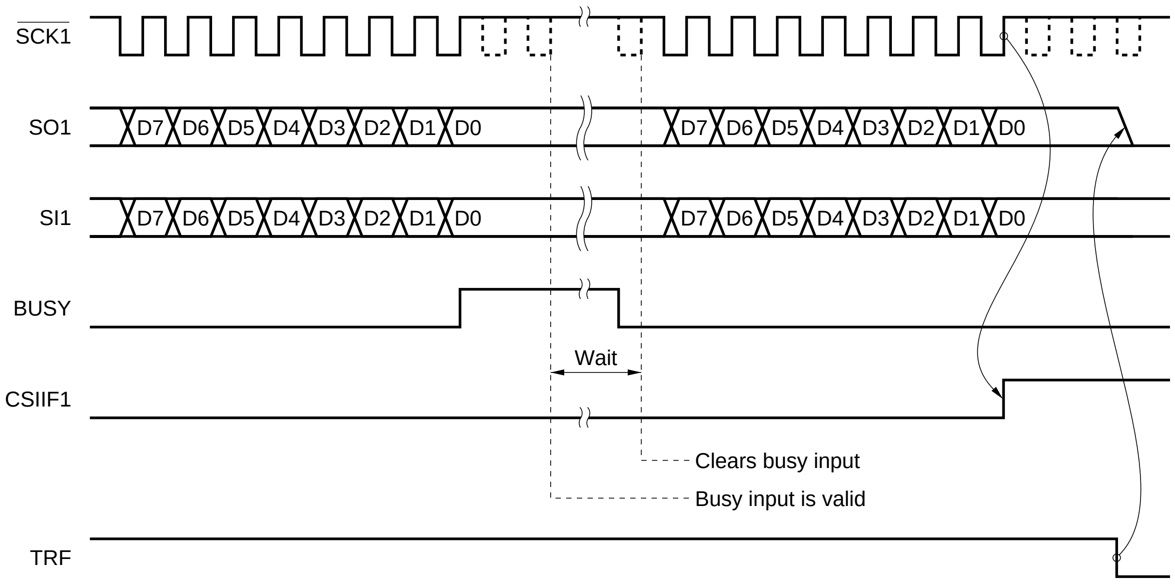
The active level of the busy signal is set by bit 0 (BUSY0) of ADTC.

BUSY0 = 0: Active high
BUSY0 = 1: Active low

When using the busy control option, select the internal clock as the serial clock. Control with the busy signal cannot be implemented with the external clock.
Figure 17-19 shows the operation timing when the busy control option is used.

Caution Busy control cannot be used simultaneously with the interval time control function of the automatic data transmit/receive interval specification register (ADTI). If used, busy control is invalid.

Figure 17-19. Operation Timing When Busy Control Option Is Used (when BUSY0 = 0)

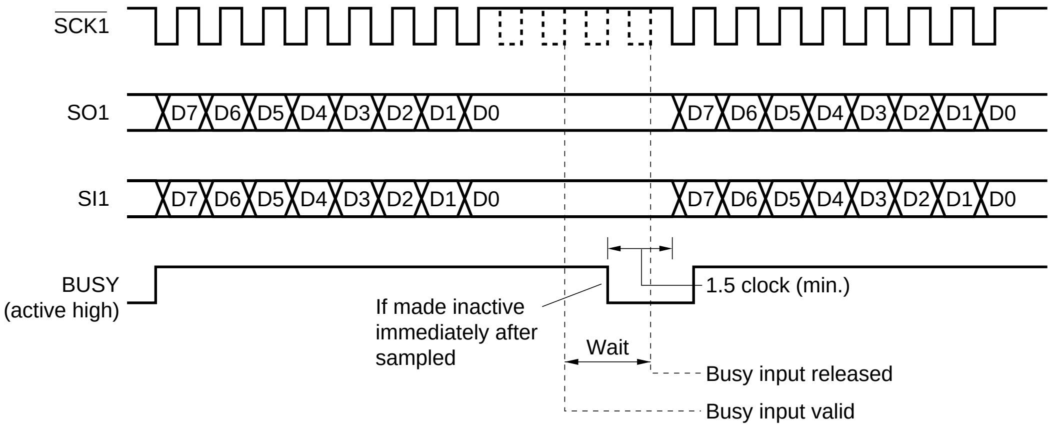


Caution If the TRF is cleared, the SO1 pin goes low.

Remark CSIIF1: Interrupt request flag
TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

When the busy signal becomes inactive, waiting is released. If the sampled busy signal is inactive, transmission/reception of the next 8-bit data is started at the falling edge of the next clock.
Because the busy signal is asynchronous with the serial clock, it takes up to 1 clock until the busy signal, even if made inactive by the slave, is sampled. It takes 0.5 clock until data transfer is started after the busy signal was sampled.
To accurately release waiting, the slave must keep the busy signal inactive at least for the duration of 1.5 clock.
Figure 17-20 shows the timing of the busy signal and releasing the waiting. This figure shows an example where the busy signal is active as soon as transmission/reception has been started.

Figure 17-20. Busy Signal and Wait Release (when BUSY0 = 0)



(b) Busy & strobe control option

Strobe control is a function to synchronize data transmission/reception between the master and slave devices. The master device outputs the strobe signal from the STB/P23 pin when 8-bit transmission/ reception has been completed. By this signal, the slave device can determine the timing of the end of data transmission. Therefore, synchronization is established even if a bit shift occurs because noise is superimposed on the serial clock, and transmission of the next byte is not affected by the bit shift.

To use the strobe control option, the following conditions must be satisfied:

- Bit 5 (ATE) of the serial operation mode register 1 (CSIM1) is set to 1.
- Bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) is set to 1.

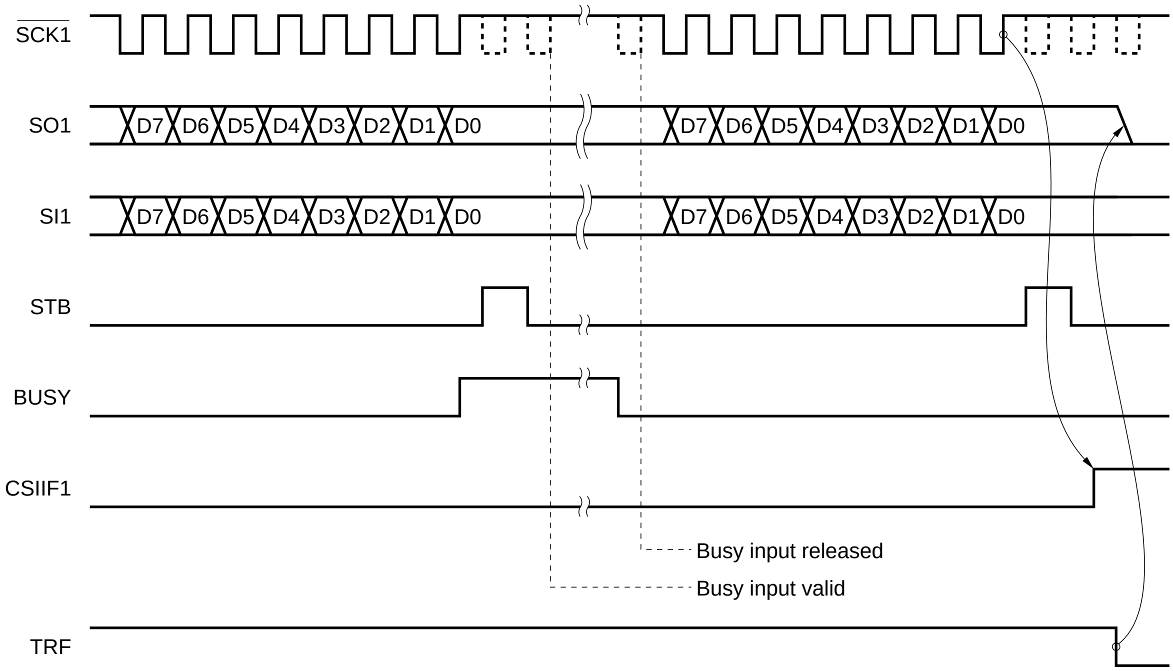
Usually, the busy control and strobe control options are simultaneously used as handshake signals. In this case, the strobe signal is output from the STB/P23 pin, and the BUSY/P24 pin is sampled, and transmission/reception can be kept waiting while the busy signal is input.

When the strobe control option is not used, the P23/STB pin can be used as a normal I/O port pin.

Figure 17-21 shows the operation timing when the busy & strobe control options are used.

When the strobe control option is used, the interrupt request flag (CSIIF1) that is set on completion of transmission/reception is set after the strobe signal is output.

Figure 17-21. Operation Timing when Busy & Strobe Control Options Are 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 transmit/receive control register (ADTC)

(c) Bit shift detection by busy signal

During automatic transmission/reception, a bit shift of the serial clock of the slave device may occur because noise is superimposed on the serial clock signal output by the master device. Unless the strobe control option is used at this time, the bit shift affects transmission of the next byte. In this case, the master can detect the bit shift by checking the busy signal during transmission by using the busy control option.

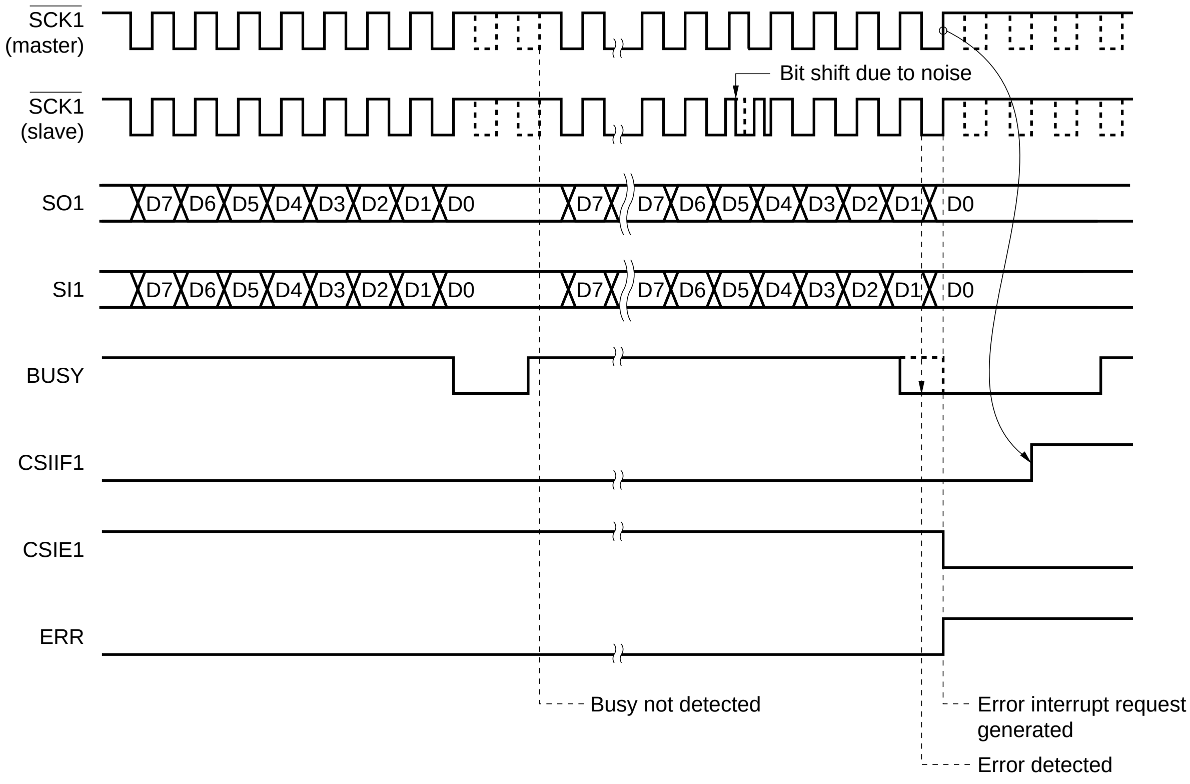
A bit shift is detected by using the busy signal as follows:

The slave outputs the busy signal after the rising of the eighth serial clock during data transmission/ reception (to not keep transmission/reception waiting by the busy signal at this time, make the busy signal inactive within 2 clocks).

The master samples the busy signal in synchronization of the falling of the leading side of the serial clock. If a bit shift does not occur, all the eight serial clocks that have been sampled are inactive. If the sampled serial clocks are active, it is assumed that a bit shift has occurred, and error processing is executed (by setting bit 4 (ERR) of the automatic transmit/receive control register (ADTC) to 1).

Figure 17-22 shows the operation timing of the bit shift detection function by the busy signal.

Figure 17-22. Operation Timing of Bit Shift Detection Function by Busy Signal (when BUSY0 = 1)



CSIIF1 : Interrupt request flag
CSIE1 : Bit 7 of serial operation mode register1 (CSIM1)
ERR : Bit 4 of automatic data transmit/receive control register (ADTC)

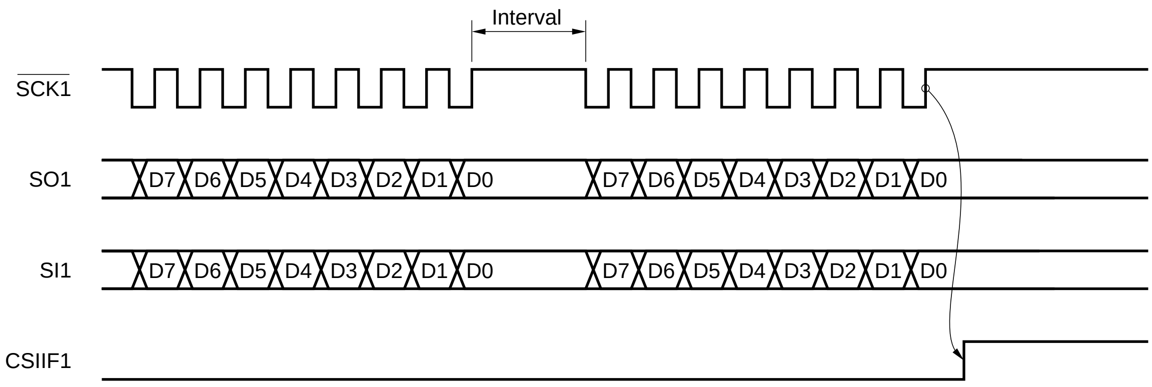
(5) Interval time of automatic transmission/reception

When using the automatic transmit/receive function, an interval time elapses after 1 byte has been transmitted or received, until the next transmission/reception is executed because data is written to or read from the buffer RAM.

To use the automatic transmit/receive function with the internal clock, the interval time is dependent on the CPU processing of the timing of the eighth rising of the serial clock and the set value of the automatic data transmit/receive interval specification register (ADTI). Whether the interval time is dependent on ADIT is selected by setting of the bit 7 (ADTI7) of ADTI. If ADTI7 is reset to 0, the interval time is dependent on the CPU processing only. If ADTI7 is set to 1, the interval time determined by the set contents of ADTI or interval time by the CPU processing is selected whichever greater.

To use the automatic transmit/receive function with the external clock, the external clock must be input in the manner that the interval time is equal to or greater than the time described in (b) below.

Figure 17-23. Interval Time of Automatic Transmission/Reception



CSIIF1: Interrupt request flag

(a) When using automatic transmit/receive function with internal clock

The internal clock is used when bit 1 (CSIM11) of the serial operation mode register 1 (CSIM1) is set to 1.

To use the automatic transmit/receive function with the internal clock, the interval time by the CPU processing is as follows:

When bit 7 (ADTI7) of the automatic data transmit/receive interval specification register (ADTI) is reset to 0, the interval time is that by the CPU processing. When ADTI7 is set to 1, the interval time is that determined by the set contents of ADTI or that by the CPU processing, whichever greater.

For the interval time by ADTI, refer to **Figure 17-5 Format of Automatic Data Transmit/Receive Interval Time Specification Register**.

Table 17-3. Interval Time by CPU Processing (with internal clock)

CPU Processing	Interval Time
With multiplication instruction used	MAX. (2.5 T_{SCK}, 13T_{CPU})
With division instruction used	MAX. (2.5 T_{SCK}, 20T_{CPU})
External access 1 wait mode	MAX. (2.5 T_{SCK}, 9T_{CPU})
Others	MAX. (2.5 T_{SCK}, 7T_{CPU})

T_{SCK} : 1/f_{SCK}

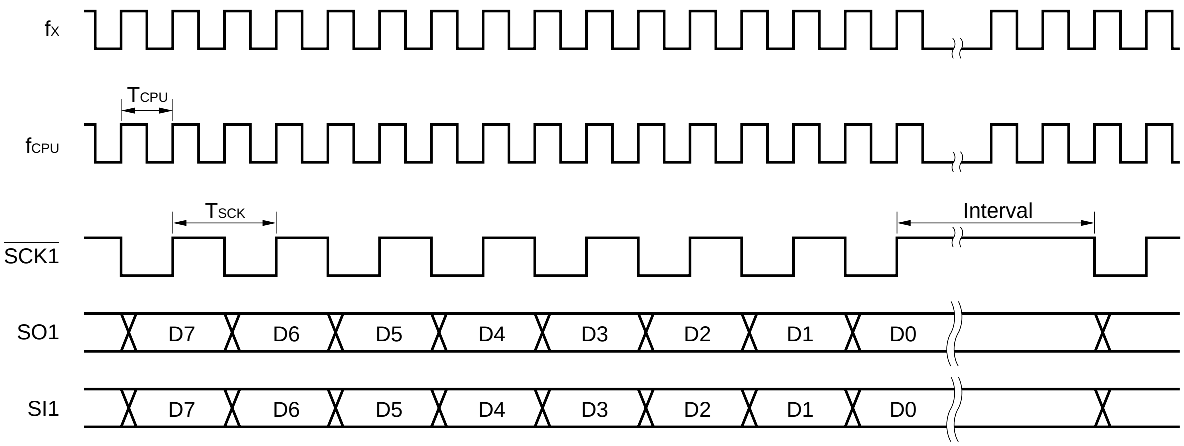
f_{SCK} : Serial clock frequency

T_{CPU} : 1/f_{CPU}

f_{CPU} : CPU clock (set by bits 0 through 2 (PCC0 through PCC2) of processor clock control register (PCC))

MAX. (a, b) : Value of a or b whichever greater

Figure 17-24. Operation Timing when Automatic Transmit/Receive Function Is Used with Internal Clock



f_X : Main system clock oscillation frequency
f_{CPU} : CPU clock (set by bits 0 through 2 (PCC0 through 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 transmit/receive function with external clock

The external clock is used when bit 1 (CSIM11) of the serial operation mode register 1 (CSIM1) is cleared to 0.

To use the automatic transmit/receive function with the external clock, the external clock must be input such that the interval time is as follows:

Table 17-4. Interval Time by CPU Processing (with external clock)

CPU Processing	Interval Time
With multiplication instruction used	$13T_{CPU}$ MIN.
With division instruction used	$20T_{CPU}$ MIN.
External access 1 wait mode	$9T_{CPU}$ MIN.
Others	$7T_{CPU}$ MIN.

T_{CPU} : $1/f_{CPU}$
f_{CPU} : CPU clock (set by bits 0 through 2 (PCC0 through PCC2) of processor clock control register (PCC))

CHAPTER 18 INTERRUPT FUNCTIONS AND TEST FUNCTIONS

18.1 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.

The non-maskable interrupts have one interrupt request source from the watchdog timer.

(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 register (PR0L, PR0H): one with higher priority and the other with 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 18-1**).

This interrupt generates a standby release signal.

As the maskable interrupts, four external interrupt request sources and eight internal interrupt request sources 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.

18.2 Interrupt Sources and Configuration

A total of 14 interrupt sources including non-maskable, maskable, and software interrupt sources are available (refer to **Table 18-1**).

Table 18-1. Interrupt Sources

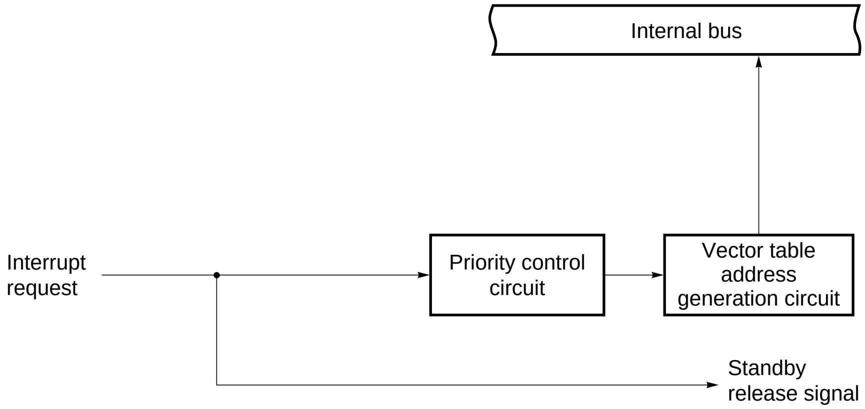
Interrupt Type	Default Priority [Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type [Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Overflow of watchdog timer (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Overflow of watchdog timer (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.

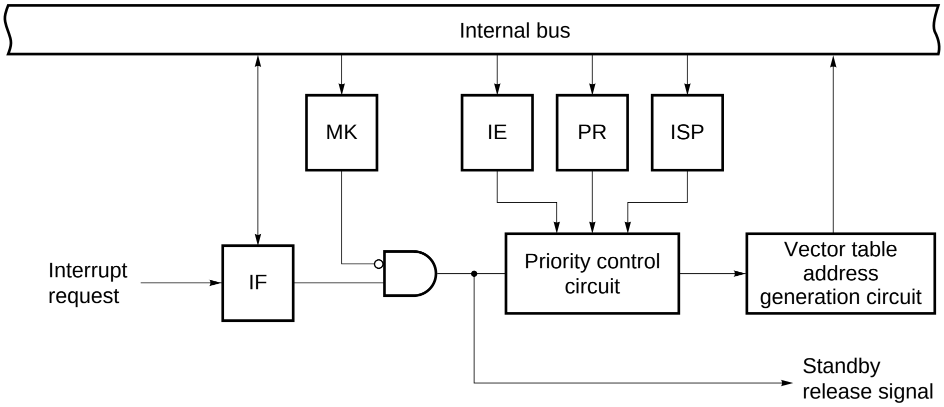
2. Basic configuration types (A) to (E) respectively correspond to (A) to (E) on the following pages.

Figure 18-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

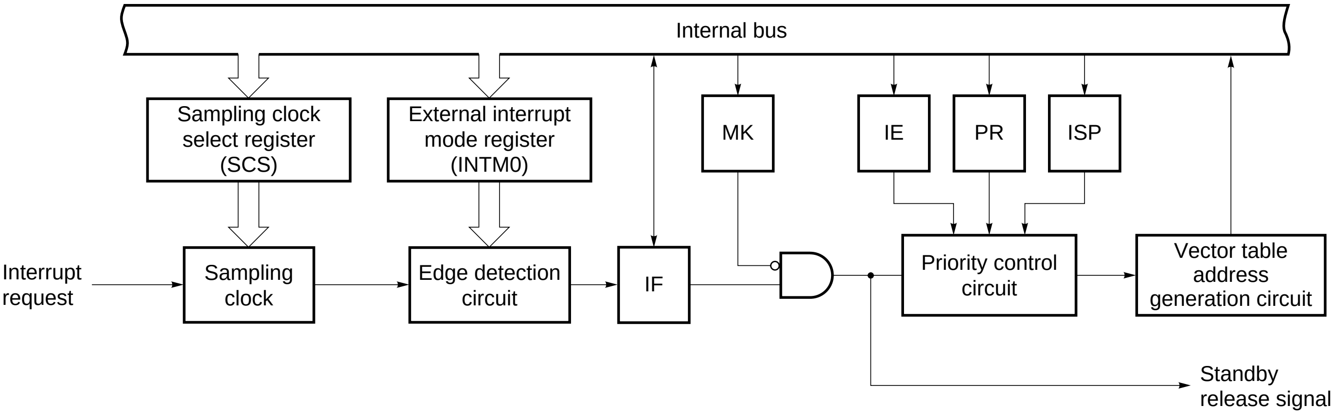
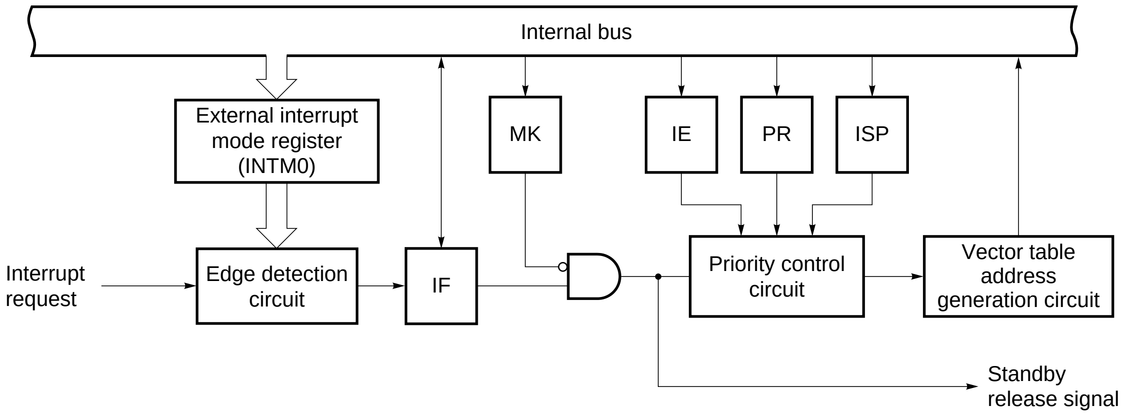
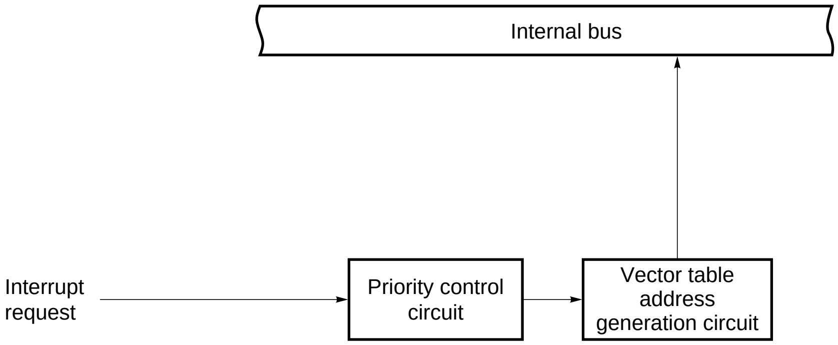


Figure 18-1. Basic Configuration of Interrupt Function (2/2)

(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

18.3 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 18-2 shows the names of the interrupt request flags, interrupt mask flags, and priority specification flags corresponding to the respective interrupt request sources.

Table 18-2. 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		CSIMK0		CSIPR0	
INTCSI1	CSIIF1		CSIMK1		CSIPR1	
INTTM3	TMIF3		TMMK3		TMPR3	
INTTM0	TMIF0	IF0H	TMMK0	MK0H	TMPR0	PR0H
INTTM1	TMIF1		TMMK1		TMPR1	
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 18-2. Format of Interrupt Request Flag Registers

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	On 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	WTIF[Note]	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. Be sure to set bits 4, 6, 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 18-3. Format of Interrupt Mask Flag Register

Symbol	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	〈0〉	Address	On reset	R/W
MK0L	TMMK3	CSIMK1	CSIMK0	PMK3	PMK2	PMK1	PMK0	TMMK4	FFE4H	FFH	R/W

Symbol	7	6	〈5〉	4	〈3〉	〈2〉	〈1〉	〈0〉	Address	On reset	R/W
MK0H	1	1	WTMK[Note]	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 function.

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. Be sure to set bits 4, 6, 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 18-4. Format of Priority Specification Flag Register

Symbol	⟨7⟩	⟨6⟩	⟨5⟩	⟨4⟩	⟨3⟩	⟨2⟩	⟨1⟩	⟨0⟩	Address	On 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. Be sure to set bits 4 through 7 of PR0H to 1.

(4) External interrupt mode register (INTM0)

This register sets the valid edges of INTP0 through 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 18-5. Format of External Interrupt Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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 18-6. Format of Sampling Clock Select Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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	$f_X/2^6$ (156kHz)
1	1	$f_X/2^7$ (78.1kHz)

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.**

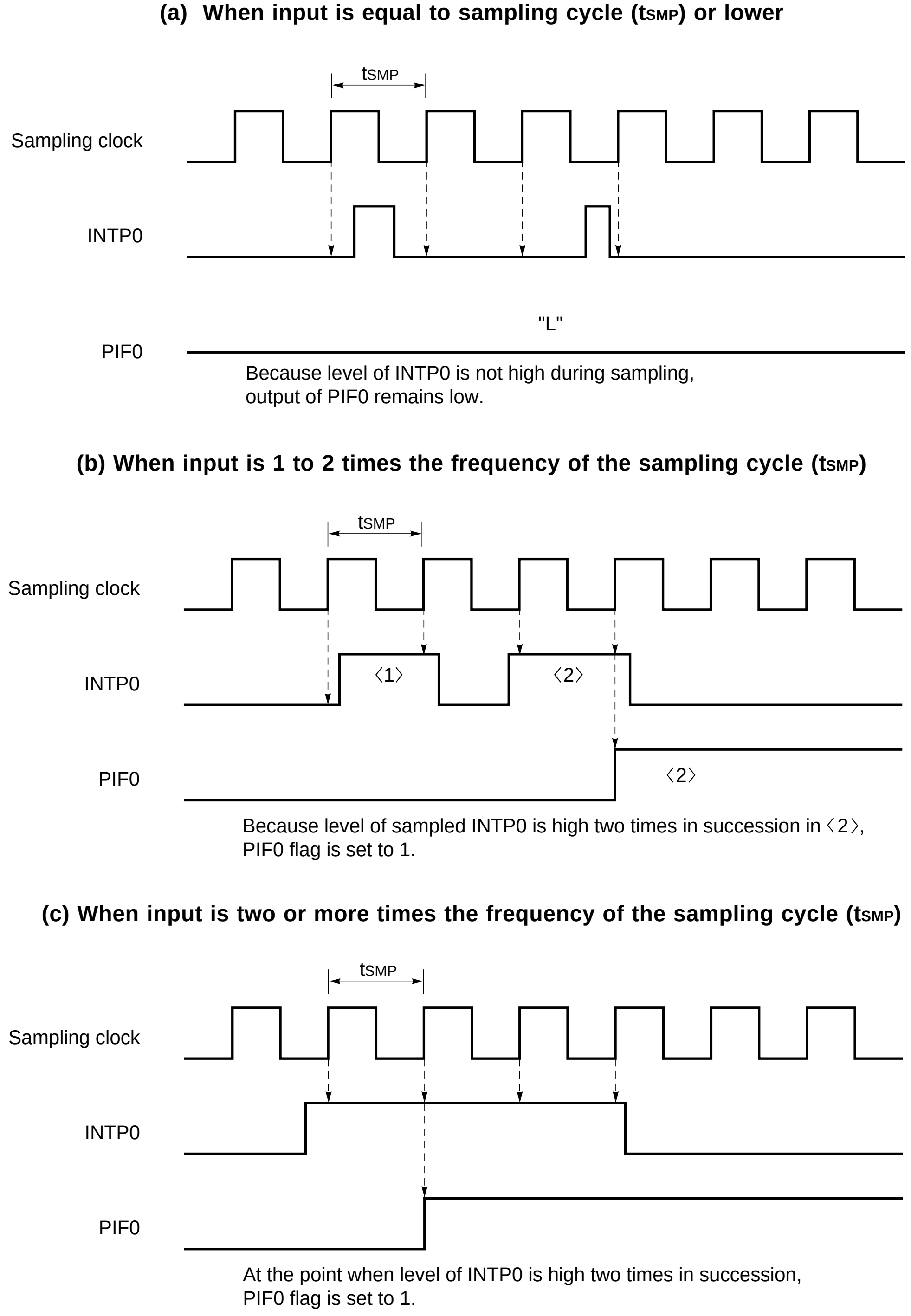
Remarks **1.** N : Value (N = 0-4) set to bits 0 through 2 (PCC0-PCC2) of processor clock control register

2. f_X : Main system clock oscillation frequency

3. ( ) : At f_X = 10.0 MHz operation

The noise eliminating circuit sets the PIF0 flag to 1 when the input level of INTP0 is active two times in succession.

Figure 18-7. 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 level of INTP0 is not high during sampling, output of PIF0 remains low.

(b) When input is 1 to 2 times the frequency of the sampling cycle (t_{SMP})

Because level of sampled INTP0 is high two times in succession in 〈2〉, PIF0 flag is set to 1.

(c) When input is two or more times the frequency of the sampling cycle (t_{SMP})

At the point when level of INTP0 is high two times in succession, PIF0 flag is set to 1.

(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 nesting 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, PSW 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 contents 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 18-8. Configuration of Program Status Word

Symbol	7	6	5	4	3	2	1	0	On 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 is not accepted, or interrupt with lower priority is processed (all maskable interrupts are enabled)

IE	Enables/disables accepting interrupt request
0	Disables
1	Enables

18.4 Interrupt Processing Operation

18.4.1 Non-maskable interrupt request acceptance operation

The non-maskable interrupt request is unconditionally accepted even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

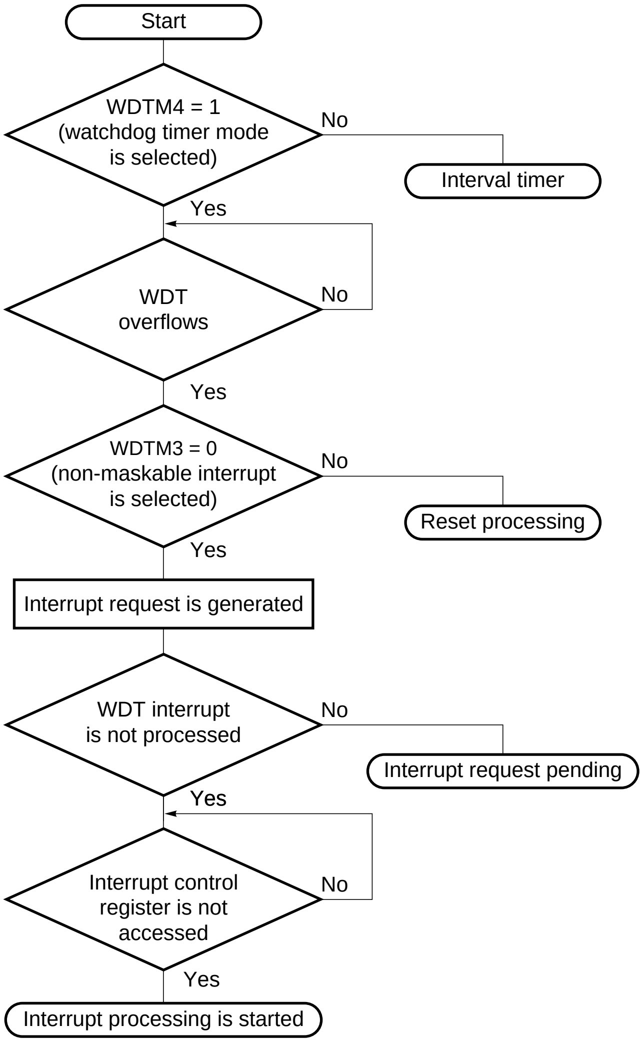
When the non-maskable interrupt request is acknowledged, program status word (PSW) and program counter (PC) are saved to the stack in that order, the IE flag and ISP flag are reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

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 18-9 shows the flowchart from generation of the non-maskable interrupt to accepting it. Figure 18-10 shows the timing of accepting the non-maskable interrupt, and Figure 18-11 shows the operation performed if the non-maskable interrupt occurs in duplicate.

Phase-out/Discontinued

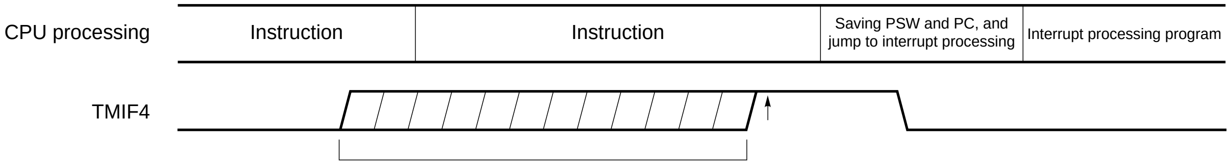
Figure 18-9. Flowchart of Non-Maskable Interrupt Request from Generation to Acceptance



WDTM : Watchdog timer mode register

WDT : Watchdog timer

Figure 18-10. Timing of Non-Maskable Interrupt Request Acceptance

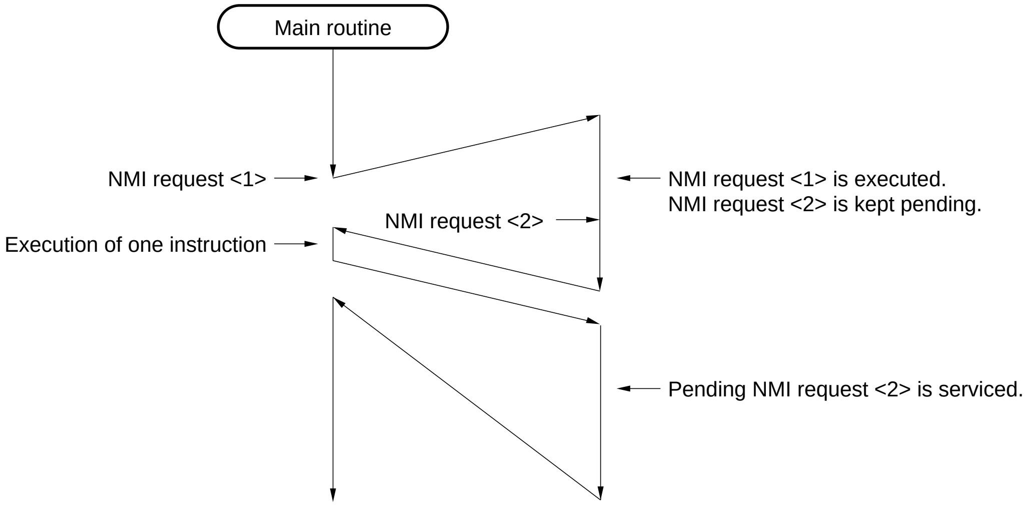

CPU processing
Instruction
Instruction
Saving PSW and PC, and jump to interrupt processing
Interrupt processing program
TMIF4

The interrupt generated in this period is accepted at the point of ↑.

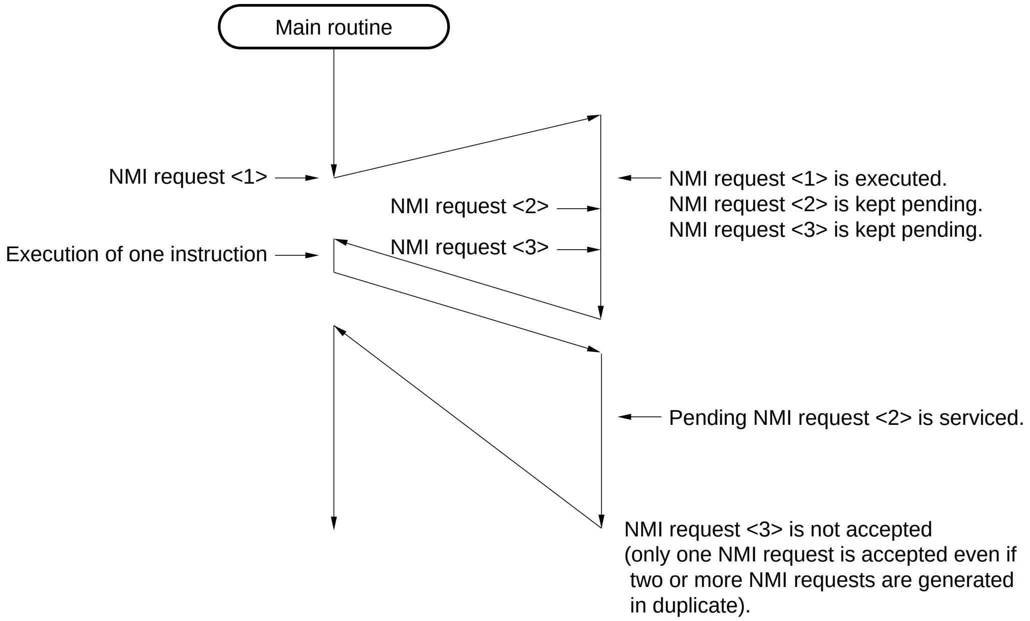
TMIF4 : Watchdog timer interrupt request flag

Figure 18-11. Non-Maskable Interrupt Acceptance Operation

(a) If a new non-maskable interrupt request is generated while non-maskable interrupt service program is being executed


Main routine
NMI request <1>
NMI request <1> is executed.
NMI request <2> is kept pending.
NMI request <2>
Execution of one instruction
Pending NMI request <2> is serviced.

(b) If two new non-maskable interrupt requests are generated while non-maskable interrupt service program is being executed


Main routine
NMI request <1>
NMI request <1> is executed.
NMI request <2> is kept pending.
NMI request <3> is kept pending.
NMI request <2>
Execution of one instruction
NMI request <3>
Pending NMI request <2> is serviced.
NMI request <3> is not accepted
(only one NMI request is accepted even if
two or more NMI requests are generated
in duplicate).

18.4.2 Maskable interrupt request acceptance operation

A maskable interrupt request can be accepted when the interrupt request flag is set to 1 and the corresponding 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 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 as follows:

For the timing of interrupt request acceptance, refer to **Figures 18-3** and **18-4.**

Table 18-3. 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, the default priority takes precedence.

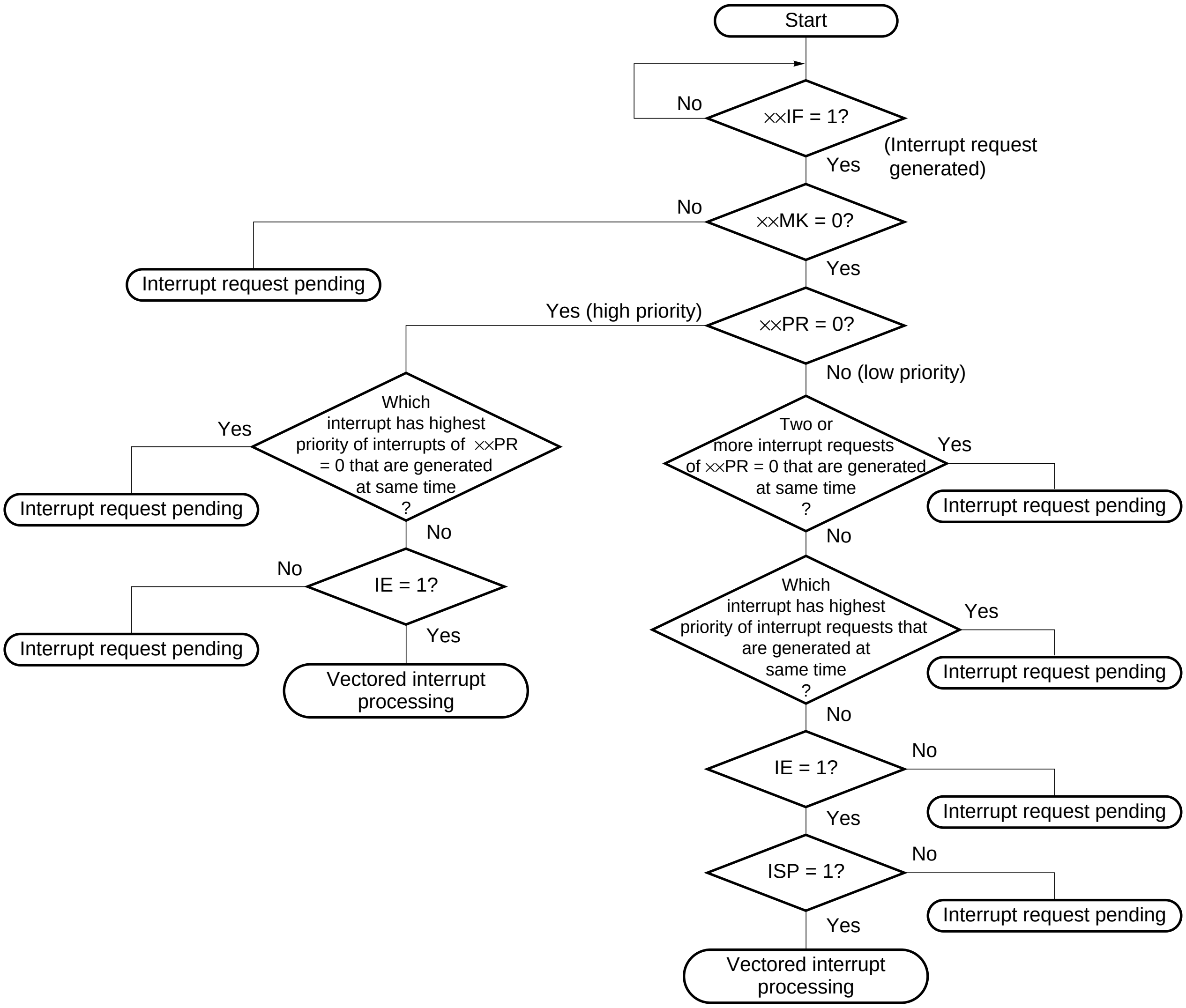
A pended interrupt is accepted when the status where it can be accepted is set.

Figure 18-12 shows the algorithm of accepting interrupts.

When a maskable interrupt request is accepted, the program status word (PSW) and program counter (PC) are saved to the stack in that order, IE flag is reset to 0, and the content of the interrupt 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 18-12. Interrupt Request Acceptance Program Algorithm


Start
No
××IF = 1?
(Interrupt request generated)
Yes
No
××MK = 0?
Interrupt request pending
Yes
Yes (high priority)
××PR = 0?
No (low priority)
Which interrupt has highest priority of interrupts of ××PR = 0 that are generated at same time ?
Yes
Interrupt request pending
No
No
IE = 1?
Interrupt request pending
Yes
Vectored interrupt processing
Two or more interrupt requests of ××PR = 0 that are generated at same time ?
Yes
Interrupt request pending
No
Which interrupt has highest priority of interrupt requests that are generated at same time ?
Yes
Interrupt request pending
No
IE = 1?
No
Interrupt request pending
Yes
ISP = 1?
No
Interrupt request pending
Yes
Vectored interrupt processing

××IF : Interrupt request flag

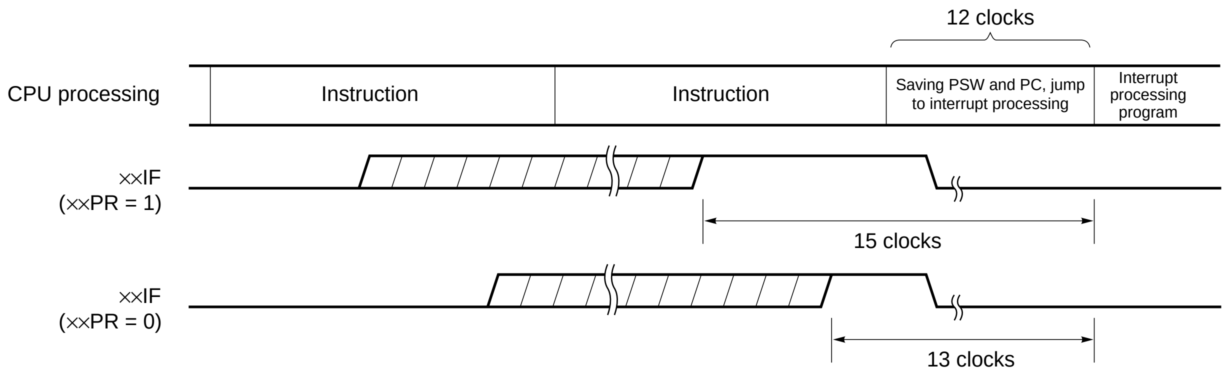
××MK : Interrupt mask flag

××PR : Priority specification flag

IE : Flag controlling accepting maskable interrupt request (1 = enable, 0 = disable)

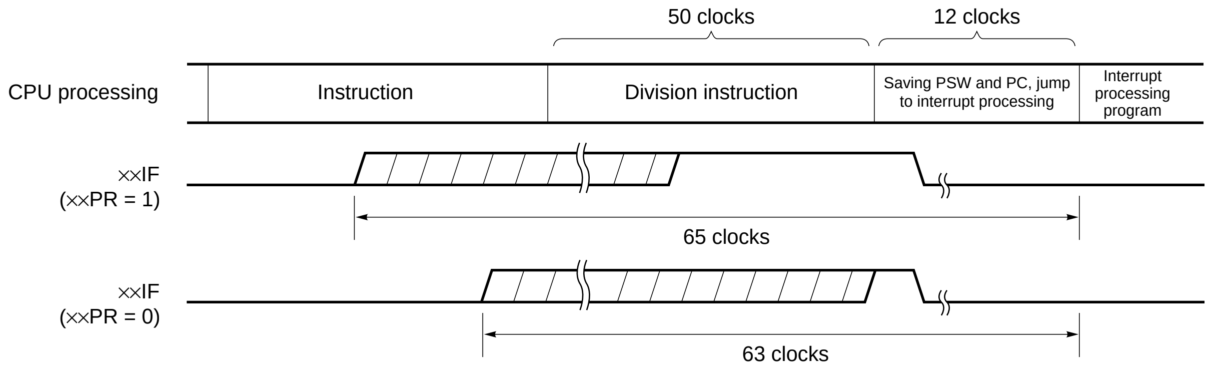
ISP : Flag indicating priority of interrupt currently serviced (0 = interrupt with high priority serviced, 1 = interrupt request is not accepted, or interrupt with low priority is serviced)

Figure 18-13. Interrupt Request Acceptance Timing (Minimum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU} : CPU clock)

Figure 18-14. Interrupt Request Acceptance Timing (Maximum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU} : CPU clock)

18.4.3 Software interrupt request acceptance operation

The software interrupt request can be accepted when the BRK instruction is executed. This interrupt cannot be disabled.

When the software interrupt request is accepted, the program status word (PSW) and program counter (PC) are saved to the stack in that order, 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.

18.4.4 Nesting

Accepting another interrupt request while an interrupt is being serviced is called nesting.

Nesting does not take place unless the interrupts (except the non-maskable interrupt) are enabled to be accepted (IE = 1). Accepting another interrupt request is disabled (IE = 0) when one interrupt has been accepted. Therefore, to enable nesting, the EI flag must be set to 1 during interrupt servicing, to enable the another interrupt.

Nesting may not occur even when the interrupts are enabled. This is controlled by the priorities of the interrupts. Although two types of priorities, default priority and programmable priority, may be assigned to an interrupt, nesting is controlled by using the programmable priority.

If an interrupt with the same level of priority as or the higher priority than the interrupt currently serviced occurs, that interrupt can be accepted and nested. If an interrupt with a priority lower than that of the currently serviced interrupt occurs, that interrupt cannot be accepted and nested.

An interrupt that is not accepted and nested because it is disabled or it has a low priority is kept pending. This interrupt is accepted after servicing of the current interrupt has been completed and one instruction of the main routine has been executed.

Nesting is not enabled while the non-maskable interrupt is being serviced.

Table 18-4 shows the interrupts that can be nested, and Figure 18-15 shows an example of nesting.

Table 18-4. Interrupt Requests that Can Be Nested during Interrupt Processing

Nesting Request Interrupt being accepted		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			PR = 0		PR = 1	
			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt processing		×	×	×	×	×
Maskable interrupt processing	ISP = 0	○	○	×	×	×
	ISP = 1	○	○	×	○	×
Software interrupt processing		○	○	×	○	×

Remarks **1.** ○ : nesting enabled

2. × : nesting disabled

3. ISP and IE are flags included in PSW.

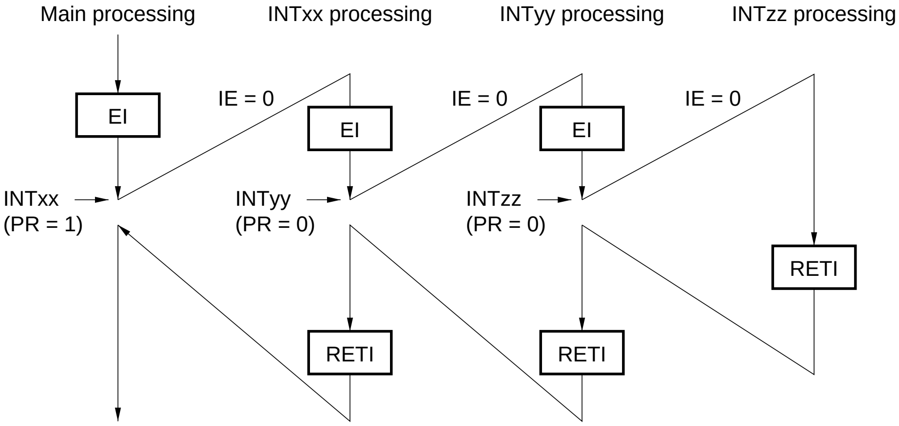
ISP = 0 : Interrupt with higher priority is processed.
ISP = 1 : Interrupt is not accepted or interrupt with lower priority is processed.
IE = 0 : Accepting interrupt is disabled
IE = 1 : Accepting interrupt is enabled

4. PR is a flag included in PR0L, PR0H.

PR = 0 : Higher priority flag
PR = 1 : Lower priority flag

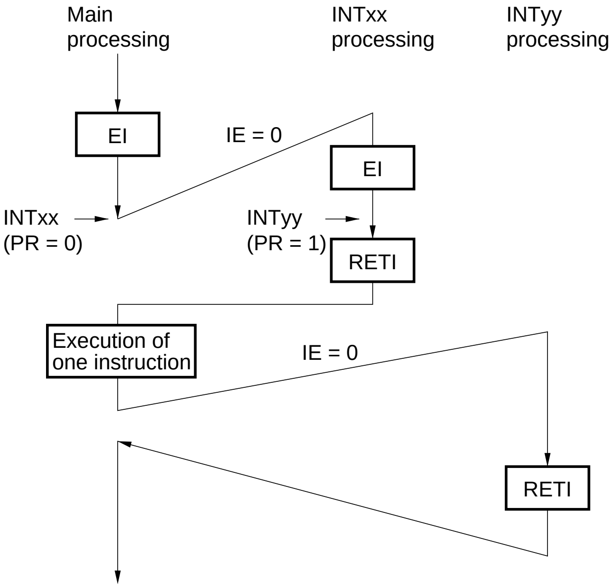
Figure 18-15. Example of Nesting (1/2)

Example 1. Example where nesting takes place two times



Two interrupt requests, INTyy and INTzz, are accepted while interrupt INTxx is serviced, and nesting takes place. Before each interrupt request is accepted, the EI instruction is always executed, and the interrupt is enabled.

Example 2. Example where nesting does not take place because of priority control

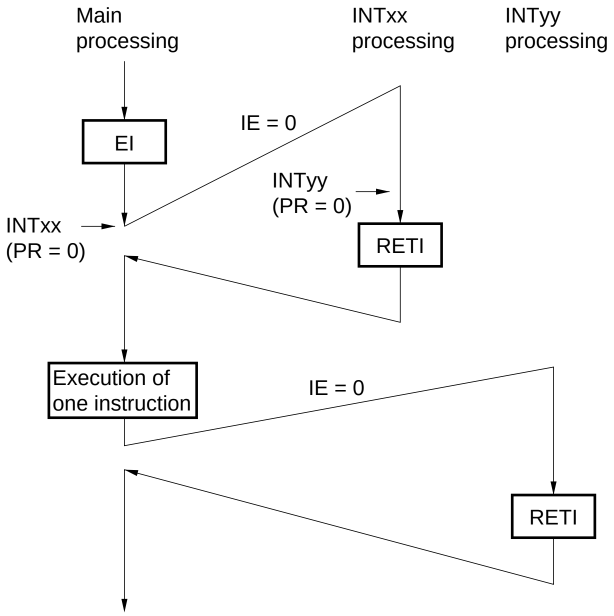


Interrupt request INTyy that is generated while interrupt INTxx is being serviced is not accepted because its priority is lower than that of INTxx, and therefore, nesting does not take place. INTyy request is kept pending, and is accepted after one instruction of the main routine has been executed.

PR = 0: High-priority level
PR = 1: Low-priority level
IE = 0 : Accepting interrupt request is disabled.

Figure 18-15. Example of Nesting (2/2)

Example 3. Example where nesting does not take place because interrupts are not enabled



Because interrupts are not enabled (EI instruction is not issued) in interrupt processing INTxx, interrupt request INTyy is not accepted, and nesting does not take place. INTyy request is kept pending, and is accepted after one instruction of the main routine has been executed.

PR = 0: High priority level
IE = 0 : Accepting interrupts is disabled.

18.4.5 Pending interrupt requests

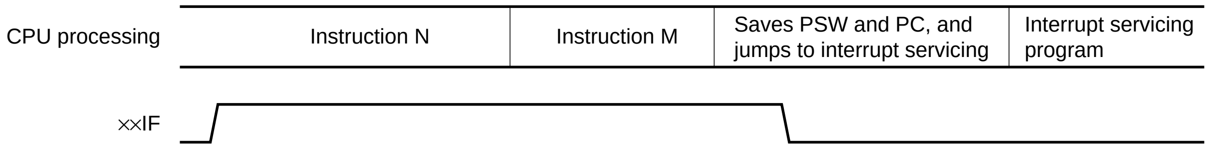
Even if an interrupt request is generated, the following instructions keep it pending until the next instruction execution has ended.

- 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
- Instructions manipulating IF0L, IF0H, MK0L, MK0H, PR0L, PR0H, and INTM0 registers

Caution The BRK instruction is not one of the above instructions that keep an interrupt request pending. However, the software interrupt that is started by execution of the BRK instruction clears the IE flag to 0. Therefore, even if a maskable interrupt request is generated while the BRK instruction is being executed, it is not accepted. However, the non-maskable interrupt is accepted.

Figure 18-16 shows the timing at which an interrupt request is accepted.

Figure 18-16. Pending Interrupt Request


CPU processing
Instruction N
Instruction M
Saves PSW and PC, and jumps to interrupt servicing
Interrupt servicing program
××IF

Remarks 1. Instruction N: Instruction that keeps interrupt request pending

2. Instruction M: Instruction that does not keep interrupt request pending

3. Operation of ××IF (interrupt request) is not affected by value of ××PR (priority level).

18.5 Test Functions

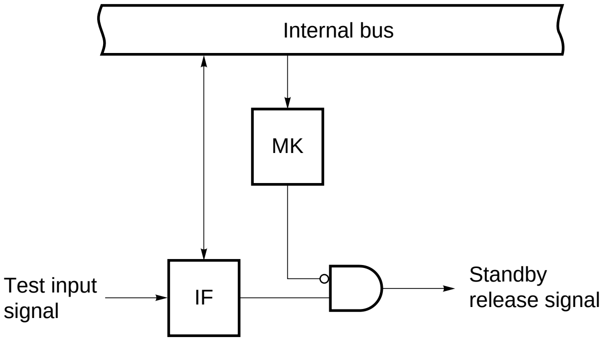
The test function sets the corresponding test input flag and generates a standby release signal when an overflow occurs in the watch timer and when a falling edge at port 4 is detected.

Unlike the interrupt function, this function does not perform vector processing.

Table 18-5. 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 18-17. Basic Configuration of Test Function



IF : Test input flag
MK : Test mask flag

18.5.1 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 18-6 shows the names of the test input flags and test mask flags corresponding to the respective test input signals.

Table 18-6. 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 18-18. Format of Interrupt Request Flag Register 0H

Symbol	7	6	〈5〉	4	〈3〉	〈2〉	〈1〉	〈0〉	Address	On 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 18-19. Format of Interrupt Mask Flag Register 0H

Symbol	7	6	〈5〉	4	〈3〉	〈2〉	〈1〉	〈0〉	Address	On 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 18-20. Format of Key Return Mode Register

Symbol	7	6	5	4	3	2	〈1〉	〈0〉	Address	On 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).

18.5.2 Test input signal acceptance operation

(1) Internal test input signal

An internal test input signal (INTWT) is generated when the watch timer overflows and the WTIF flag is set by it. At this time, the standby release signal is generated if it is not masked by the interrupt mask flag (WTMK). By checking the WTIF flag in a cycle shorter than the overflow cycle of the watch timer, the watch function can be effected.

(2) External test signal

If a falling edge is input to a pin of port 4 (P40 to P47), an external test input signal (INTPT4) is generated, setting the KRIF flag. At this time, the standby release signal is generated if it is not masked by the interrupt mask flag (KRMK). By using port 4 for key return signal input of a key matrix, the presence or absence of a key input can be checked by the status of the KRIF flag.

CHAPTER 19 EXTERNAL DEVICE EXTENSION FUNCTION

19.1 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 19-1. Pin Functions in External Memory Extension Mode

Pin Function when External Device is Connected		Shared by:
Name	Function	
AD0-AD7	Multiplexed address/data bus	P40-P47
A8-A15	Address bus	P50-P57
$\overline{\text{RD}}$	Read strobe signal	P64
$\overline{\text{WR}}$	Write strobe signal	P65
$\overline{\text{WAIT}}$	Wait signal	P66
ASTB	Address strobe signal	P67

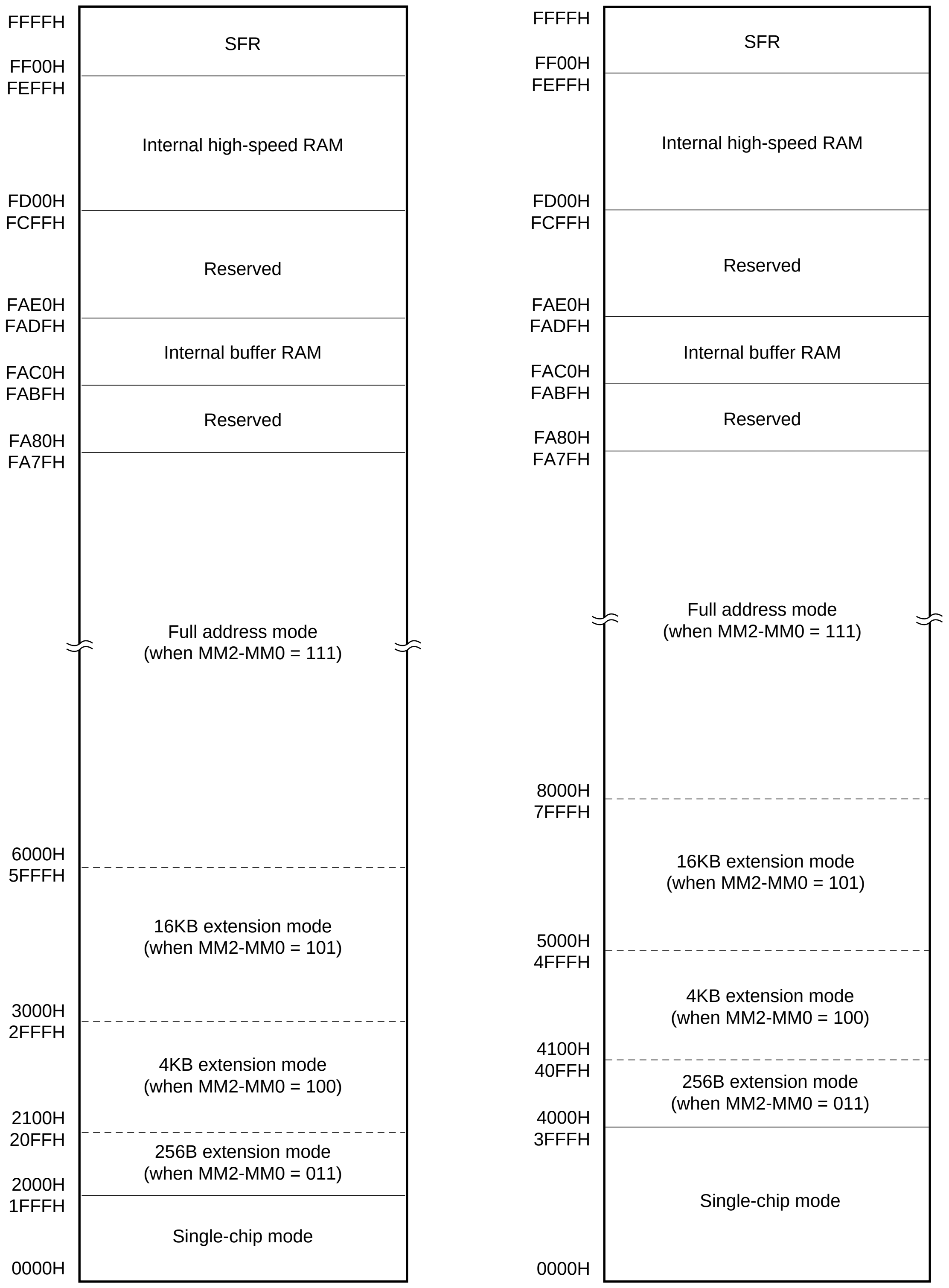
Table 19-2. Status of Ports 4-6 in External Memory Extension Mode

Port External Extension Mode	Port 4	Port 5								Port 6							
	0-7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Single-chip mode	Port	Port								Port							
256B extension mode	Address/data	Port								Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			
4KB extension mode	Address/data	Address				Port				Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			
16KB extension mode	Address/data	Address						Port		Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			
Full address mode	Address/data	Address								Port				$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$, ASTB			

Caution When the external wait function is not used, the $\overline{\text{WAIT}}$ 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.

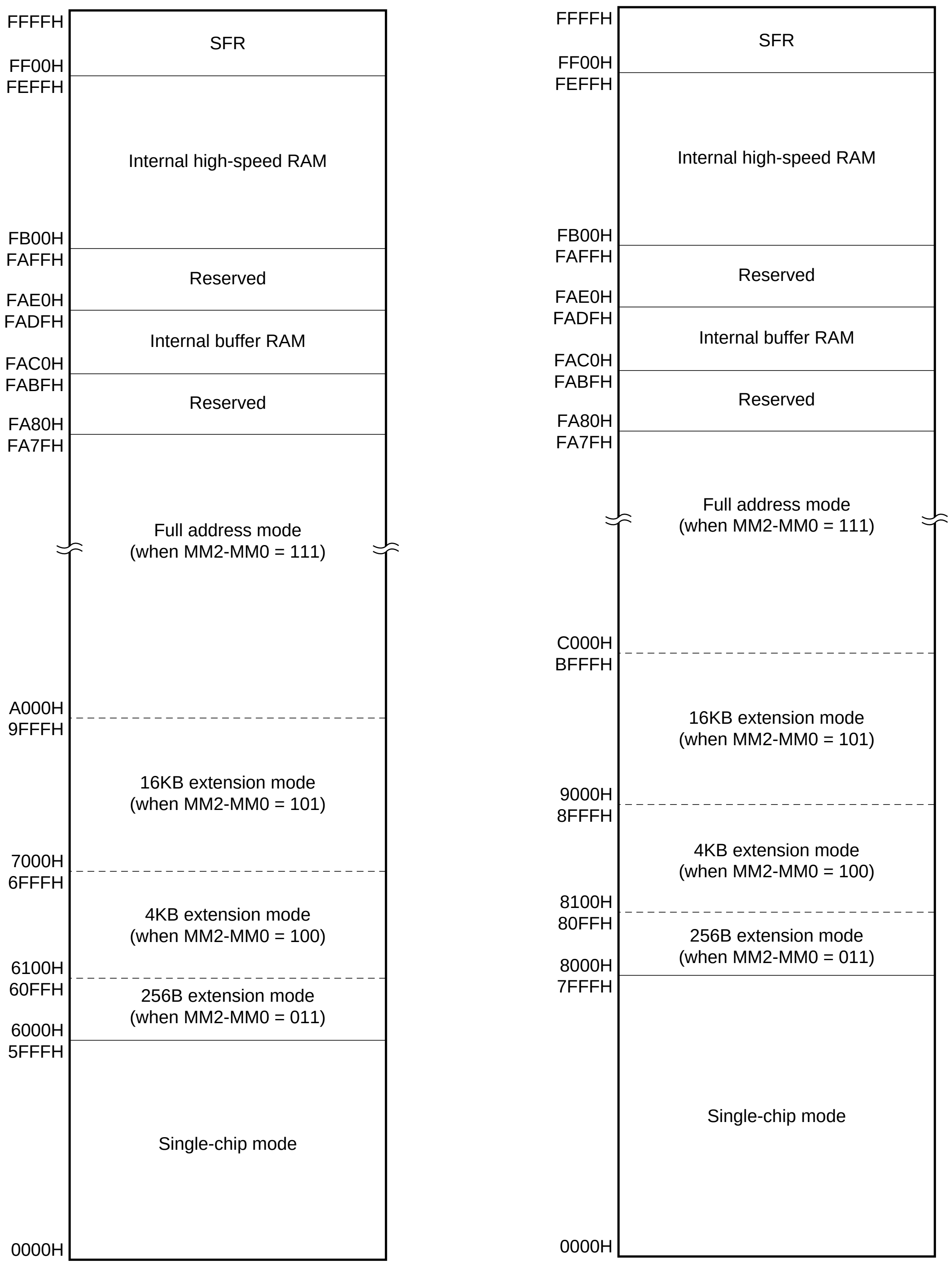
Figure 19-1. Memory Map when External Device Extension Function Is Used (1/4)



(a) Memory map of μPD78011F, 78011FY and μPD78P018F, 78P018FY with 8 KB internal PROM

(b) Memory map of μPD78012F, 78012FY and μPD78P018F, 78P018FY with 16 KB internal PROM

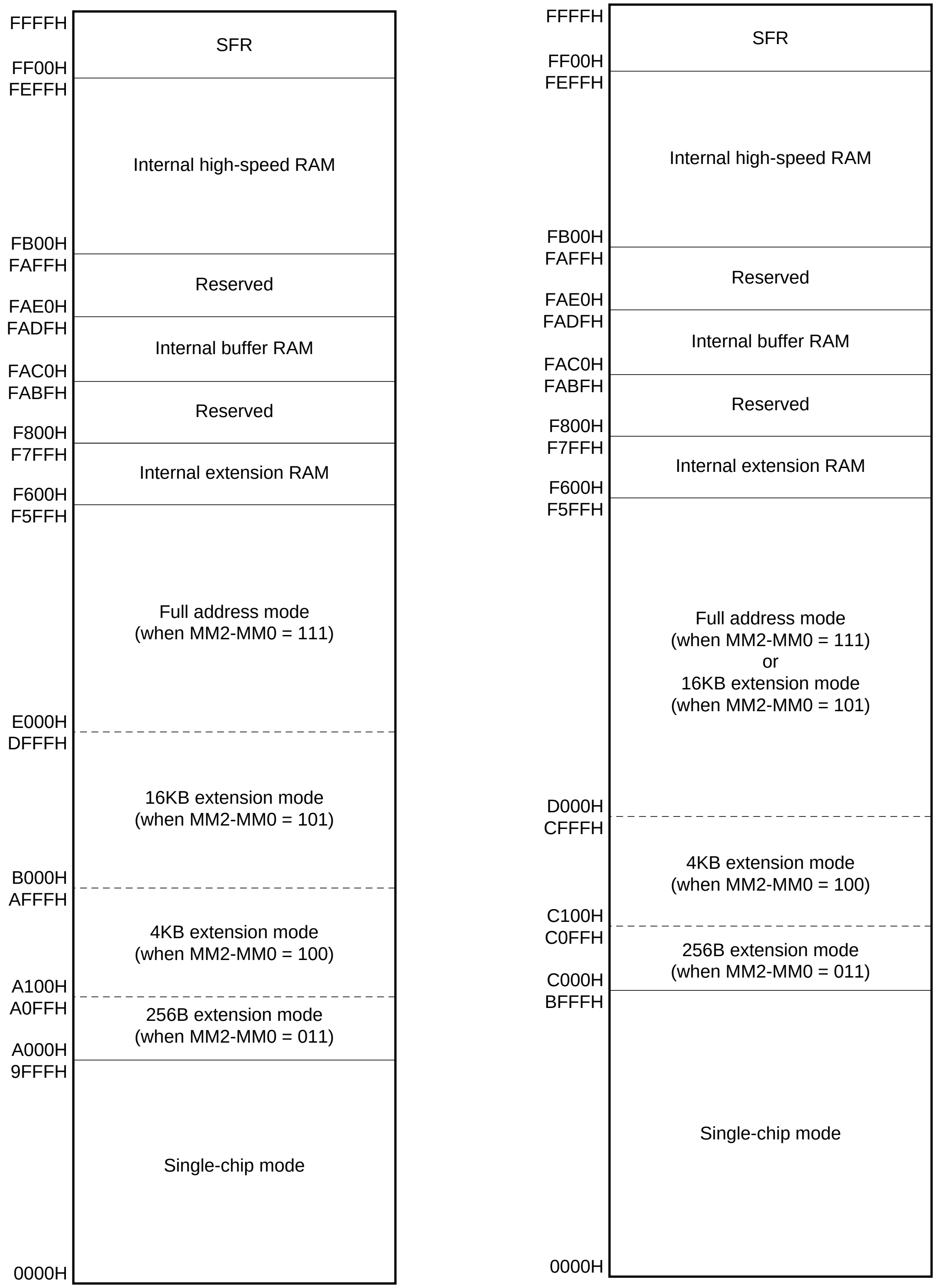
Figure 19-1. Memory Map when External Device Extension Function Is Used (2/4)



(c) Memory map of μPD78013F, 78013FY and μPD78P018F, 78P018FY with 24 KB internal PROM

(d) Memory map of μPD78014F, 78014FY and μPD78P018F, 78P018FY with 32 KB internal PROM

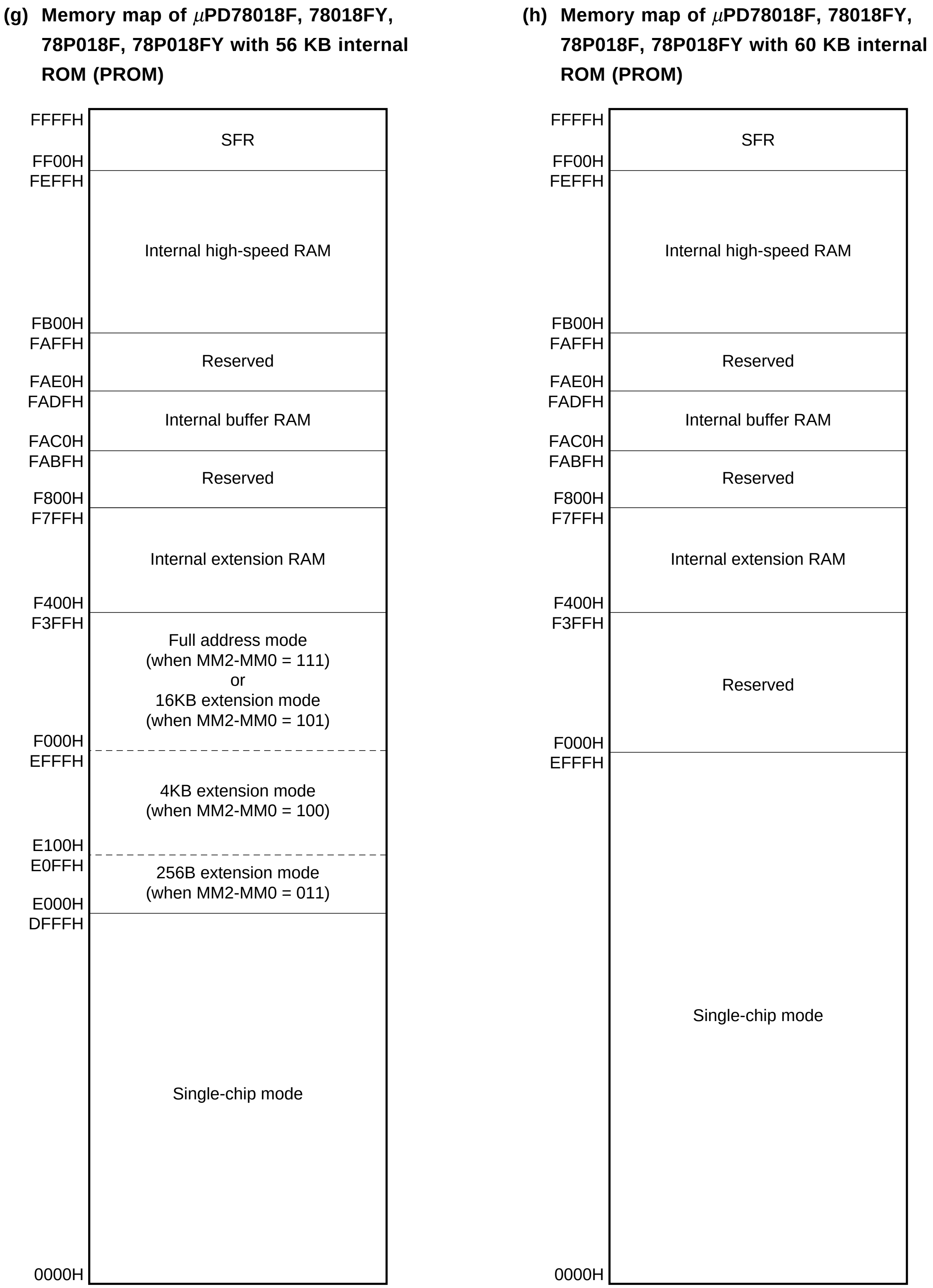
Figure 19-1. Memory Map when External Device Extension Function Is Used (3/4)



(e) Memory map of μPD78015F, 78015FY and μPD78P018F, 78P018FY with 40 KB internal PROM

(f) Memory map of μPD78016F, 78016FY and μPD78P018F, 78P018FY with 48 KB internal PROM

Figure 19-1. Memory Map when External Device Extension Function Is Used (4/4)



(g) Memory map of μPD78018F, 78018FY, 78P018F, 78P018FY with 56 KB internal ROM (PROM)

(h) Memory map of μPD78018F, 78018FY, 78P018F, 78P018FY with 60 KB internal ROM (PROM)

Caution When the internal ROM (PROM) capacity is 60 KB, the area F000H-F3FFH is reserved. This area can be used as an external memory by setting the internal ROM (PROM) capacity to 56 KB or less by using the memory size select register (IMS).

19.2 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 a 1-bit or an 8-bit memory manipulation instruction.

Its value is set to 10H at $\overline{\text{RESET}}$.

Figure 19-2. Format of Memory Extension Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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-P47, P50-P57, P64-P67 pins					
					P40-P47		P50-P53	P54, P55	P56, P57	P64-P67
0	0	0	Single-chip mode		Port mode	Input	Port mode			
0	0	1				Output				
0	1	1	Memory extension mode	256B mode	AD0-AD7		Port mode			P64 = $\overline{\text{RD}}$ P65 = $\overline{\text{WR}}$ P66 = $\overline{\text{WAIT}}$ P67 = ASTB
1	0	0		4KB mode			A8-A11	Port mode		
1	0	1		16KB mode				A12, A13	Port mode	
1	1	1		Full[Note] address mode					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 64K address space, except the internal ROM, RAM, SFR, and unused areas, can be externally extended.

Remark The P60-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. When the external device extension function of the μPD78018F or 78018FY is used, set the internal ROM capacity to 56K bytes or less.

IMS is set by using an 8-bit memory manipulation instruction.

The value of this register is as shown in Table 19-3 at $\overline{\text{RESET}}$.

Figure 19-3. Format of Memory Size Select Register

Symbol	7	6	5	4	3	2	1	0	Address	At 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	8K
0	1	0	0	16K bytes
0	1	1	0	24K bytes
1	0	0	0	32K bytes
1	0	1	0	40K bytes
1	1	0	0	48K bytes
1	1	1	0	56K bytes
1	1	1	1	60K bytes
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 19-3**).

Table 19-3. Value of Memory Size Select Register on Reset

Part Number	Set Value of IMS
μPD78011F, 78011FY	42H
μPD78012F, 78012FY	44H
μPD78013F, 78013FY	C6H
μPD78014F, 78014FY	C8H
μPD78015F, 78015FY	CAH
μPD78016F, 78016FY	CCH
μPD78018F, 78018FY	CFH

19.3 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-AD7, A8-A15 pins (shared by P40-P47, P50-P57)

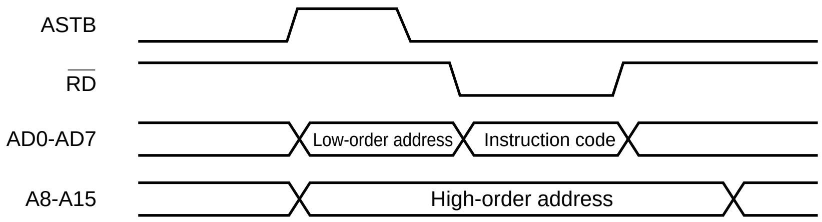
These pins output address and data signals. The valid signals are output or input when instructions are fetched or data is accessed from the external memory. The status of the signal also changes when the internal memory is accessed.

(the output contents are undefined)

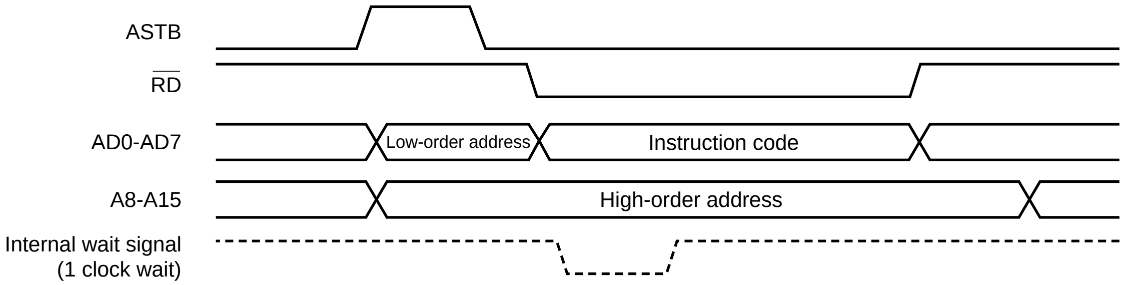
Figures 19-4 through 19-7 show the timing charts.

Figure 19-4. 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)

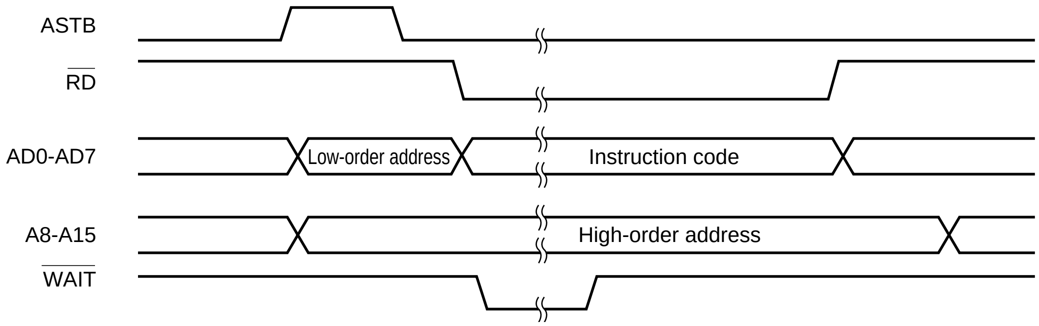
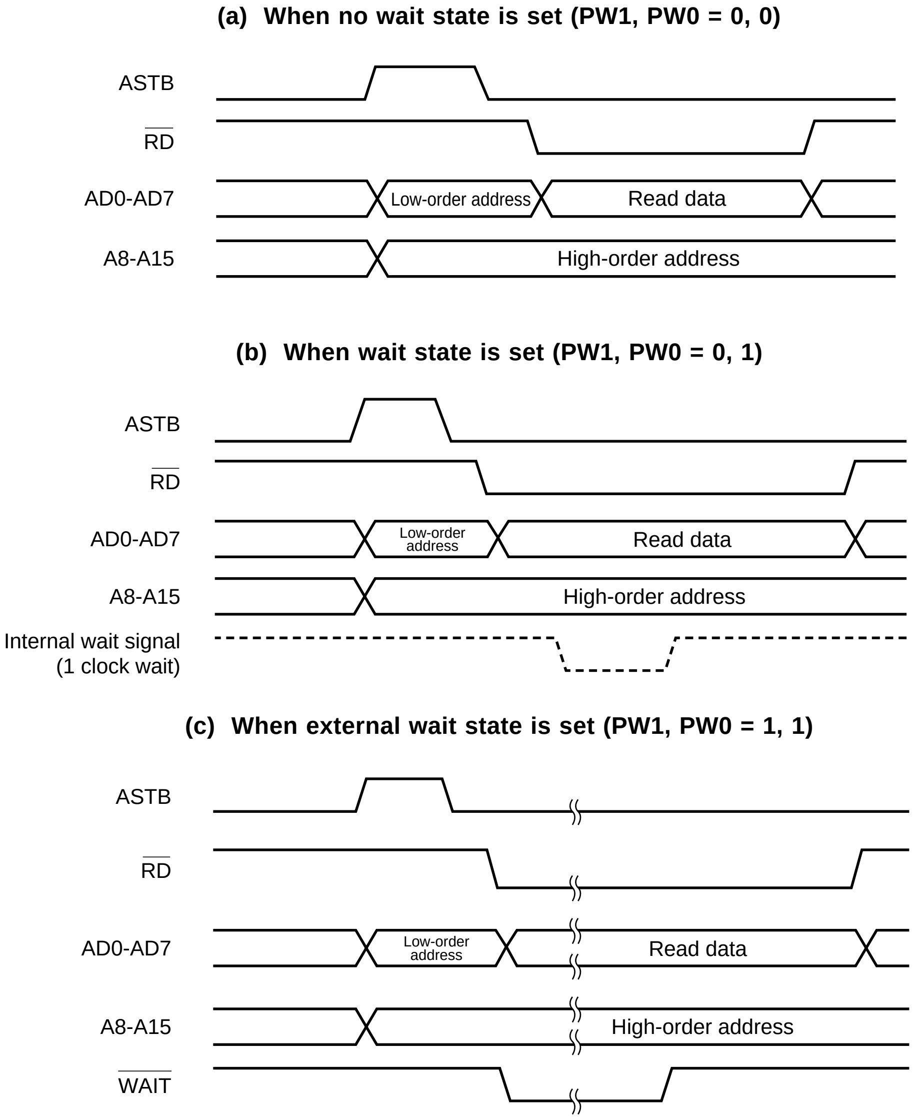


Figure 19-5. Read Timing of External Memory

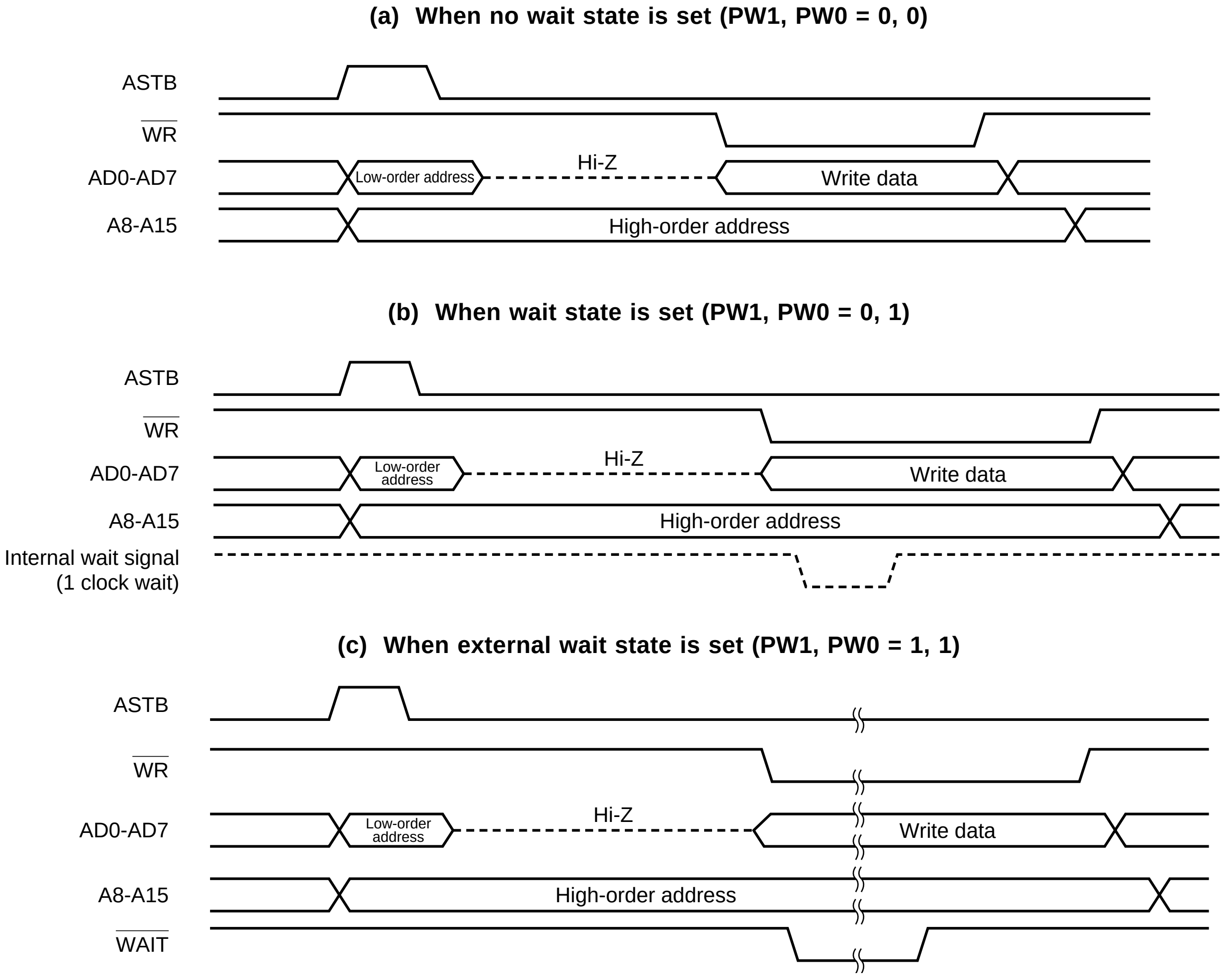


(a) When no wait state is set (PW1, PW0 = 0, 0)

(b) When wait state is set (PW1, PW0 = 0, 1)

(c) When external wait state is set (PW1, PW0 = 1, 1)

Figure 19-6. Write Timing of External Memory

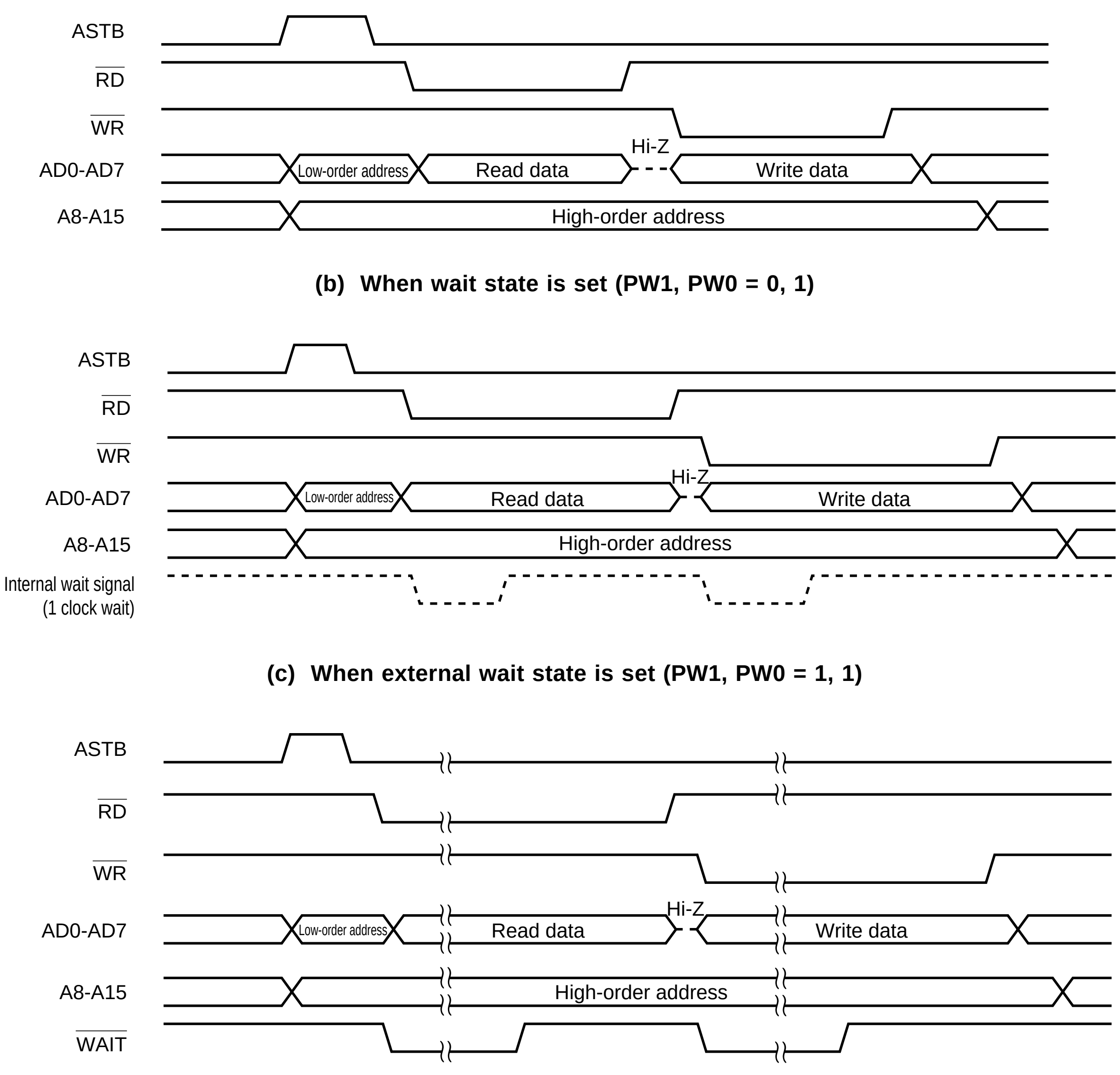


(a) When no wait state is set (PW1, PW0 = 0, 0)

(b) When wait state is set (PW1, PW0 = 0, 1)

(c) When external wait state is set (PW1, PW0 = 1, 1)

Figure 19-7. Read-Modify-Write Timing of External Memory



(a) When no wait state is set (PW1, PW0 = 0, 0)

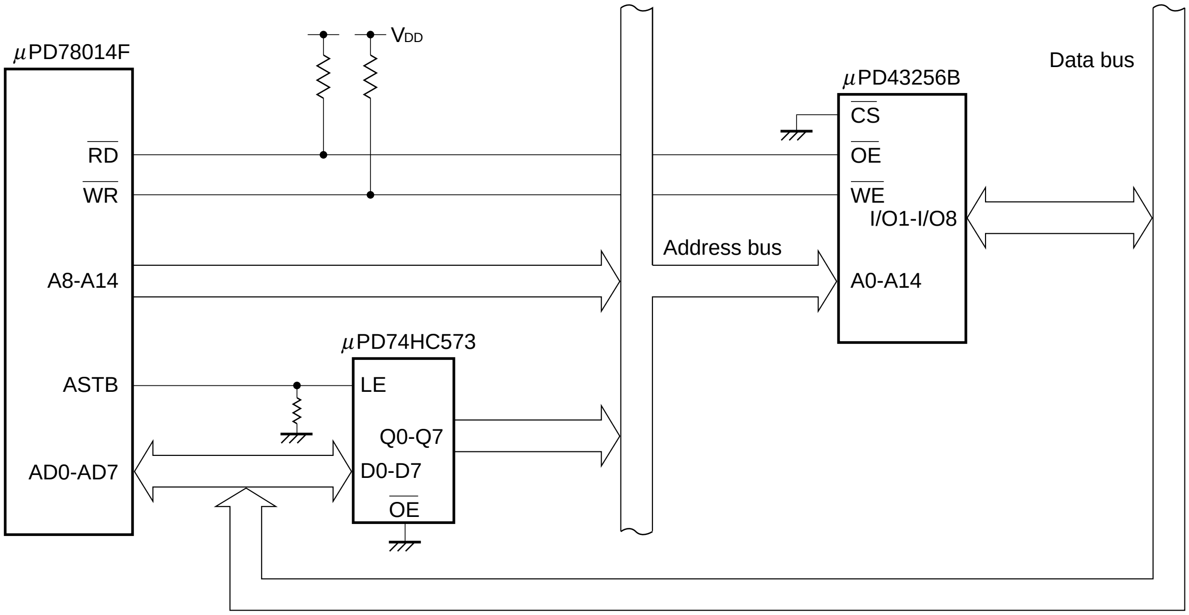
(b) When wait state is set (PW1, PW0 = 0, 1)

(c) When external wait state is set (PW1, PW0 = 1, 1)

19.4 Example of Connection with Memory

Figure 19-8 shows an example of connecting the μPD78014F and external memories. In this application example, SRAM is connected. In addition, the external device extension function is used in the full address mode, and 32K bytes of addresses, 0000H through 7FFFH, are allocated to internal ROM; addresses 8000H and higher are allocated to SRAM.

Figure 19-8. Example of Connecting μPD78014F and Memories



[MEMO]

CHAPTER 20 STANDBY FUNCTION

20.1 Standby Function and Configuration

20.1.1 Standby function

The standby function is to reduce the power consumption 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 current consumption 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 current consumption 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 consumption 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.

20.1.2 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.

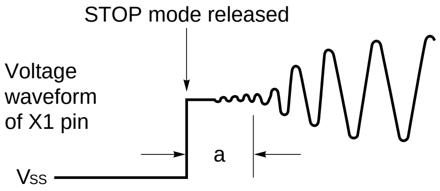
This register is set to 04H when the $\overline{\text{RESET}}$ signal is input. Therefore, to release the STOP mode by inputting the $\overline{\text{RESET}}$ signal, the time required to release the mode is $2^{18}/f_X$.

Figure 20-1. Format of Oscillation Stabilization Time Select Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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	$2^{13}/f_X$ (819μs)
0	0	1	$2^{15}/f_X$ (3.28ms)
0	1	0	$2^{16}/f_X$ (6.55ms)
0	1	1	$2^{17}/f_X$ (13.1ms)
1	0	0	$2^{18}/f_X$ (26.2ms)
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

20.2 Operation of Standby Function

20.2.1 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 20-1. Operation Status in HALT Mode (1/2)

(a) When HALT instruction is executed while system operates on main system clock

Setting of HALT Mode Item		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 fx/2^8 selected as count clock	Operatable
Serial interface	Other than automatic transmit/receive function	Operable	
	Automatic transmit/receive function	Stops operation	
External interrupt	INTP0	Operable when clock to peripheral hardware (fx/2^6, fx/2^7) selected as sampling clock	
	INTP1-INTP3	Operable	
Externally extended bus line	AD0-AD7	High impedance	
	A8-A15	Retains previous status before setting HALT mode	
	ASTB	Low level	
	$\overline{\text{WR}}$, $\overline{\text{RD}}$	High level	
	$\overline{\text{WAIT}}$	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 20-1. 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 transmit/receive function	Operable	Operable when external clock selected
	Automatic transmit/receive 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-INTP3	Operable	
Externally extended bus line	AD0-AD7	High impedance	
	A8-A15	Retains previous status before setting HALT mode	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

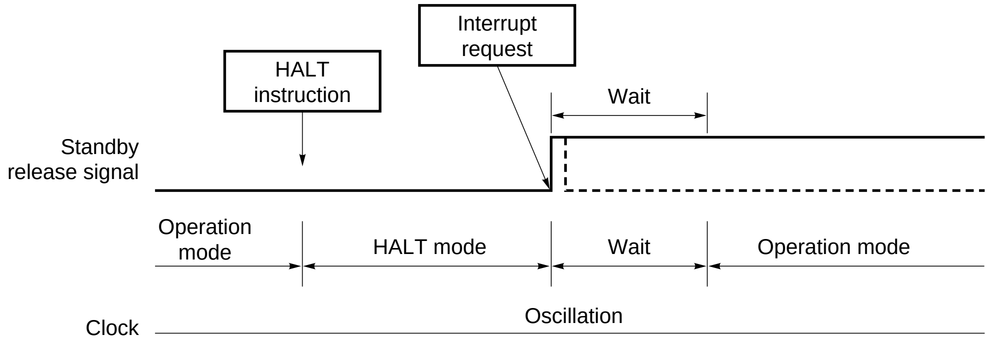
(2) Releasing HALT mode

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 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 20-2. Releasing HALT Mode by Interrupt Request



Remarks 1. The dotted 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

The HALT mode is released by a non-maskable interrupt request regardless of whether the interrupt request is enabled or disabled, and vectored interrupt processing is performed.

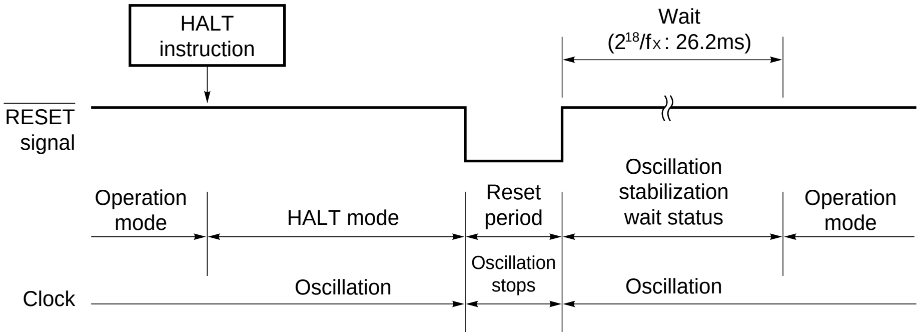
(c) Releasing by unmasked test input

The HALT mode is released by an unmasked test signal input, and 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 the $\overline{\text{RESET}}$ signal input, and execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 20-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx : Main system clock oscillation frequency

2. ( ) : At fx = 10.0 MHz operation

Table 20-2. 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

20.2.2 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 20-3. Operation Status in STOP Mode

Setting of STOP Mode Item		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 transmit/receive function	Operable only when external input clock is selected as serial clock	
	Automatic transmit/receive function	Stops operation	
External interrupt	INTP0	Cannot operate	
	INTP1-INTP3	Operable	
Externally extended bus line	AD0-AD7	High impedance	
	A8-A15	Retains previous status immediately before STOP instruction execution	
	ASTB	Low level	
	$\overline{WR}$, $\overline{RD}$	High level	
	$\overline{WAIT}$	High impedance	

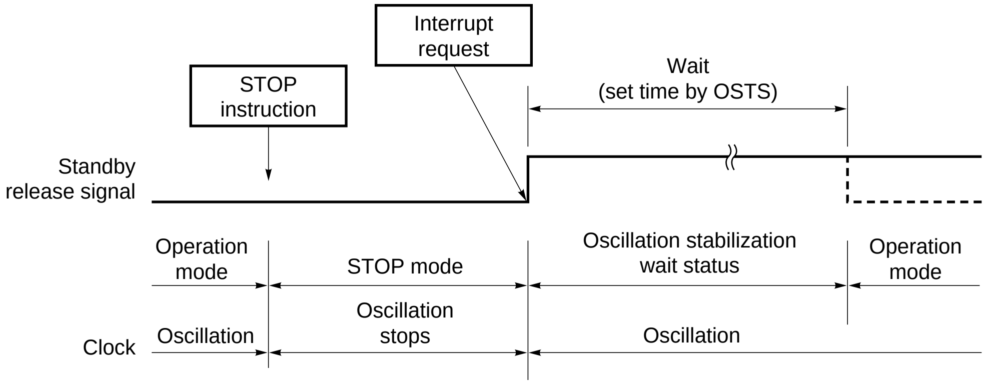
(2) Releasing STOP mode

The STOP mode can be released by the following three types of sources:

(a) Releasing by unmasked interrupt request

The STOP mode is released by 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 is disabled to be accepted, the instruction at the next address is executed.

Figure 20-4. Releasing STOP Mode by Interrupt Request



Remark The dotted line indicates the case where the interrupt request that has released the standby mode is accepted.

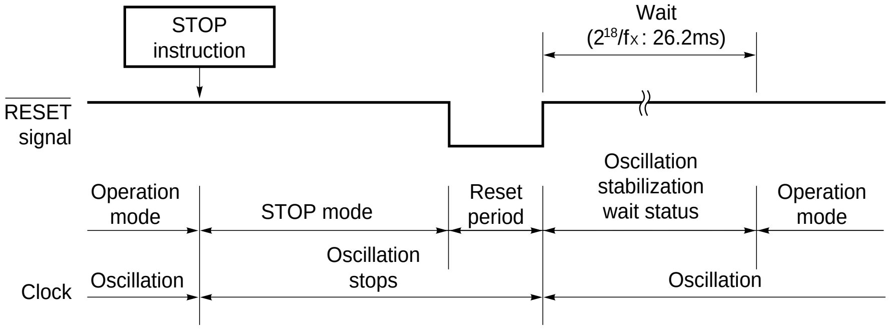
(b) Releasing by unmasked test input

The STOP mode is released by an unmasked test signal input, and 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 the $\overline{\text{RESET}}$ signal input, and the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 20-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx : Main system clock oscillation frequency

2. ( ): At fx = 10.0 MHz operation

Table 20-4. 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

[MEMO]

CHAPTER 21 RESET FUNCTION

21.1 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 inadvertent loop time detection by watchdog timer

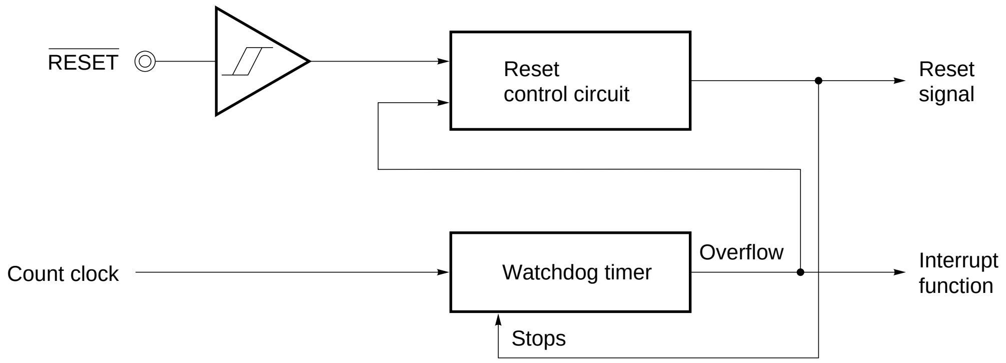
There is no functional difference between the external reset and internal reset, and execution of the program is started from addresses written to addresses 0000H and 0001H when the $\overline{\text{RESET}}$ signal is input.

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 21-1. 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 21-2** through **21-4**).

Cautions 1. Input a low-level signal to the $\overline{\text{RESET}}$ pin for 10 μs or longer to execute external reset.
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 21-1. Block Diagram of Reset Function



Phase-out/Discontinued

Figure 21-2. Reset Timing by $\overline{\text{RESET}}$ Input

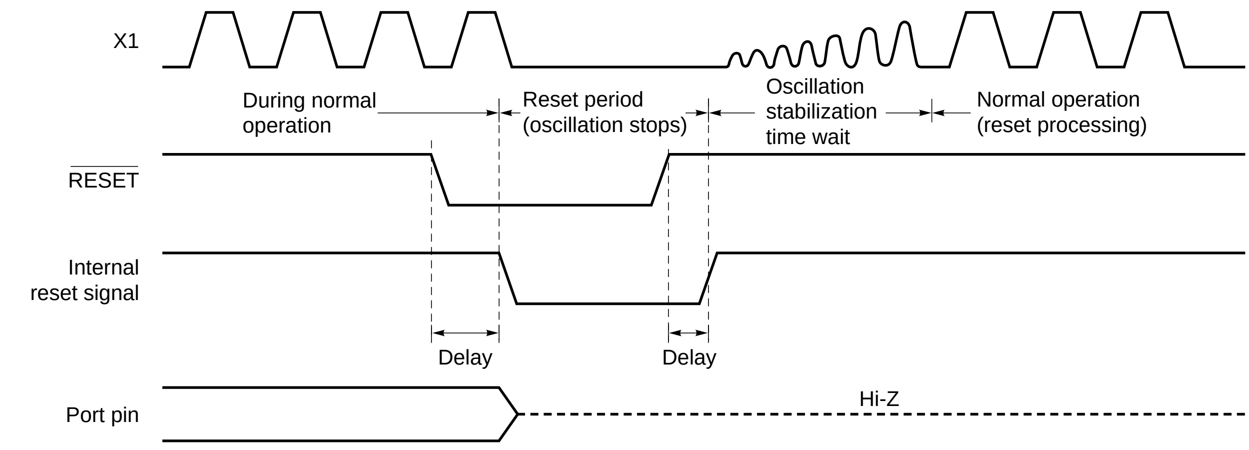

X1
During normal operation
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 21-3. Reset Timing by Overflow in Watchdog Timer

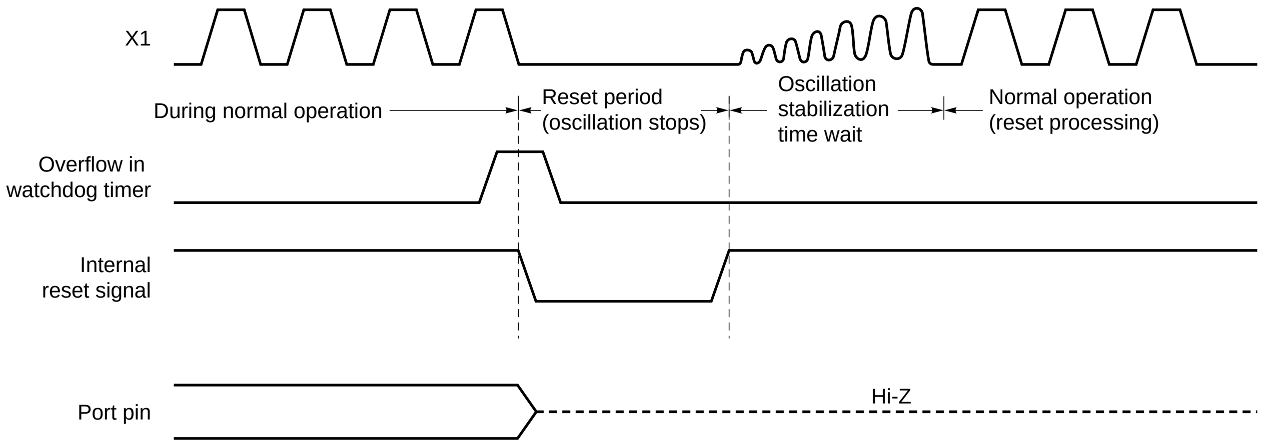

X1
During normal operation
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
Overflow in watchdog timer
Internal reset signal
Port pin
Hi-Z

Figure 21-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

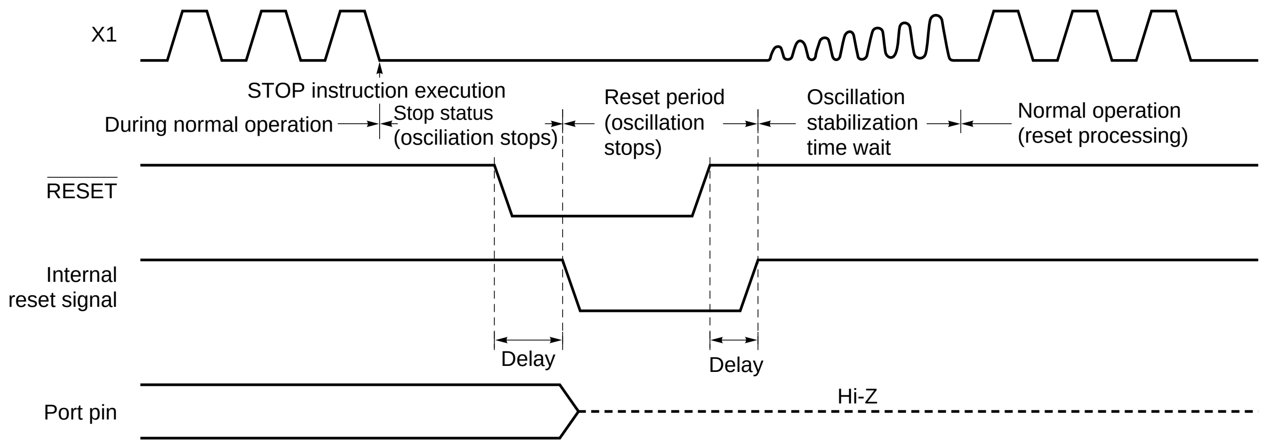

X1
STOP instruction execution
During normal operation
Stop status (osciliation stops)
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Table 21-1. 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 0-3 (P0-P3)	00H
	Ports 4-6 (P4-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**
Internal extension RAM size select register (IXS)		**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. The status before reset is retained in the standby mode.

3. The values at reset of the memory size select register (IMS) and internal extension RAM size select register (IXS) differ depending on the model, as follows:

	μPD78011F μPD78011FY	μPD78012F μPD78012FY	μPD78013F μPD78013FY	μPD78014F μPD78014FY	μPD78015F μPD78015FY	μPD78016F μPD78016FY	μPD78018F μPD78018FY	μPD78P018F μPD78P018FY
IMS	42H	44H	C6H	C8H	CAH	CCH	CFH	
IXS	0CH				0BH		0AH	

When using the mask ROM model, do not set a value other than that at reset to IMS and IXS, except when the external device extension function of the μPD78018F and 78018FY is used.

Table 21-1. 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 transmit/receive control register (ADTC)	00H
	Automatic data transmit/receive interval specification register (ADTI)	00H
	Automatic data transmit/receive 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
ROM correction	Correction address register (CORAD0, CORAD1)Note	0000H
	Correction control register (CORCN)Note	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

Note Provided to μPD78015F, 78015FY, 78016F, 78018F, 78018FY, 78016FY, 78P018F, 78P018FY only.

CHAPTER 22 ROM CORRECTION

22.1 Function of ROM Correction

The μPD78015F, 78015FY, 78016F, 78016FY, 78018F, and 78018FY can execute a part of the program in the mask ROM, replacing it with the program in the internal extension RAM.

By using ROM correction, instruction bugs found in the mask ROM can be avoided or the flow of the program can be changed.

ROM correction can be used in up to two places in the internal ROM (program).

Caution ROM correction cannot be emulated by an in-circuit emulator (IE-78000-R, IE-78000-R-A, IE-78001-R-A, and IE-78K0-NS).

22.2 Configuration of ROM Correction

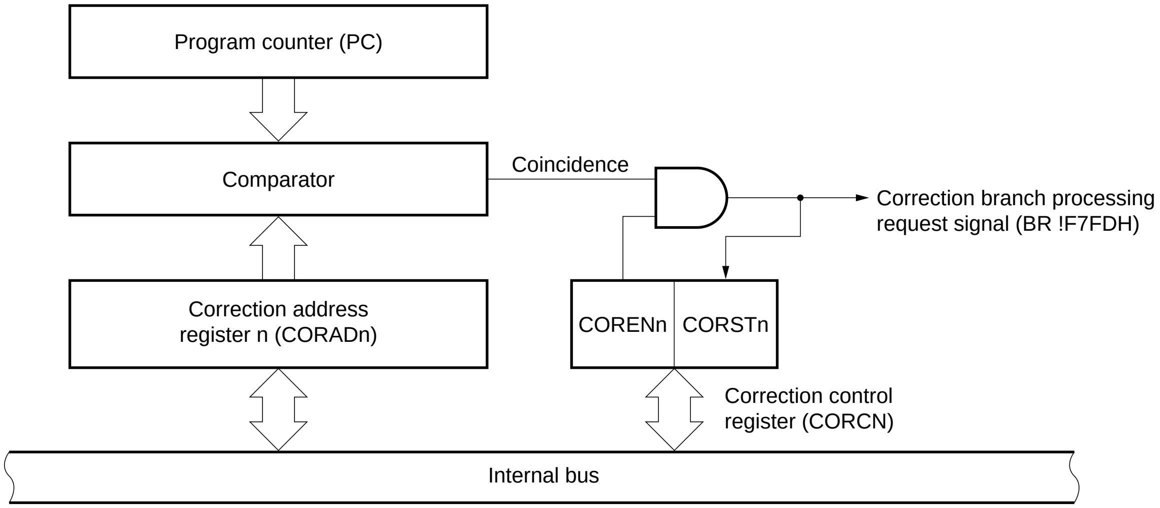
ROM correction consists of the following hardware:

Table 22-1. Configuration of ROM Correction

Item	Configuration
Register	Correction address registers 0 and 1 (CORAD0 and CORAD1)
Control register	Correction control register (CORCN)

Figure 22-1 shows the block diagram of ROM correction.

Figure 22-1. Block Diagram of ROM Correction


Program counter (PC)
Comparator
Coincidence
Correction branch processing request signal (BR !F7FDH)
Correction address register n (CORADn)
CORENn
CORSTn
Correction control register (CORCN)
Internal bus

Remark n = 0, 1

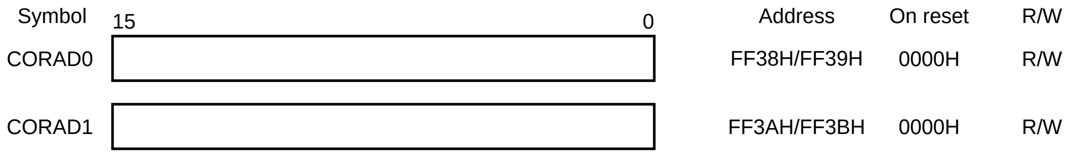
(1) Correction address registers 0 and 1 (CORAD0 and CORAD1)

These registers set the first address (correction address) of the instruction in the mask ROM to be corrected. ROM correction can correct up to two places in a program. Therefore, two addresses can be set in CORAD0 and CORAD1. To correct only one place, put the address in either of the registers.

CORAD0 and CORAD1 are set by using a 16-bit memory manipulation instruction.

These registers are set to 0000H when the $\overline{\text{RESET}}$ signal is input.

Figure 22-2. Formats of Correction Address Registers 0 and 1



Symbol	Address	On reset	R/W
CORAD0	FF38H/FF39H	0000H	R/W
CORAD1	FF3AH/FF3BH	0000H	R/W

Cautions 1. Set CORAD0 and CORAD1 when bits 1 (COREN0) and 3 (COREN1) of the correction control register (CORCN: refer to Figure 22-3) are 0.

2. Only an address to an instruction code can be set in CORAD0 and CORAD1.

3. Do not put the following addresses in CORAD0 and CORAD1:

- **Address value of table area of table reference instruction (CALLT instruction): 0040H through 007FH**
- **Address value of vector table area: 0000H through 003FH**

(2) Comparator

The comparator always compares the correction address values in correction address registers 0 and 1 (CORAD0 and CORAD1) with the fetch address value. If the correction address coincides with the fetch address value when bit 1 (COREN0) or bit 3 (COREN1) of the correction control register (CORCN) is 1, the ROM correction circuit generates a correction branch processing request signal (BR !F7FDH).

22.3 Registers Controlling ROM Correction

ROM correction is controlled by the correction control register (CORCN).

(1) Correction control register (CORCN)

This register controls generation of the correction branch processing request signal when the correction address in correction address register 0 or 1 coincides with the fetch address. It consists of correction enable flags (COREN0 and COREN1) that enable or disable detection of coincidence by the comparator and correction status flags (CORST0 and CORST1) that indicate coincidence.

CORCN 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 22-3. Format of Correction Control Register

Symbol	7	6	5	4	<3>	<2>	<1>	<0>	Address	On reset	R/W
CORCN	0	0	0	0	COREN1	CORST1	COREN0	CORST0	FF8AH	00H	R/W[Note]

CORST0	Flag detecting coincidence between correction address register 0 and fetch address
0	Coincidence not detected
1	Coincidence detected

COREN0	Controls detection of coincidence between correction address register 0 and fetch address
0	Detection disabled
1	Detection enabled

CORST1	Flag detecting coincidence between correction address register 1 and fetch address
0	Coincidence not detected
1	Coincidence detected

COREN1	Controls detection of coincidence between correction address register 1 and fetch address
0	Detection disabled
1	Detection enabled

Note Bits 0 and 2 are read-only bits.

22.4 Using ROM Correction

<1> Store the correction address and corrected instruction (corrected program) to an external non-volatile memory (EEPROM™).
If two places must be corrected, a branch destination identification program is also stored. This program identifies at which of the two addresses in CORAD0 or CORAD1 correction branch processing occurred.

Figure 22-4. Example of Storing in EEPROM (when only one place is to be corrected)

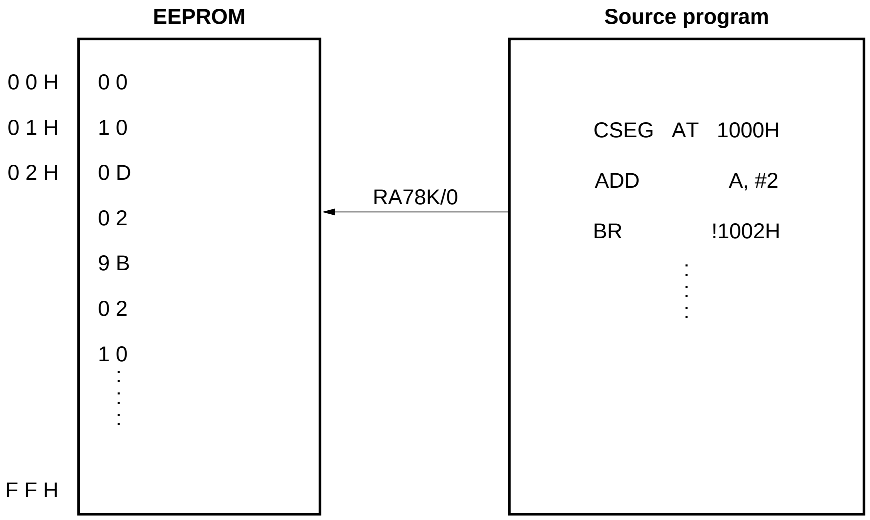
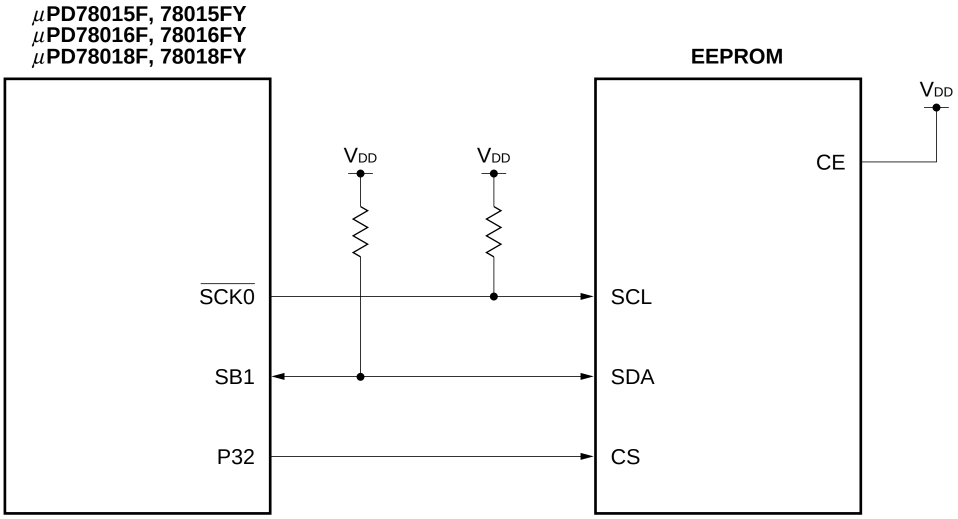
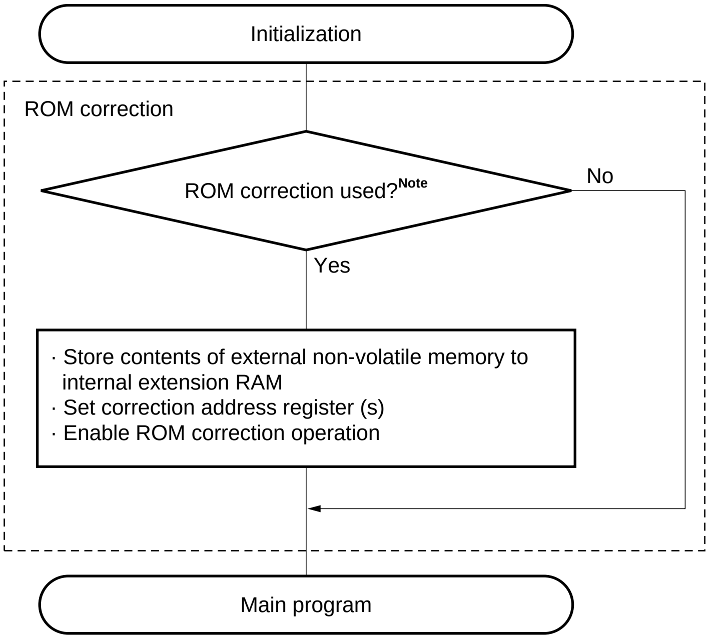


Figure 22-5. Example of Connecting EEPROM (in 2-wire serial I/O mode)



<2> Develop an initialization routine in advance such as the one shown in Figure 22-6, so that the program can be corrected.

Figure 22-6. Initialization Routine


Initialization
ROM correction
ROM correction used?Note
No
Yes
· Store contents of external non-volatile memory to internal extension RAM
· Set correction address register (s)
· Enable ROM correction operation
Main program

Note Whether ROM correction is used is determined by the input level of a port. For example, “if the input level of P20 is high, ROM correction is used; if it is low, ROM correction is not used”.

<3> After reset, store the contents of an external non-volatile memory to the internal extension RAM by using the user’s initialization routine for ROM correction (refer to **Figure 22-6**).
Also put the first address of the instruction to be corrected in CORAD0 or CORAD1, and set bits 1 and 3 (COREN0 and COREN1) of the correction control register (CORCN) to 1.

<4> Place the whole-address-space branch instruction (BR !addr16) at the specific address (F7FDH) in the internal extension RAM by using the main program.

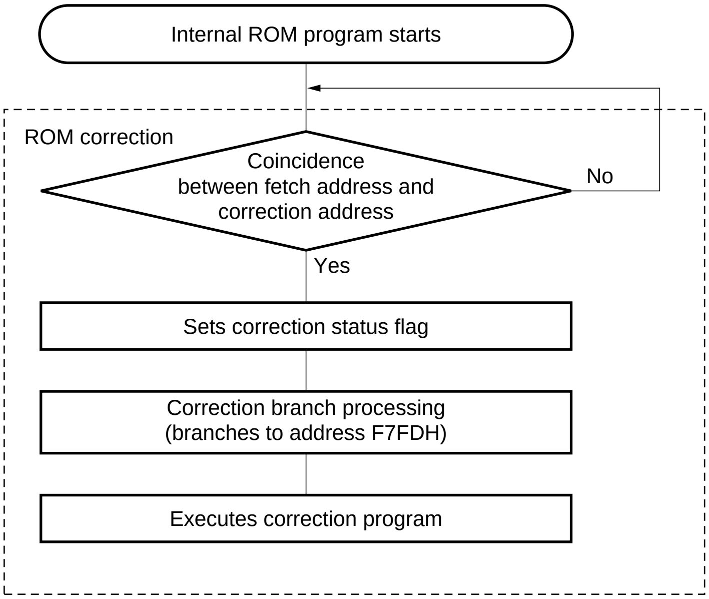
<5> After the main program has been started, the comparator in the ROM correction circuit always compares the value in CORAD0 or CORAD1 with the fetch address value. When the two address values coincide, a correction branch processing request signal is generated. At the same time, the correction status flags (CORST0 and CORST1) are set to 1.

<6> Execution branches to address F7FDH by the correction branch processing request signal.

<7> The whole-address-space branch instruction at address F7FDH causes execution to branch to the address in the internal extension RAM set by the main program.

<8> If only one place is to be corrected, execute the correction program.
If two places must be corrected, check the correction status flag by operation of the branch destination identification program, and then branch to the correction program.

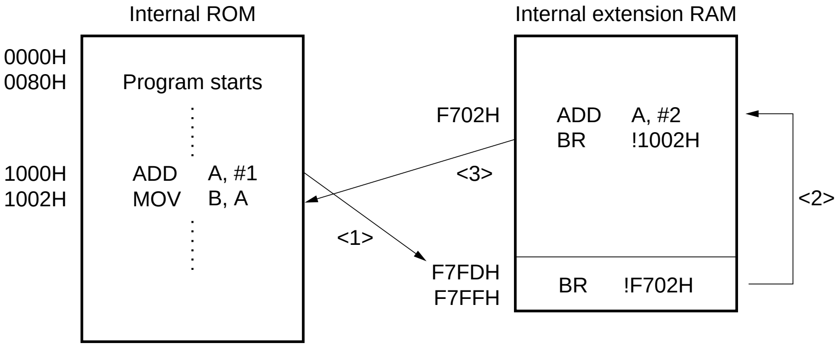
Figure 22-7. Operation of ROM Correction


Internal ROM program starts
ROM correction
Coincidence between fetch address and correction address
No
Yes
Sets correction status flag
Correction branch processing (branches to address F7FDH)
Executes correction program

22.5 Example of Using ROM Correction

Here is an example of using ROM correction to change instruction “ADD A, #1” at address 1000H to “ADD A, #2”.

Figure 22-8. Example of Using ROM Correction



<1> If 1000H, placed in advance in the correction address register, coincides with the fetch address value after the main program has started, execution branches to address F7FDH.

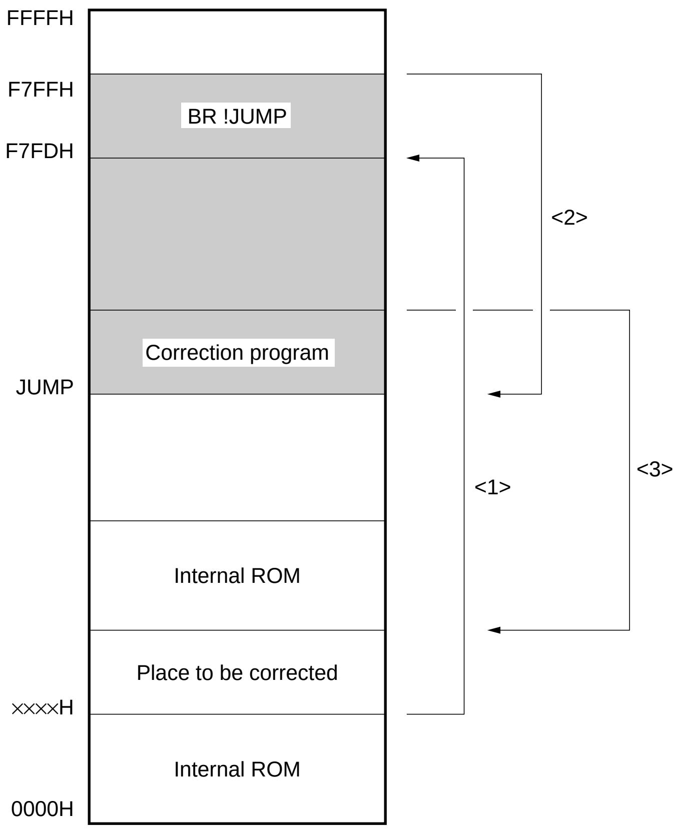
<2> By placing the whole-address-space branch instruction (BR !addr16) at address F7FDH by operation of the main program, execution branches to any address (address F702H in this example).

<3> After executing the alternate instruction ADD A, #2, execution returns to the internal ROM program.

22.6 Program Execution Flow

Figures 22-9 and 22-10 show the sequence of program execution when ROM correction is used.

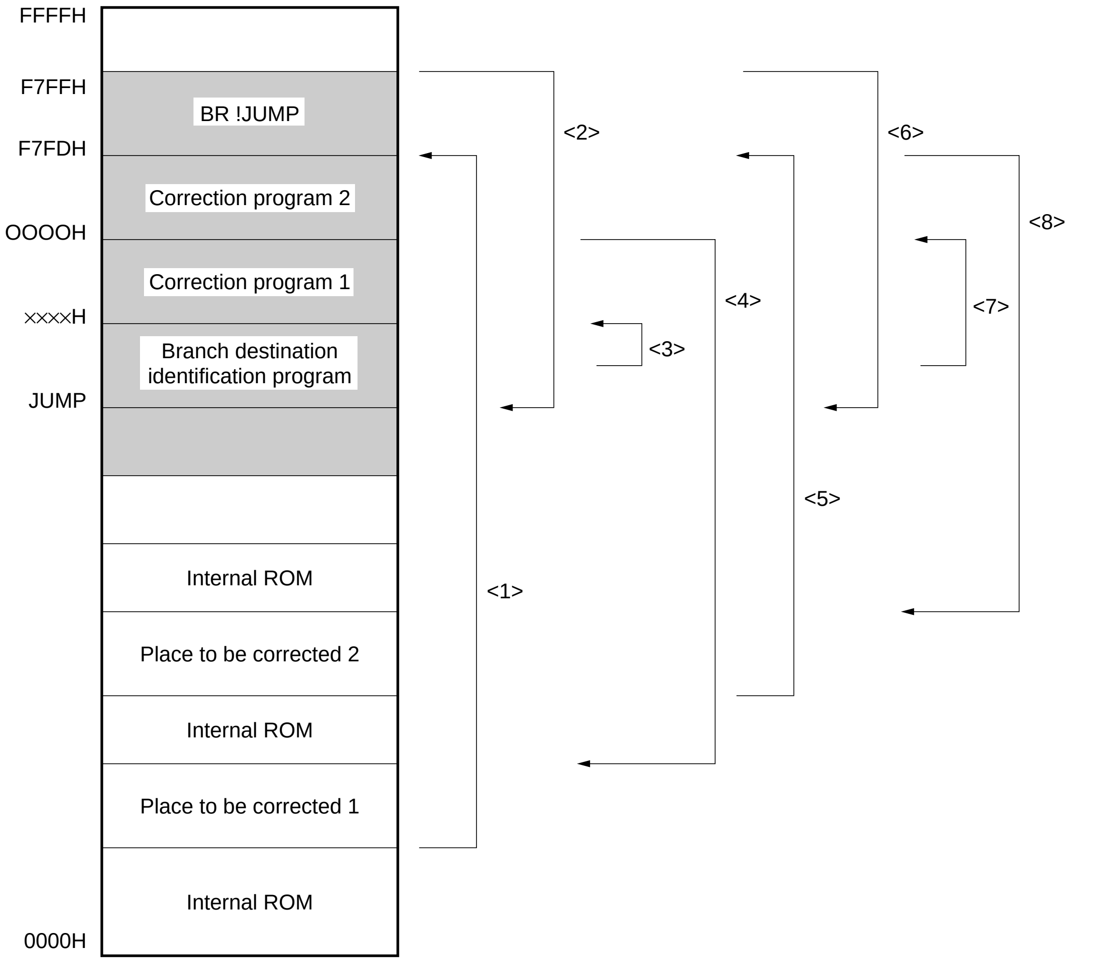
Figure 22-9. Program Sequence (when only one place is corrected)



<1> Execution branches to address F7FDH when the fetch address and correction address coincide.
<2> Execution branches to the correction program.
<3> Execution returns to the internal ROM program.

Remark ▨ : internal extension RAM
JUMP : correction program start address

Figure 22-10. Program Sequence (when two places are corrected)



<1> Execution branches to address F7FDH when the fetch address and correction address coincide.

<2> Execution branches to the branch destination identification program.

<3> The branch destination identification program (BTCLR !CORST0, $××××H) causes execution to branch to correction program 1.

<4> Execution returns to the internal ROM program.

<5> Execution branches to address F7FDH when the fetch address and correction address coincide.

<6> Execution branches to the branch destination identification program.

<7> The branch destination identification program (BTCLR !CORST1, $○○○○H) causes execution to branch to correction program 2.

<8> Execution returns to the internal ROM program.

Remark ▨ : internal extension RAM

JUMP : correction program start address

22.7 Notes on ROM Correction

(1) Be sure to set the value of an address where an instruction code is stored in correction address registers 0 and 1 (CORAD0 and CORAD1).

(2) Set correction address registers 0 and 1 (CORAD0 and CORAD1) when the corresponding correction enable flags (COREN0 and COREN1) are 0 (i.e., when correction branch processing is disabled). If an address is put into CORAD0 or CORAD1 when COREN0 or COREN1 is 1 (when correction branch processing is enabled), there is a possibility that correction branch processing will be started from an address different from that intended.

(3) Do not put an instruction address immediately following an instruction that sets a correction enable flag (COREN0 or COREN1) into a correction address registers 0 or 1 (CORAD0 or CORAD1) (because correction branch processing may not be started).

(4) Do not put an address value (0040H to 007FH) in the table area of the table reference instruction (CALLT), or an address value (0000H to 003FH) in the vector table area into a correction address register 0 or 1 (CORAD0 or CORAD1).

(5) Do not put the two addresses immediately following any of the instructions below into a correction address register 0 or 1 (CORAD0 or CORAD1) (if the destination address to which these instructions are mapped is N, do not use addresses N+1 and N+2).

- RET
- RETI
- RETB
- BR $addr16
- STOP
- HALT

CHAPTER 23 μPD78P018F, 78P018FY

The μPD78P018F and 78P018FY are provided with a one-time PROM which can be written only once, or an EPROM from/to which a program can be written, erased, and rewritten. The differences between the PROM models (μPD78P018F, 78P018FY) and the mask ROM models (μPD78011F, 78011FY, 78012F, 78012FY, 78013F, 78013FY, 78014F, 78014FY, 78015F, 78015FY, 78016F, 78016FY, 78018F, and 78018FY) are shown in Table 23-1.

Table 23-1. Differences between μPD78P018F, 78P018FY, and Mask ROM Models

Item	μPD78P018F, 78P018FY	Mask ROM Model
Internal ROM structure	One-time PROM/EPROM	Mask ROM
Internal ROM capacity	60K bytes	μPD78011F, 78011FY: 8K bytes μPD78012F, 78012FY: 16K bytes μPD78013F, 78013FY: 24K bytes μPD78014F, 78014FY: 32K bytes μPD78015F, 78015FY: 40K bytes μPD78016F, 78016FY: 48K bytes μPD78018F, 78018FY: 60K bytes
Internal high-speed RAM capacity	1024 bytes	μPD78011F, 78011FY: 512 bytes μPD78012F, 78012FY: 512 bytes μPD78013F, 78013FY: 1024 bytes μPD78014F, 78014FY: 1024 bytes μPD78015F, 78015FY: 1024 bytes μPD78016F, 78016FY: 1024 bytes μPD78018F, 78018FY: 1024 bytes
Internal extension RAM capacity	1024 bytes	μPD78011F, 78011FY: None μPD78012F, 78012FY: None μPD78013F, 78013FY: None μPD78014F, 78014FY: None μPD78015F, 78015FY: 512 bytes μPD78016F, 78016FY: 512 bytes μPD78018F, 78018FY: 1024 bytes
Changing internal ROM and internal high-speed RAM capacities by using memory size select register (IMS)	Yes**Note 1**	No
Changing of internal extended RAM capacity by using internal extension RAM size select register (IXS)	Yes**Note 2**	No
IC pin	None	Provided
V_{PP} pin	Provided	None
Mask option to contain pull-up resistor of P60-P63 pins	Not provided	Provided
Electrical characteristics and recommended soldering conditions	Refer to individual **Data Sheet**.	

Notes 1. The internal PROM is set to 60K bytes and internal high-speed RAM is set to 1024 bytes at $\overline{\text{RESET}}$.

2. The internal extension RAM is set to 1024 bytes at $\overline{\text{RESET}}$.

Caution The noise immunity and radiation differ between the PROM model and mask ROM model. To replace a PROM model with a mask ROM model in the course from experimental production to mass production, evaluate your system with the CS model (not ES model) of the mask ROM model.

23.1 Memory Size Select Register

The μPD78P018F and 78P018FY are provided with a memory size select register (IMS) that can select the internal memory. By setting IMS, the memory of the μPD78P018F and 78P018FY can be mapped in the same manner as that of the mask ROM models each having a different memory capacity from those of the others.

To map the memory of the μPD78P018F and 78P018FY in the same manner as that of a mask ROM model, set the value of the mask ROM model at reset to IMS.

IMS of a mask ROM model need not to be set.

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

This register is set as shown in Table 23-2 when the $\overline{\text{RESET}}$ signal is input.

Caution When using the mask ROM model, do not set a value other than the value at reset shown in Table 23-2 to IMS, except when the external device extension function of the μPD78018F and 78018FY is used.

Figure 23-1. Format of Memory Size Select Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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 KB
0	1	0	0	16 KB
0	1	1	0	24 KB
1	0	0	0	32 KB
1	0	1	0	40 KB
1	1	0	0	48 KB
1	1	1	0	56 KB$^{\text{Note}}$
1	1	1	1	60 KB
Others				Setting prohibited

RAM2	RAM1	RAM0	Selects internal high-speed RAM capacity
0	1	0	512 B
1	1	0	1024 B
Others			Setting prohibited

Note Set the internal ROM capacity to 56K bytes or less when using the external device extension function of the μPD78018F, 78018FY, 78P018F, or 78P018FY.

The set value of IMS to map the memory of the μPD78P018F and 78P018FY in the same manner as that of the mask ROM model is shown in Table 23-2.

Table 23-2. Set Value of Memory Size Select Register

Model	Set value of IMS
μPD78011F, 78011FY	42H
μPD78012F, 78012FY	44H
μPD78013F, 78013FY	C6H
μPD78014F, 78014FY	C8H
μPD78015F, 78015FY	CAH
μPD78016F, 78016FY	CCH
μPD78018F, 78018FY	CFH

23.2 Internal Extension RAM Size Select Register

The internal extension RAM of the μPD78P018F and 78P018FY can be mapped in the same manner as the memory map of a mask ROM model with a different sized internal extension RAM by setting the internal extension RAM size select register (IXS).

IXS of a mask ROM model does not need to be set.

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 To use a mask ROM model, do not set a value other than those listed in Table 23-3 to IXS.

Figure 23-2. Format of Internal Extension RAM Size Select Register

Symbol	7	6	5	4	3	2	1	0	Address	On 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 extension RAM capacity
1	0	1	0	1024 B (F400H-F7FFH)
1	0	1	1	512 B (F600H-F7FFH)
1	1	0	0	0 B
Others				Setting prohibited

The set value of IXS to map the internal extension RAM of the μPD78P018F and 78P018FY in the same manner as that of the mask ROM model is shown in Table 23-3.

Table 23-3. Set Value of Internal Extension RAM Size Select Register

Model	Value at reset
μPD78011F, 78011FY	0CH
μPD78012F, 78012FY	
μPD78013F, 78013FY	
μPD78014F, 78014FY	
μPD78015F, 78015FY	0BH
μPD78016F, 78016FY	
μPD78018F, 78018FY	0AH

23.3 PROM Programming

The μPD78P018F and 78P018FY are provided with 60 KB PROM as a program memory. This memory is set in the PROM programming mode, when it is programmed, by using the V_{PP} and $\overline{\text{RESET}}$ pins. When these pins are not used, process them by referring to **1.5** or **2.5 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.

23.3.1 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 $\overline{\text{RESET}}$ pin. In this mode, the operation modes shown in Table 23-4 can be set by using the $\overline{\text{CE}}$, $\overline{\text{OE}}$, and $\overline{\text{PGM}}$ pins.

By setting the read mode, the contents of the PROM can be read.

Table 23-4. Operation Modes for PROM Programming

Pin Operation Mode	$\overline{\text{RESET}}$	V_{PP}	V_{DD}	$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{PGM}}$	D0-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 $\overline{\text{CE}}$ = L and $\overline{\text{OE}}$ = L.

(2) Output disable mode

The data output goes into a high-impedance state and the output disable mode is set when $\overline{\text{OE}}$ = H.

When two or more μPD78P018F's or 78P018FY's are connected to the data bus, therefore, data can be read from any one of the devices by controlling the $\overline{\text{OE}}$ pin.

(3) Standby mode

The standby mode is set when $\overline{\text{CE}}$ = H.

In this mode, the data output goes into a high-impedance state regardless of the status of $\overline{\text{OE}}$.

(4) Page data latch mode

The page data latch mode is set when $\overline{\text{CE}}$ = H, $\overline{\text{PGM}}$ = H, and $\overline{\text{OE}}$ = 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 $\overline{\text{PGM}}$ pin with $\overline{\text{CE}}$ = H and $\overline{\text{OE}}$ = H after latching 1-page, 4-byte address and data in the page data latch mode. After that, the program can be verified when $\overline{\text{CE}}$ = L and $\overline{\text{OE}}$ = L.

If the program cannot be written by one program pulse, repeatedly execute write and verify X times ($X \leq 10$).

(6) Byte write mode

Byte write is executed when a 0.1-ms program pulse (active low) is applied to the $\overline{\text{PGM}}$ pin with $\overline{\text{CE}}$ = L and $\overline{\text{OE}}$ = H. After that, the program can be verified when $\overline{\text{OE}}$ = L.

If the program cannot be written by one program pulse, repeatedly execute write and verify X times ($X \leq 10$).

(7) Program verify mode

The program verify mode is set when $\overline{\text{CE}}$ = L, $\overline{\text{PGM}}$ = H, and $\overline{\text{OE}}$ = L.

Use this mode to confirm that the program has written correctly.

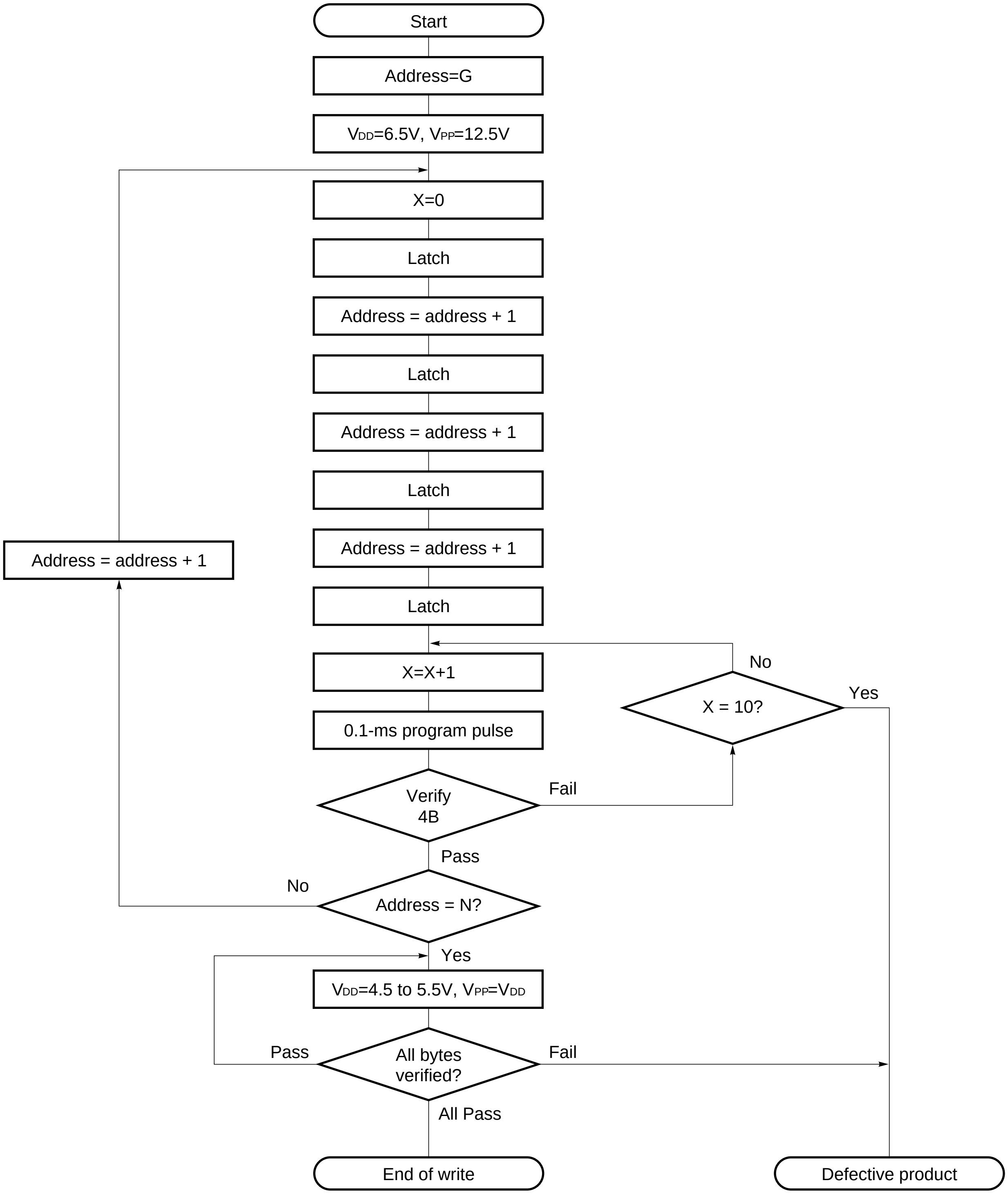
(8) Program inhibit mode

The program inhibit mode is used to write one device of the plural μPD78P018F's and 78P018FY's whose $\overline{\text{OE}}$, V_{PP}, and D0-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 $\overline{\text{PGM}}$ pin is high.

23.3.2 PROM write sequence

Figure 23-3. Page Program Mode Flowchart



G = start address

N = last address of program

Figure 23-4. Page Program Mode Timing

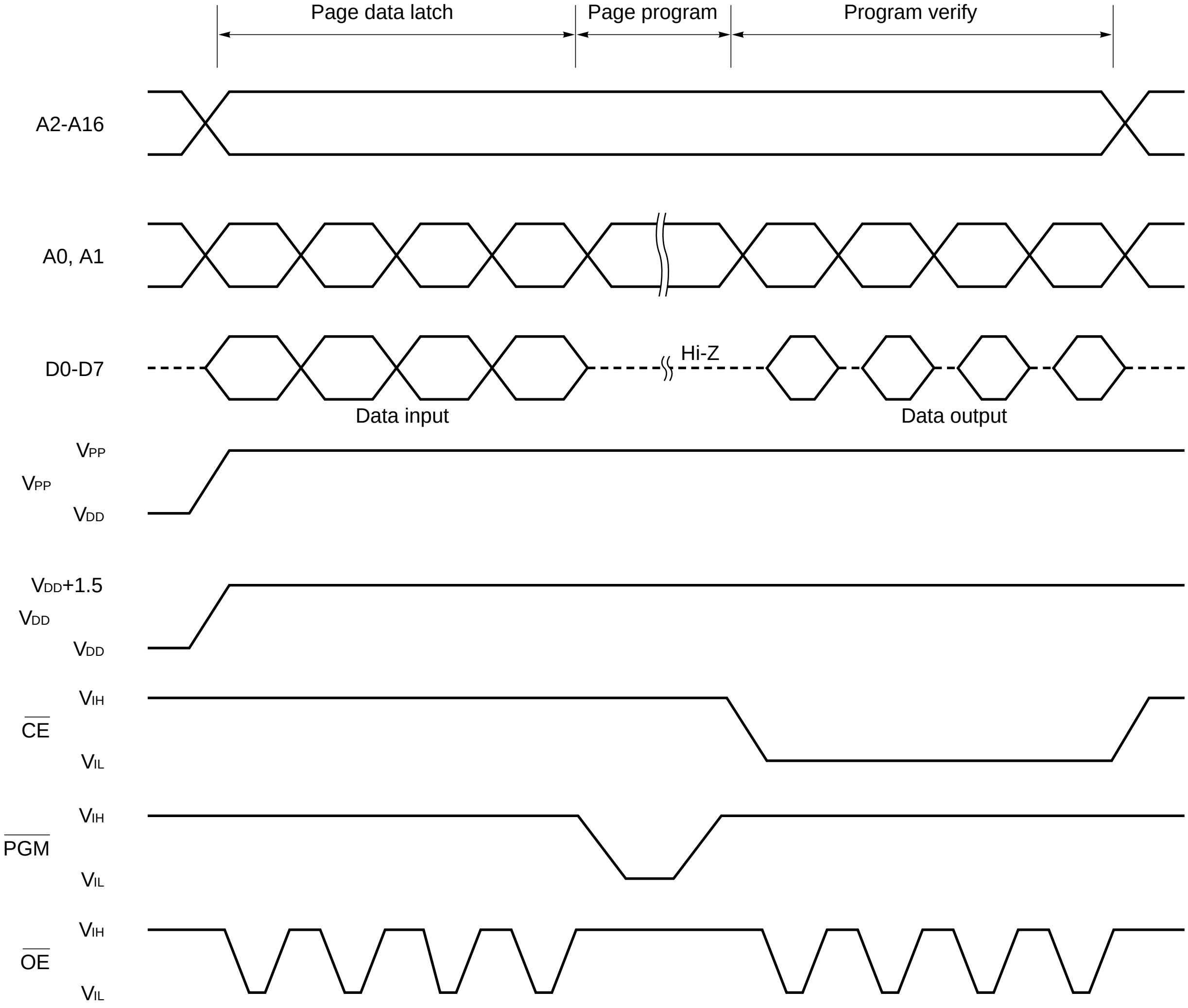
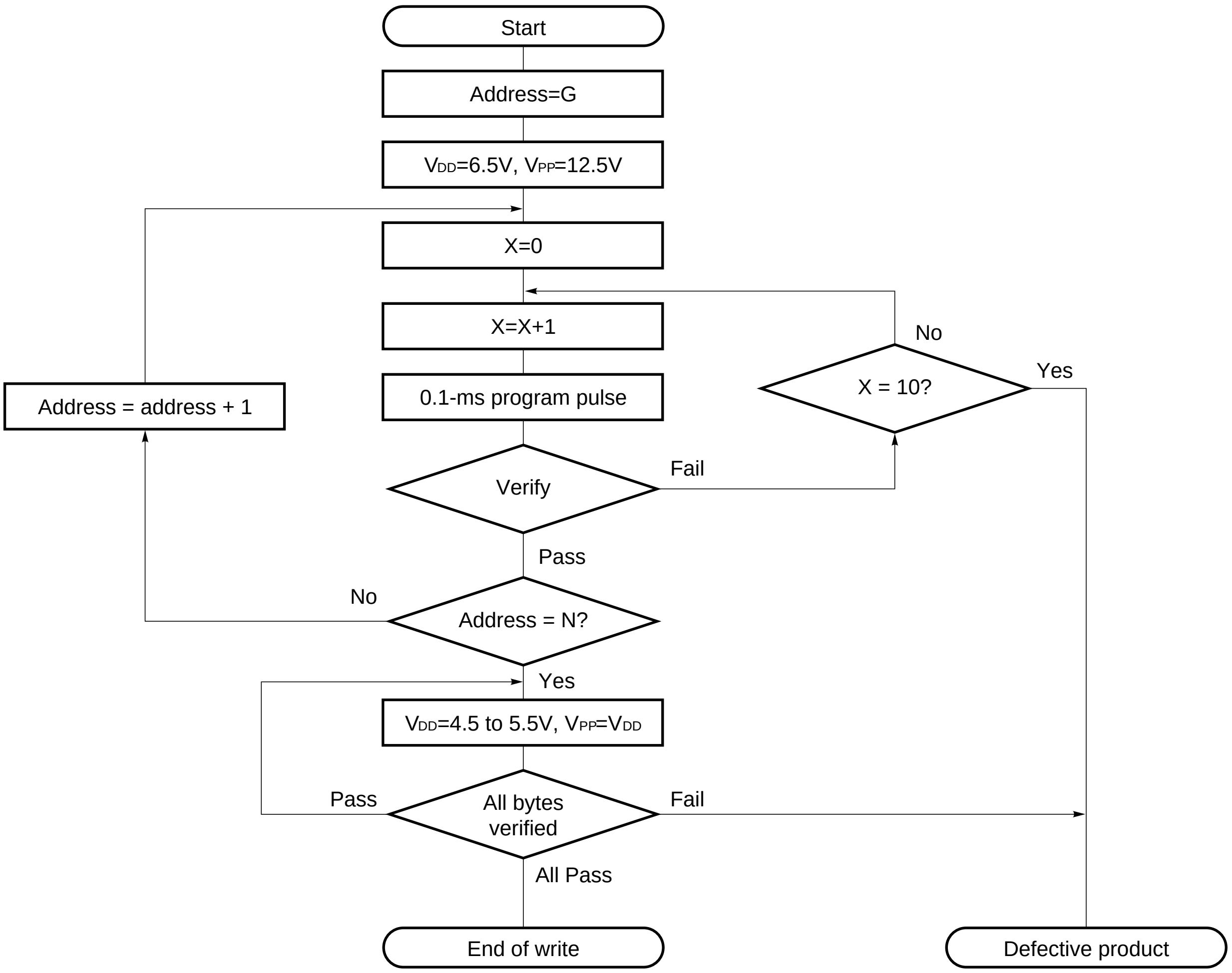


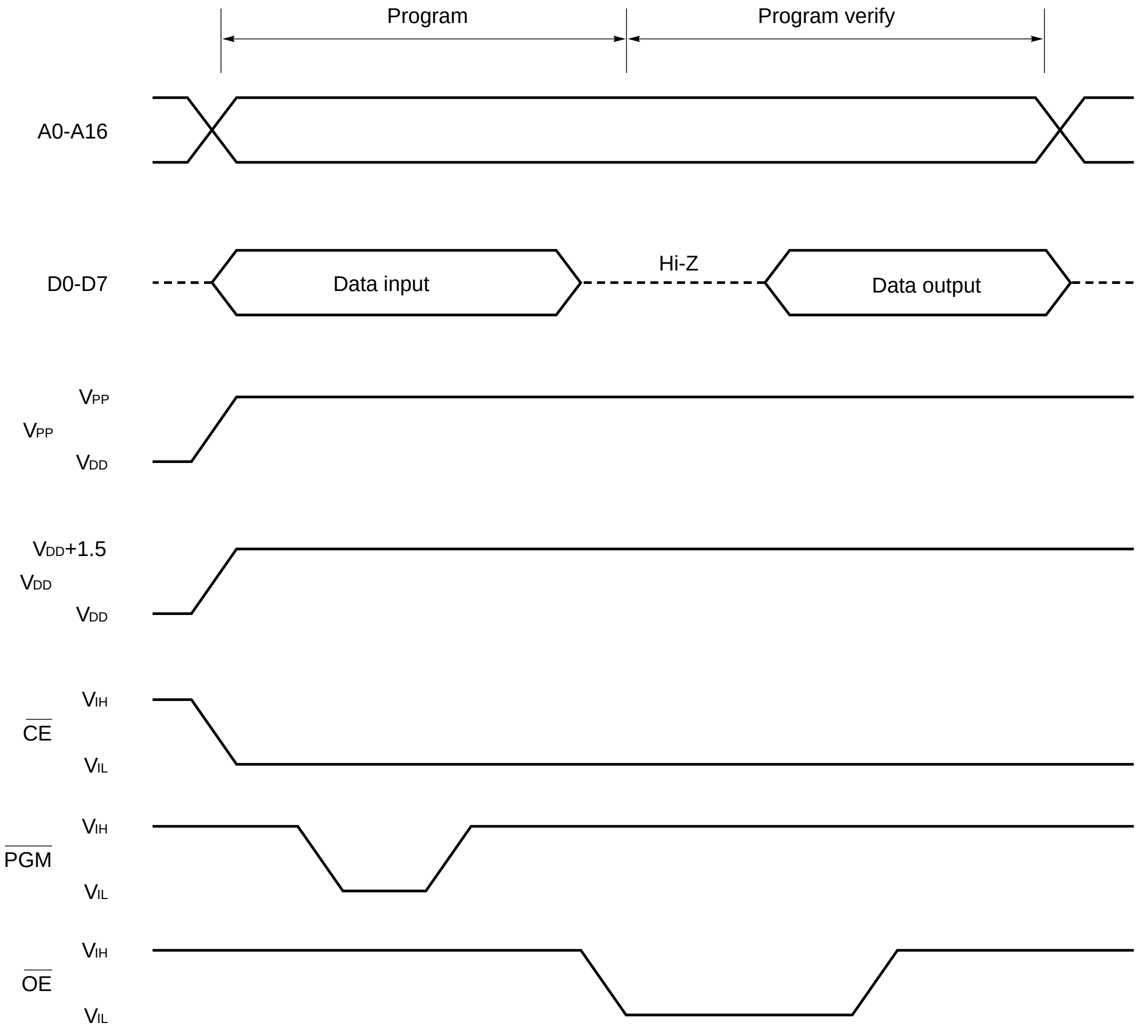
Figure 23-5. Byte Program Mode Flowchart



G = start address

N = last address of program

Figure 23-6. 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 and 78P018FY may be degraded.

23.3.3 PROM read sequence

The contents of the PROM can be read to the external data bus (D0-D7) in the following sequence:

(1) Fix the $\overline{\text{RESET}}$ pin to the low level. Supply +5 V to the V_{PP} pin. Process the pins not used by referring to **1.5** or **2.5 Pin Configuration (2) PROM programming mode**.

(2) Supply +5 V to the V_{DD} and V_{PP} pins.

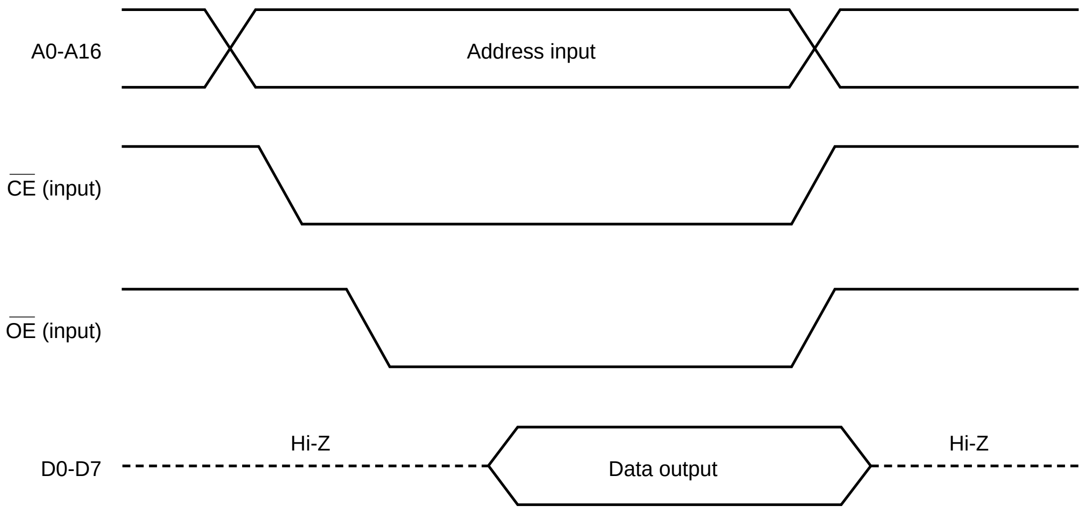
(3) Input the address of the data to be read to the A0-A16 pins.

(4) Read mode.

(5) Data is output to the D0-D7 pins.

Figure 23-7 shows the timing of steps (2) to (5) above.

Figure 23-7. PROM Read Timing


A0-A16
Address input
$\overline{\text{CE}}$ (input)
$\overline{\text{OE}}$ (input)
D0-D7
Hi-Z
Data output
Hi-Z

23.4 Erasure: μPD78P018FDW, 78P018FKK-S μPD78P018FYDW, 78P018FYKK-S

The data written to the program memory of the μPD78P018FDW, 78P018FKK-S, 78P018FYDW and 78P018FYKK-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 erase 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.

23.5 Erasure Window Sticker: μPD78P018FDW, 78P018FKK-S μPD78P018FYDW, 78P018FYKK-S

To protect from accidental erasure by light sources other than that of the EPROM eraser, and to protect internal circuit other than the EPROM from misoperation due to light, stick a protective sticker over the erasure window when EPROM erasure is not being performed.

23.6 Screening of One-Time PROM Model

Because of their structure, the one-time PROM models (μPD78P018FCW, 78P018FGC-AB8, 78P018FGK-8A8, 78P018FYCW, 78P018FYGC-AB8, and 78P018FYGK-8A8) 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^{TM} microcontroller, for writing, marking, screening, and verifying one-time PROMs. For details, consult NEC.

[MEMO]

CHAPTER 24 INSTRUCTION SET

This chapter lists the instruction set of the μPD78018F and 78018FY 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)**.

24.1 Legend

24.1.1 Operand representation and formats

In the operand field of each instruction, an operand is written according to the format for operand representation of that instruction (for details, refer to the assembler specifications). Some operands may be written in two or more formats. In this case, select one of them. Uppercase characters, #, !, $, and [ ] are keywords and must be written as is. The meanings of the symbols are as follows:

- # : immediate data
- ! : absolute address
- $: relative address
- [ ] : indirect address

To write immediate data, also use an appropriate numeric value or label. To write a label, be sure to use #, !, $, or [ ].

Register formats r or rp for an operand can be written 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 24-1. Operand Representation and Formats

Representation	Format
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7), AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol[Note] Special function register symbol (only even address of register that can be manipulated in 16-bit units)[Note]
saddr saddrp	FE20H-FF1FH immediate data or label FE20H-FF1FH immediate data or label (even address only)
addr16 addr11 addr5	0000H-FFFFH immediate data or label (even address only for 16-bit data transfer instruction) 0800H-0FFFH immediate data or label 0040H-007FH immediate data or label (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0-RB3

Note FFD0H-FFDFH cannot be addressed.

Remark For the symbols of the special function registers, refer to **Table 5-7. Special Function Register List**.

24.1.2 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 ( )
$\times_H$, $\times_L$	:	high-order 8 bits and low-order 8 bits of 16-bit register
$\wedge$	:	logical product (AND)
$\vee$	:	logical sum (OR)
$\veebar$	:	exclusive logical sum (exclusive OR)
——	:	inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	signed 8-bit data (displacement value)

24.1.3 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

24.2 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 ← CY, $A_{m-1} \leftarrow A_m$) × 1 time			×
	ROLC	A, 1	1	4	–	(CY ← A_7, A_0 ← 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.

24.3 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

2nd Operand 1st 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

2nd Operand 1st 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

2nd Operand 1st 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

2nd Operand 1st 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

[MEMO]

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)

Part Number Item	µPD78014 Subseries	µPD78014H Subseries	µPD78018F Subseries
EMI noise measure	None	Provided	None
I^{2}C bus model (Y subseries)	Provided	None	Provided
PROM model	µPD78P014	µPD78P018F	
Supply voltage	V_{DD} = 2.7 to 6.0 V	V_{DD} = 1.8 to 5.5 V	
Internal high-speed RAM size	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 extension RAM size	None		µPD78011F: None µPD78012F: None µPD78013F: None µPD78014F: None µPD78015F: 512 bytes µPD78016F: 512 bytes µPD78018F: 1024 bytes µPD78P018F: 1024 bytes
Operation mode of serial interface (Y subseries)	3-wire/2-wire/SBI/I^{2}C: 1 ch 3-wire (with automatic transmission/reception): 1ch	—	3-wire/2-wire/I^{2}C: 1 ch 3-wire (with automatic transmission/reception): 1 ch
Bit 5 (SIC) of interrupt timing specification register (SINT) in SBI mode (selection of INTCSI0 interrupt source)	When SIC = 1: sets CSIIF0 (interrupt request flag) on detection of bus release	When SIC = 1: Sets CSIIF0 (interrupt request flag) on detection of bus release and at end of transfer	
Bit 5 (SIC) of interrupt timing specification register (SINT) in I^{2}C bus mode (selection of INTCSI0 interrupt source)	When SIC = 1: Sets CSIIF0 (interrupt request flag) on detection of stop condition	—	When SIC = 1: Sets CSIIF0 (interrupt request flag) on detection of stop condition and at end of transfer

Table A-1. Major Differences Between μPD78014, 78014H, and 78018F Subseries (2/2)

Part Number Item	μPD78014 Subseries	μPD78014H Subseries	μPD78018F Subseries
Function of bit 7 (BSYE) of serial bus interface control register (SBIC) (Y subseries)	Control of synchronous bus signal output • When BSYE = 0 Disables output of busy signal in synchronization with falling edge of clock of $\overline{\text{SCK0}}$ immediately after instruction that clears this bit to 0 in SBI mode. Make sure that BSYE = 0 in I^{2}C bus mode. • When BSYE = 1 Outputs busy signal from falling edge of $\overline{\text{SCK0}}$ following acknowledge signal in SBI mode.	—	Control of N-ch open-drain output for transmission in I^{2}C bus mode • When BSYE = 0 Enables output (transmission) • When BSYE = 1 Disables output (reception)
Automatic data transmit/ receive interval specification register (ADTI)	None	Provided	
Package	• 64-pin plastic shrink DIP (750 mil) • 64-pin ceramic shrink DIP (w/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 (w/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 or IE-78014-R-EM-A	IE-78014-R-EM-A	
Access timing to external memory	Differs between μPD78014 subseries and other subseries. Refer to individual data sheet		
Electrical characteristics, recommended soldering conditions	Refer to individual data sheet.		

Note PROM model only

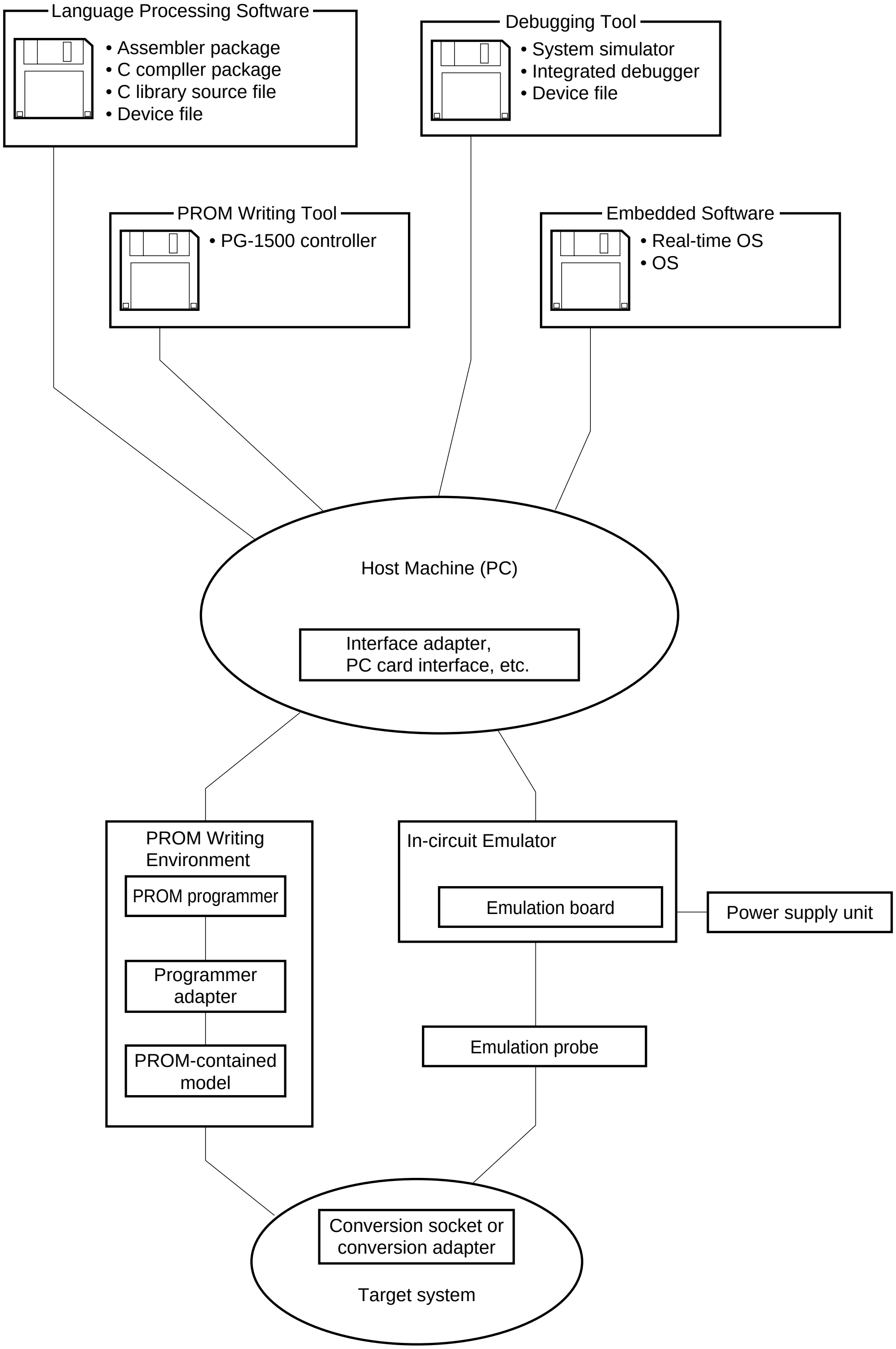
★ APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD78018F and 78018FY subseries.

Figure B-1 shows development tools.

Figure B-1. Development Tool Configuration (1/2)

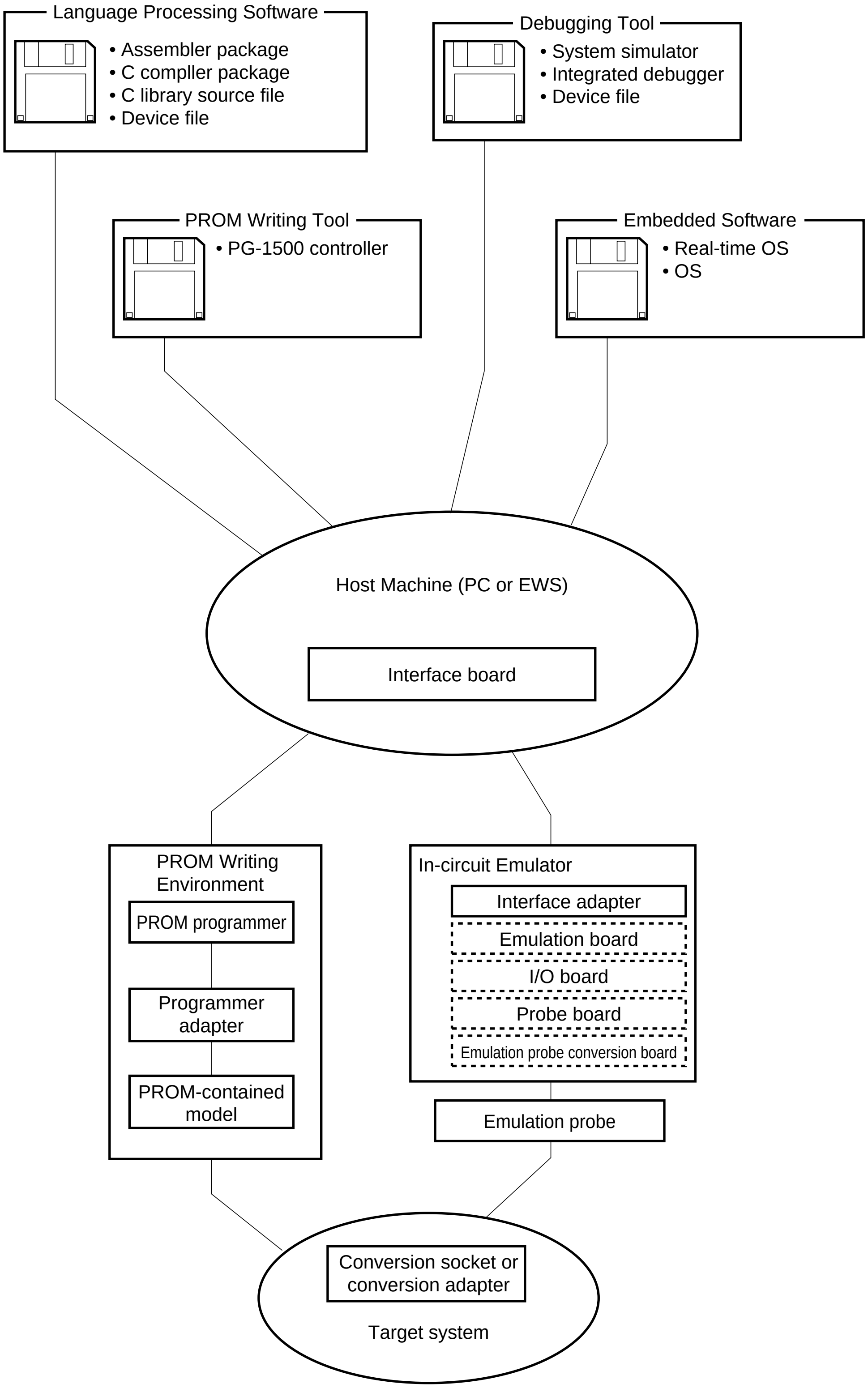
(1) When using the in-circuit emulator IE-78K0-NS



Phase-out/Discontinued

Figure B-1. Development Tool Configuration (2/2)

(2) When using the in-circuit emulator IE-78001-R-A



Remark Items in broken line boxes differ according to the development environment. Refer to **B.3.1. Hardware**.

B.1 Language Processing Software

RA78K/0 Assembler Package	This assembler converts programs written in mnemonics into an object codes executable with a microcomputer. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with an optical device file (DF78014). <Precaution when using RA78K/0 in PC environment> This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part Number: μS××××RA78K0
CC78K/0 C Compiler Package	This compiler converts programs written in C language into object codes executable with a microcomputer. This compiler should be used in combination with an optical assembler package (RA78K/0) and device file (DF78014). <Precaution when using RA78K/0 in PC environment> This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part Number: μS××××CC78K0
DF78014[Note] Device File	This file contains information peculiar to the device. This device file should be used in combination with an optical tool (RA78K/0, CC78K/0, SM78K0, ID78K0-NS, and ID78K0). Corresponding OS and host machine differ depending on the tool to be used with.
	Part Number: μS××××DF78014
CC78K/0-L C Library Source File	This is a source file of functions configuring the object library included in the C compiler package (CC78K/0). This file is required to match the object library included in C compiler package to the customer's specifications.
	Part Number: μS××××CC78K0-L

Note The DF78014 can be used in common with the RA78K/0, CC78K/0, SM78K0, ID78K0-NS, and ID78K0.

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××DF78014
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows (Japanese version) Note 1, 2	3.5-inch 2HD FD
AB13	IBM PC/AT^{TM} and its compatibles	Windows (Japanese version) Note 1, 2	3.5-inch 2HC FD
BB13		Windows (English version) Note 1, 2	
3P16	HP9000 Series 700^{TM}	HP-UX^{TM} (Rel. 9.05)	DAT (DDS)
3K13	SPARCstation^{TM}	SunOS^{TM} (Rel. 4.1.4)	3.5-inch 2HC FD
3K15			1/4-inch CGMT
3R13	NEWS^{TM} (RISC)	NEWS-OS^{TM} (Rel. 6.1)	3.5-inch 2HC FD

Notes 1. Can be operated in DOS environment.

2. Not support WindowsNT^{TM}

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 from 256K-bit to 4M-bit models.
PA-78P018CW PA-78P018GC PA-78P018GK PA-78P018KK-S PROM Programmer Adapter	PROM programmer adapter for µPD78P018F and 78P018FY and connected to PG-1500. PA-78P018CW : for 64-pin plastic shrink DIP (750 mil) PA-78P018GC : for 64-pin plastic QFP (GC-AB8 type) PA-78P018GK : for 64-pin plastic LQFP (GK-8A8 type) PA-78P018KK-S : for 64-pin ceramic WQFN (KK-S type)

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 controller is a DOS-based application. It should be used in the DOS Prompt when using in 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 Medium
5A13	PC-9800 series	MS-DOS (Ver.3.30 to Ver. 6.2**Note**)	3.5-inch 2HD FD
7B13	IBM PC/AT and its compatibles	Refer to **B.4**.	3.5-inch 2HC FD

Note MS-DOS ver.5.0 or above 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 this 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 required 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/AT and its compatible computers as the IE-78K0-NS host machine.
IE-78018-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-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 **Figure B-2**)	This conversion socket connects the NP-64GC to the target system board designed to mount a 64-pin plastic QFP (GC-AB8 type). Instead of connecting the NP-64GC, the μPD78P018FKK-S or 78P018FYKK-S (ceramic WQFN) can also be connected.
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 LQFP (GK-8A8 type).
	TGK-064SBW Conversion Adapter (Refer to **Figure B-4**)	This conversion socket connects the NP-64GK to the target system board designed to mount a 64-pin plastic LQFP (GK-8A8 type).
EV-9900		Jig used to remove the μPD78P018FKK-S or 78P018FYKK-S from the EV9200GC-64.

Note Under development

Remarks **1.** NP-64GC and NP-64GK are products of Naito Densei Machida Mfg. Co., Ltd.
Naito Densei Machida Mfg. Co., Ltd (TEL (044) 822-3813).

2. TGK-064SBW is a product of TOKYO ELETECH Corporation.
Reference: Daimaru Kogyo Ltd. Electronics Dept. (TEL: Tokyo 03-3820-7112)
Electronics 2nd Dept. (TEL: Osaka 06-244-6672)

3. EV-9200GC-64 is sold in five units.

4. TGK-064SBW is sold in one units.

B.3.1 Hardware (2/2)

(2) When using the in-circuit emulator IE-78001-R-A

IE-78001-R-A[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). This emulator should be used in combination with emulation probe and interface adapter, which is required to connect this emulator to the host machine.
IE-70000-98-IF-B or IE-7000-98-IF-C[Note] Interface Adapter		This adapter is required when using the PC-9800 Series computer (except notebook type) as the IE-78001-R-A host machine.
IE-70000-PC-IF-B or IE-7000-PC-IF-C[Note] Interface Adapter		This adapter is required when using the IBM PC/AT and its compatible computers as the IE-78001-R-A host machine.
IE-78000-R-SV3 Interface Adapter		This is adapter and cable required when using an EWS computer as the IE-78001-R-A host machine, and is used connected to the board in the IE-78000-R-A. As Ethernet^{TM}, 10Base-5 is supported. If the other methods are used, a commercially available conversion adapter is necessary.
IE-78018-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 and emulation conversion board.
	IE-78K0-R-EX1[Note] Emulation Probe Conversion Board	This board is required when using the IE-78018-NS-EM1 on the IE-78001-R-A.
IE-78014R-EM-A Emulation Board		This board emulates the operations of the peripheral hardware peculiar to a device (3.0 to 6.0 V supported). It should be used in combination with the IE-78001-R-A.
EP-78240GC-R 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 **Figure B-2**)	This conversion socket connects the EP-78240GC-R to the target system board designed to mount a 64-pin plastic QFP (GC-AB8 type). Instead of connecting the EP-78240GC-R, the μPD78P018FKK-S or 78P018FYKK-S (ceramic WQFN) can also be connected.
EP-78012GK-R Emulation Probe		This probe is used to connect the in-circuit emulator to the target system and is designed for 64-pin plastic LQFP (GK-8A8 type).
	TGK-064SBW Conversion Adapter (Refer to **Figure B-4**)	This conversion socket connects the EP-78012GK-R to the target system board designed to mount a 64-pin plastic LQFP (GK-8A8 type).
EV-9900		Jig used to remove the μPD78P018FKK-S or 78P018FYKK-S from the EV9200GC-64.

Note Under development

Remarks 1. TGK-064SBW is a product of TOKYO ELETECH Corporation.
Reference: Daimaru Kogyo Ltd. Electronics Dept. (TEL: Tokyo 03-3820-7112)
Electronics 2nd Dept. (TEL: Osaka 06-244-6672)

2. EV-9200GC-64 is sold in five units.

3. TGK-064SBW is sold in one units.

B.3.2 Software (1/2)

SM78K0 System Simulator	This system simulator is used to perform debugging at C source level or assembler level while simulating the operation of the target system on a host machine. This simulator runs on Windows. Use of the SM78K0 allows the execution of application logical testing and performance testing on an independent basis from hardware development without having to use an in-circuit emulator, thereby providing higher development efficiency and software quality. The SM78K0 should be used in combination with the optical 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 Medium
AA13	PC-9800 Series	Windows (Japanese version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[Note]	

Note Not support WindowsNT

B.3.2 Software (2/2)

ID78K0-NS[Note] Integrated Debugger (supporting in-circuit emulator IE-78K0-NS)	This debugger is a control program to debug 78K/0 Series microcontrollers. It adopts a graphical user interface, which is equivalent visually and operationally to Windows or OSF/Motif™. It also has an enhanced debugging function for C language programs, and thus trace results can be displayed on screen in C-language level by using the windows integration function which links a trace result with its source program, disassembled display, and memory display. In addition, by incorporating function modules such as task debugger and system performance analyzer, the efficiency of debugging programs, which run on real-time OSs can be improved. It should be used in combination with the optical device file (DF78014).
ID78K0 Integrated Debugger (supporting 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 Medium
AA13	PC-9800 Series	Windows (Japanese version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[Note]	

Note Not support WindowsNT

μS××××ID78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows (Japanese version)[Note]	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version)[Note]	3.5-inch 2HC FD
BB13		Windows (English version)[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 Not support WindowsNT

B.4 OS for IBM PC

The following OSs for the IBM PC are supported.

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 and above have a task swap function, this function cannot be used with this software.

B.5 System Upgrade from Former In-circuit Emulator for 78K/0 Series to IE-78001-R-A

If you already have a former in-circuit emulator for 78K/0 Series microcontrollers (IE-78000-R or IE-78000-R-A), that in-circuit emulator can operate as an equivalent to the IE-78001-R-A by replacing its internal break board with the IE-78001-R-BK (under development).

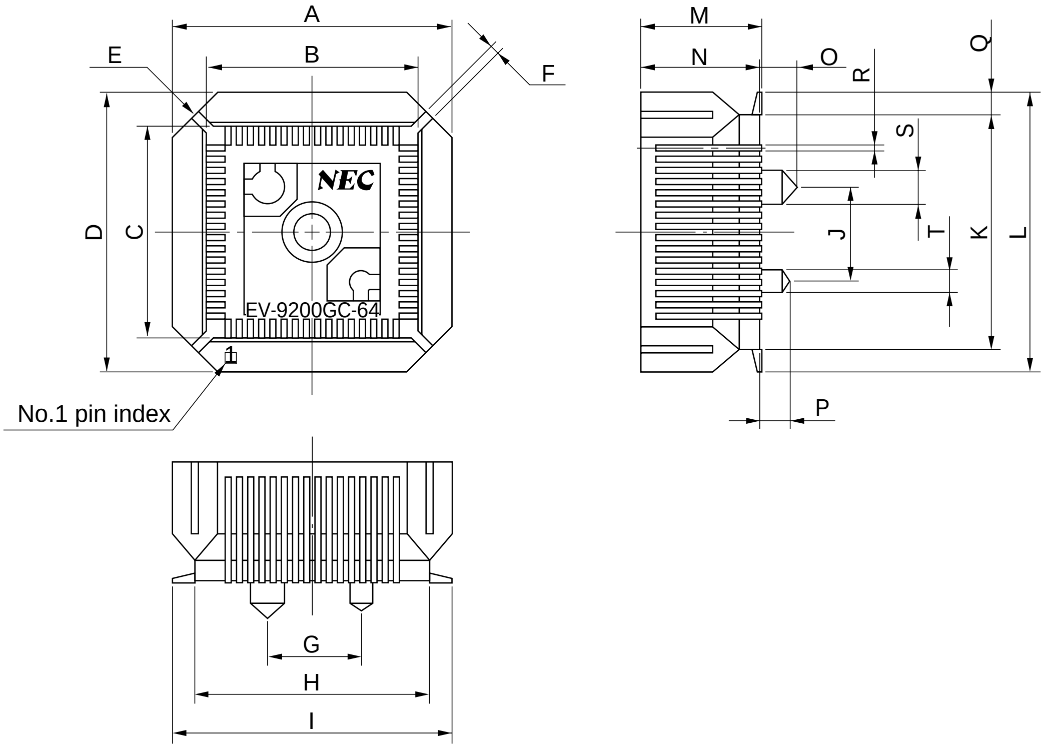
Table B-1. System-up Method from Former In-circuit Emulator for 78K/0 Series to the IE-78001-R-A

In-circuit Emulator Owned	In-circuit Emulator Cabinet System-up[Note]	Board to be Purchased
IE-78000-R	Required	IE-78001-R-BK
IE-78000-R-A	Not required	

Note For system-up of a cabinet, send your in-circuit emulator to NEC.

Dimensions of Conversion Socket (EV-9200GC-64) and Recommended Board Mounting Pattern

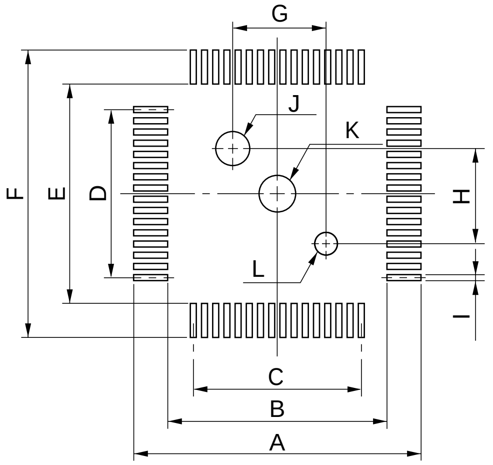
Figure B-2. EV-9200GC-64 Dimensions (Reference)



EV-9200GC-64-G0E

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 Pattern of Board Mounting (Reference)



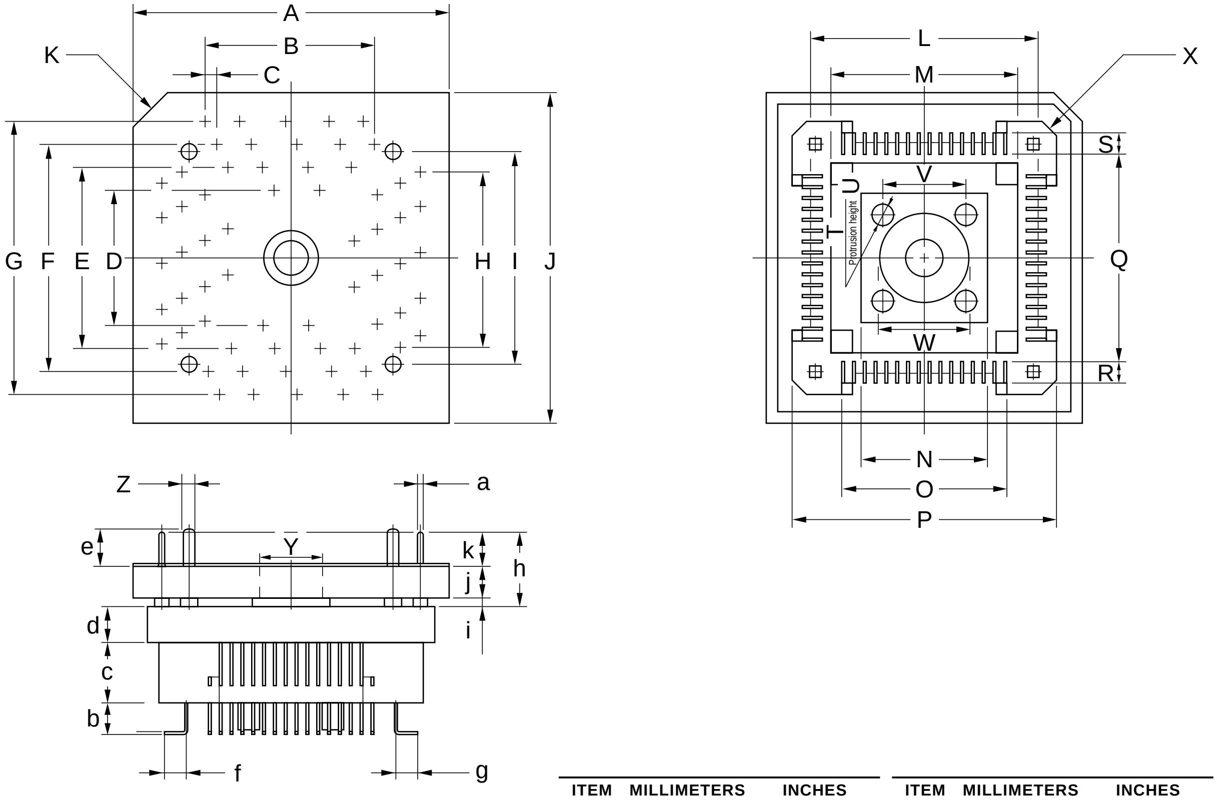
EV-9200GC-64-P1E

ITEM	MILLIMETERS	INCHES
A	19.5	0.768
B	14.8	0.583
C	0.8±0.02 × 15=12.0±0.05	$0.031^{+0.002}_{-0.001} \times 0.591 = 0.472^{+0.003}_{-0.002}$
D	0.8±0.02 × 15=12.0±0.05	$0.031^{+0.002}_{-0.001} \times 0.591 = 0.472^{+0.003}_{-0.002}$
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	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	ϕ2.2±0.1	$\phi 0.087^{+0.004}_{-0.005}$
L	ϕ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

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 Adapter (TGK-064SBW)

Figure B-4. TGK-064SBW Dimensions (Reference)



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-G0E

note: Product by TOKYO ELETECH CORPORATION.

★ APPENDIX C EMBEDDED SOFTWARE

For efficient program development and maintenance of μPD78018F and 78018FY subseries, the following embedded softwares are available.

Real-Time OS (1/2)

RX78K/0 Real-time OS	RX 78K/0 is a real-time OS conforming to the μITRON specifications. Tool (configurator) for generating nucleus of RX78K/0 and plural information tables is supplied. Used in combination with an optical assembler package (RA78/0) and device file (DF78014). <Precaution when using RX78K/0 in PC environment> The real-time OS is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part Number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K/0, fill in the purchase application form in advance and sign the User Agreement.

Remark ×××× and ΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Source program for mass-produced object

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows (Japanese version)[Note 1, 2]	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version)[Note 1, 2]	3.5-inch 2HC FD
BB10		Windows (English version)[Note 1, 2]	
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 also be operated in DOS environment.

2. Not support WindowsNT.

Real-Time OS (2/2)

MX78K0 OS	MX78K/0 is an OS for μITRON specification subsets. A nucleus for the MX78K/0 is also included as a companion product. This manages tasks, events, and time. In the task management, determining the task execution order and switching from task to the next task are performed. <Precaution when using MX78K/0 in PC environment> The MX78K/0 is a DOS-based application. It should be used in the DOS Prompt when using in Windows.
	Part Number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Use in preproduction stages.
××	Mass-production object	Use in mass production stages.
S01	Source program	Only the users who purchased mass-production objects are allowed to purchase this program.

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 Series	Windows (Japanese version)[Note 1, 2]	3.5-inch 2HD FD
AB13	IBM PC/AT and its compatibles	Windows (Japanese version)[Note 1, 2]	3.5-inch 2HC FD
BB10		Windows (English version)[Note 1, 2]	
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 also be operated in DOS environment.

2. Not support WindowsNT.

Phase-out/Discontinued

[MEMO]

APPENDIX D   REGISTER INDEX

D.1 Register Index (In Alphabetical Order with Respect to Register Name)

D.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

APPENDIX E REVISION HISTORY

The following table shows the revision history of this manual. “Chapter” indicates the chapter of the previous edition.

(1/3)

Edition	Major Revisions	Chapter
2nd edition	The following products have been developed: • μPD78013F, 78014F, 78015F, 78016F	Throughout
	Addition of the following products: • μPD78011F, 78012F • μPD78011FY, 78012FY, 78013FY, 78014FY, 78015FY, 78016FY, 78P018FY	
	Change in supply voltage • V_{DD} = 2.0 to 6.0 V → V_{DD} = 1.8 to 5.5 V	
	Addition of CHAPTER 2 GENERAL (μPD78018FY SUBSERIES)	CHAPTER 2 GENERAL (μPD78018FY SUBSERIES)
	Correction to recommended connections when unused	CHAPTER 3 PIN FUNCTIONS (μPD78018F SUBSERIES)
	Addition of CHAPTER 4 PIN FUNCTIONS (μPD78018FY SUBSERIES)	CHAPTER 4 PIN FUNCTIONS (μPD78018FY SUBSERIES)
	Addition of 6.2.4 Port 2 (μPD78018FY subseries)	CHAPTER 6 PORT FUNCTIONS
	Change of setting bit 0 of A/D converter mode register (ADM) from 1 → HSC	CHAPTER 14 A/D CONVERTER
	Addition of notes on processing of AV_{DD} pin	
	Addition of CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD78018FY SUBSERIES)	CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD78018FY SUBSERIES)
	Addition of automatic data transmission/reception time interval specification register (ADTI)	CHAPTER 17 SERIAL INTERFACE CHANNEL 1
	Addition of CHAPTER 22 ROM CORRECTION	CHAPTER 22 ROM CORRECTION
	Addition of HP9000 series 700 as host machine for language processing software	APPENDIX A DEVELOPMENT TOOLS
	Addition of system simulator (SM78K0)	
	Addition of A.4 OS for IBM PC	
	Addition of HP9000 series 700 as host machine for real-time OS (RX78K/0)	APPENDIX B EMBEDDED SOFTWARE
	Addition of OS (MX78K0)	
3rd edition	Addition of following applicable models: • μPD78018F and 78018FY	Throughout
	Change of following products from “under development” to “developed”: • μPD78P018F, 78P018FY (except EPROM models)	
	Change of quality grade of following products from “standard” to “not applicable” • μPD78P018FDW, 78P018FKK-S, 78P018FYDW, 78P018FYKK-S	

(2/3)

Edition	Revision from Previous Edition	Page
3rd edition	Change of format of watchdog timer mode register (WDTM) and addition of Notes	CHAPTER 11 WATCHDOG TIMER
	Addition of Notes on serial I/O shift register 0 (SIO0) of μPD78018FY subseries	CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD78018FY SUBSERIES)
	Correction of Figure 16-21. Example of Communication from Master to Slave (with 9-clock wait selected for both master and slave)	
	Correction of Figure 16-22. Example of Communication from slave to Master (with 9-clock wait selected for both master and slave)	
	Correction of 16.4.6 Restrictions using I^{2}C bus mode	
	Addition of Notes to Figure 17-5. Format of Automatic Data Transmission/Reception Interval Specification Register	CHAPTER 17 SERIAL INTERFACE CHANNEL 1
	Addition of Notes to 17.4.3 (3) (d) Busy control option	
	Addition of following products: IE-78000-R-A, IE-70000-98-IF-B, IE-70000-98N-IF, IF-70000-PC-IF-B, IE-78000-R-SV3, ID78K0	APPENDIX A DEVELOPMENT TOOLS
	Addition of description on how to upgrade other in-circuit emulators to IE-78000-R-A	
	Version of supported OS upgraded	APPENDIX A DEVELOPMENT TOOLS, APPENDIX B EMBEDDED SOFTWARE
4th edition	Addition of following products as applicable models: μPD78011F(A), 78012F(A), 78013F(A), 78014F(A), 78015F(A), 78016F(A), 78018F(A), 78P018F(A), 78012F(A2) μPD78011FYGK, 78012FYGK, 78014FYGK, 78P018FYGK	Throughout
	Following products developed: μPD78P018FDW, 78P018FKK-S, 78P018FYDW, 78P018FYKK-S	
	Addition of notes on changing operation mode of serial interface channel 0	CHAPTER 15 SERIAL INTERFACE CHANNEL 0 (μPD78018F SUBSERIES)
		CHAPTER 16 SERIAL INTERFACE CHANNEL 0 (μPD78018FY SUBSERIES)
	Addition of notes to Table 23-1 Differences between μPD78P018F, 78P018FY, and Mask ROM Model	CHAPTER 23 μPD780P018F, 78P018FY
	Addition of APPENDIX A DIFFERENCES BETWEEN μPD78014, 78014H, AND 78018F SUBSERIES	APPENDIX A DIFFERENCES BETWEEN μPD78014, 78014H, AND 78018F SUBSERIES
	Deletion of 5-inch FD model from Windows-supporting development tools	APPENDIX B DEVELOPMENT TOOLS
	Following products developed: • IE-78000-R-A • ID78K0	
	Change of conversion adapter name from EV-9500GK-64 to TGK-064SBW	

(3/3)

Edition	Revision from Previous Edition	Page
5th edition	Change of **Figures 6-6**, **6-8 Block Diagram of P20, P21, and P23-P26**, **Figures 6-7, 6-9 Block Diagram of P22 and P27**	CHAPTER 6 PORT FUNCTIONS
	Addition of **Figures 9-10, 9-13 Square Wave Output Operation**	CHAPTER 9 8-BIT TIMER/EVENT COUNTER
	Addition of Caution to **14.2 (6) ANI0-ANI7 pins**	CHAPTER 14 A/D CONVERTER
	Addition of Caution to **14.2 (7) AV$_{REF}$ pin**	
	Change of Note on BSYE flag in **Figure 15-4 Format of Serial Bus Interface Control Register**	CHAPTER 15 SERIAL INTERFACE CHANNEL 0
	Addition of Caution to **15.4.3 (1) (a) Bus release signal (REL)**, **(b) Command signal (CMD)**, and **(11) Notes on SBI mode**	
	APPENDIX B DEVELOPMENT TOOLS Throughout: Support for in-circuit emulator IE-78K0-NS	APPENDIX B DEVELOPMENT TOOLS
	APPENDIX C EMBEDDED SOFTWARE Throughout: Deletion of fuzzy inference development support system	APPENDIX C EMBEDDED SOFTWARE

Phase-out/Discontinued

[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	□	□	□	□