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.

User’s Manual

μPD78044F Subseries

8-bit Singlechip Microcontrollers

μPD78042F
μPD78043F
μPD78044F
μPD78045F
μPD78P048A

Document No. U10908EJ2V0UM00 (2nd edition)
Date Published July 1997 N

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed:	μPD78P048AKL-S
The customer must judge the need for license:	μPD78042FGF-xxx-3B9, μPD78043FGF-xxx-3B9, μPD78044FGF-xxx-3B9, μPD78045FGF-xxx-3B9, μPD78P048AGF-3B9

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.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

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.

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: 800-366-9782
Fax: 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.
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

J96. 8

Major Revisions in This Edition

Page	Description
Throughout	Change of all target devices from “under development” to “developed”
p.2	Addition of **1.4 Quality Grade**
p.89	Change of Caution in **5.4.1 Main system clock oscillator**
p.93	**5.5 Clock Generator Operations** Correction of description on current consumption in STOP mode
p.124	**6.6 16-Bit Timer/Event Counter Operating Precautions** Addition of **(6) OVF0 flag operation**
p.142, 146	Addition of figure of **Square Wave Output Operation Timings**
p.160	Change of **Figure 9-3 Watchdog Timer Mode Register Format** and addition of Note and Caution
p.190, 195	Addition of Caution on selecting serial interface channel 0 operating mode
p.244, 245, 254, 255	Addition of Caution on busy control when controlling interval time by using automatic data transmit/receive interval specify register (ADTI)
p.326	**8.2.2 STOP mode** Correction of Caution on setting STOP mode
p.333	**Table 20-1 Differences between μPD78P048A and Mask ROM Versions** Addition of Caution on switching from PROM version to mask ROM version
p.361	**Table A-1 Differences among μPD78044, 78044A, and 78044F Subseries** Change of display output current of FIP controller/driver of μPD78044F Subseries
p.363	Addition of following products in **APPENDIX B DEVELOPMENT TOOLS** IE-78000-R-A, IE-70000-98-IF-B, IE-70000-98N-IF, IE-70000-PC-IF-B, IE-78000-R-SV3, ID78K0
p.383	Addition of **APPENDIX E REVISION HISTORY**

The mark ★ shows major revised points.

PREFACE

Readers This manual has been prepared for user engineers who want to understand the functions of the μPD78044F Subseries and design and develop its application systems and programs.

★ **Caution Among the μPD78044F Subseries products, the μPD78P048AKL-S does not have the reliability level required for mass production. Use this model for experiment or function evaluation only.**

Purpose This manual is intended for users to understand the functions described in the Organization below.

Organization The μPD78044F Subseries manual consists of two parts: this manual and Instructions (common to the 78K/0 Series)

μPD78044F Subseries User's Manual (This manual)	78K/0 Series User's Manual — Instructions —
• Pin functions • Internal block functions • Interrupts • Miscellaneous on-chip peripheral functions	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

Before reading this manual, you must have general knowledge of electric and logic circuits and microcontrollers.

- When you want to understand the functions in general:
  —> Read this manual in the order listed in **CONTENTS**.
- How to interpret the register format:
  —> For the circled bit number, the bit name is defined as a reserved word in RA78K/0, and in the CC78K/0, already defined in the header file named sfrbit.h.
- When confirming the details of a register whose register name is known:
  —> Refer to **APPENDIX D** "**REGISTER INDEX**."
- When you want to know differences with the μPD78044 and μPD78044A Subseries:
  —> Read **APPENDIX A** "**DIFFERENCES AMONG μPD78044, μPD78044A, AND μPD78044F SUBSERIES**."
- When you want to know the details of the μPD78044F Subseries instruction function:
  —> Refer to **78K/0 Series User's Manual: Instructions (U12326E)**.
- ★ When you want to know the electrical specifications of the μPD78044F Subseries:
  —> Refer to **μPD78042F, 78043F, 78044F, and 78045F Data Sheet (U10700E)** and **μPD78P048A Data Sheet (U10611E)**.
- ★ When you want to know the application examples of each function of the μPD78044F Subseries:
  —> Refer to **78K/0 Series Application Note: Basics (II) (U10121E)**.

Legend

Data representation weight : High digits on the left and low digits on the right
Active low representations : $\overline{xxx}$ (top bar over pin or signal name)
Note : Description of “Note” in the text.
Caution : Information requiring particular attention
Remark : Additional explanatory material
Numeral representations : Binary .... xxxx or xxxxB
Decimal xxxx
Hexadecimal .... xxxxH

RELATED DOCUMENTS

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

• Documents related to devices

Document name	Document number	
	Japanese version	English version
*μ*PD78042F, 78043F, 78044F, 78045F Data Sheet	U10700J	U10700E
*μ*PD78P048A Data Sheet	U10611J	U10611E
★ *μ*PD78044F Subseries User's Manual	U10908J	This manual
78K/0 Series Instruction Table	U10903J	—
78K/0 Series Instruction Set	U10904J	—
*μ*PD78044A, 78044F Subseries Special Function Register Table	U10701J	—
78K/0 Series User’s Manual: Instructions	U12326J	U12326E
★ 78K/0 Series Application Note: Basics (II)	U10121J	U10121E

• Documents related to development tools (User's Manual) (1/2)

Document name		Document number	
		Japanese version	English version
RA78K Series Assembler Package	Operation	EEU-809	EEU-1399
	Language	EEU-815	EEU-1404
RA78K Series Structured Assembler Preprocessor		EEU-817	EEU-1402
★ RA78K0 Assembler Package	★ Operation	U11802J	U11802E
	★ Assembly Language	U11801J	U11801E
	★ Structured Assembly Language	U11789J	U11789E
CC78K Series C Compiler	Operation	EEU-656	EEU-1280
	Language	EEU-655	EEU-1284
★ CC78K0 C Compiler	★ Operation	U11517J	U11517E
	★ Language	U11518J	U11518E
CC78K/0 C Compiler Application Note	Programming Know-How	EEA-618	EEA-1208

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

• Documents related to development tools (User's Manual) (2/2)

Document name		Document number	
		Japanese version	English version
CC78K Series Library Source File		U12322J	—
PG-1500 PROM Programmer		U11940J	EEU-1335
PG-1500 Controller— PC-9800 Series (MS-DOS™) Based		EEU-704	EEU-1291
PG-1500 Controller— IBM PC Series (PC DOS™) Based		EEU-5008	U10540E
IE-78000-R		U11376J	U11376E
★ IE-78000-R-A		U10057J	U10057E
IE-78000-R-BK		EEU-867	EEU-1427
IE-78044-R-EM		EEU-833	EEU-1424
EP-78130GF-R		EEU-943	EEU-1470
SM78K0 System Simulator Windows™ Based	Reference	U10181J	U10181E
SM78K Series System Simulator	External User Open Interface Specification	U10092J	U10092E
★ ID78K0 Integrated Debugger EWS Based	Reference	U11151J	—
★ ID78K0 Integrated Debugger PC Based	Reference	U11539J	U11539E
★ ID78K0 Integrated Debugger Windows Based	Guide	U11649J	U11649E
SD78K/0 Screen Debugger PC-9800 Series (MS-DOS) Based	Introduction	EEU-852	U10539E
	Reference	U10952J	—
SD78K/0 Screen Debugger IBM PC/AT™ (PC DOS) Based	Introduction	EEU-5024	EEU-1414
	Reference	U11279J	U11279E

• Documents related to embedded software (User's Manual)

Document name		Document number	
		Japanese version	English version
78K/0 Series Real-Time OS	Basics	U11537J	—
	Installation	U11536J	—
78K/0 Series OS MX78K0	Basics	U12257J	—
Fuzzy Knowledge Data Input Tools		EEU-829	EEU-1438
78K/0, 78K/II, 87AD Series Fuzzy Inference Development Support System (Translator)		EEU-862	EEU-1444
78K/0 Series Fuzzy Inference Development Support System (Fuzzy Inference Module)		EEU-858	EEU-1441
78K/0 Series Fuzzy Inference Development Support System (Fuzzy Inference Debugger)		EEU-921	EEU-1458

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

- **Other documents**

Document name	Document number	
	Japanese version	English version
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
Electrostatic Discharge (ESD) Test	MEM-539	—
Guide to Quality Assurance for Semiconductor Devices	C11893J	MEI-1202
Guide to Microcomputer-Related Products — Third Party Manufacturers	U11416J	—

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

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

Figure No. Title Page

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

[MEMO]

CHAPTER 1 OUTLINE

1.1 Features

- On-chip large-capacity ROM and RAM

Item / Part Number	Program Memory		Data memory			
	ROM	PROM	Internal high-speed RAM	Internal expansion RAM	Buffer RAM	FIP display RAM
μPD78042F	16 Kbytes	—	512 bytes	None	64 bytes	48 bytes
μPD78043F	24 Kbytes	—				
μPD78044F	32 Kbytes	—	1024 bytes			
μPD78045F	40 Kbytes	—				
μPD78P048A	—	60 Kbytes $^{\text{Note 1}}$	1024 bytes $^{\text{Note 2}}$	1024 bytes $^{\text{Note 3}}$		

Notes **1.** 16 K, 24 K, 32 K, 40 K or 60 Kbytes can be selected by the memory size switching register (IMS).

2. 512 or 1024 bytes can be selected by IMS.

3. 0 or 1024 bytes can be selected by the internal expansion RAM size switching register (IXS).

- Minimum instruction execution time changeable from high speed (0.4 μs: @ 5.0 MHz with main system clock) to ultra-low speed (122 μs: @ 32.768 kHz with subsystem clock)
- 68 I/O ports
- FIP™ controller/driver: 34 display outputs in total
  - Segments: 9 to 24
  - Digits: 2 to 16
- 8-bit resolution A/D converter: 8 channels
  - Power supply voltage (AV_{DD} = 4.0 to 5.5 V)
- Serial interface: 2 channels
  - 3-wire serial I/O/SBI/2-wire serial I/O mode: 1 channel
  - 3-wire serial I/O mode (Automatic transmit/receive function): 1 channel
- Timer: 6 channels
  - 16-bit timer/event counter: 1 channel
  - 8-bit timer/event counter: 2 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
  - 6-bit up/down counter: 1 channel
- 16 vectored interrupt sources
- One test input
- Two types of on-chip clock oscillators (for main system and subsystem clocks)
- Power supply voltage: V_{DD} = 2.7 to 5.5 V

1.2 Application Fields

CD players, cassette decks, tuners, integrated mini stereo systems, VCRs, microwave ovens, ECRs, etc.

1.3 Ordering Information

Part Number	Package	Internal ROM
μPD78042FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Mask ROM
μPD78043FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Mask ROM
μPD78044FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Mask ROM
μPD78045FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Mask ROM
μPD78P048AGF-3B9	80-pin plastic QFP (14 × 20 mm)	One-time PROM
μPD78P048AKL-S	80-pin ceramic WQFN	EPROM

Remark xxx is ROM code suffix.

★ 1.4 Quality Grade

Part Number	Package	Quality Grade
μPD78042FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Standard (for general electronic equipment)
μPD78043FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Standard (for general electronic equipment)
μPD78044FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Standard (for general electronic equipment)
μPD78045FGF-xxx-3B9	80-pin plastic QFP (14 × 20 mm)	Standard (for general electronic equipment)
μPD78P048AGF-3B9	80-pin plastic QFP (14 × 20 mm)	Standard (for general electronic equipment)
μPD78P048AKL-S	80-pin ceramic WQFN	Not applied (for function evaluation)

Caution The μPD78P048AKL-S does not have the reliability level required for mass production. Use this model for experiment or function evaluation only.

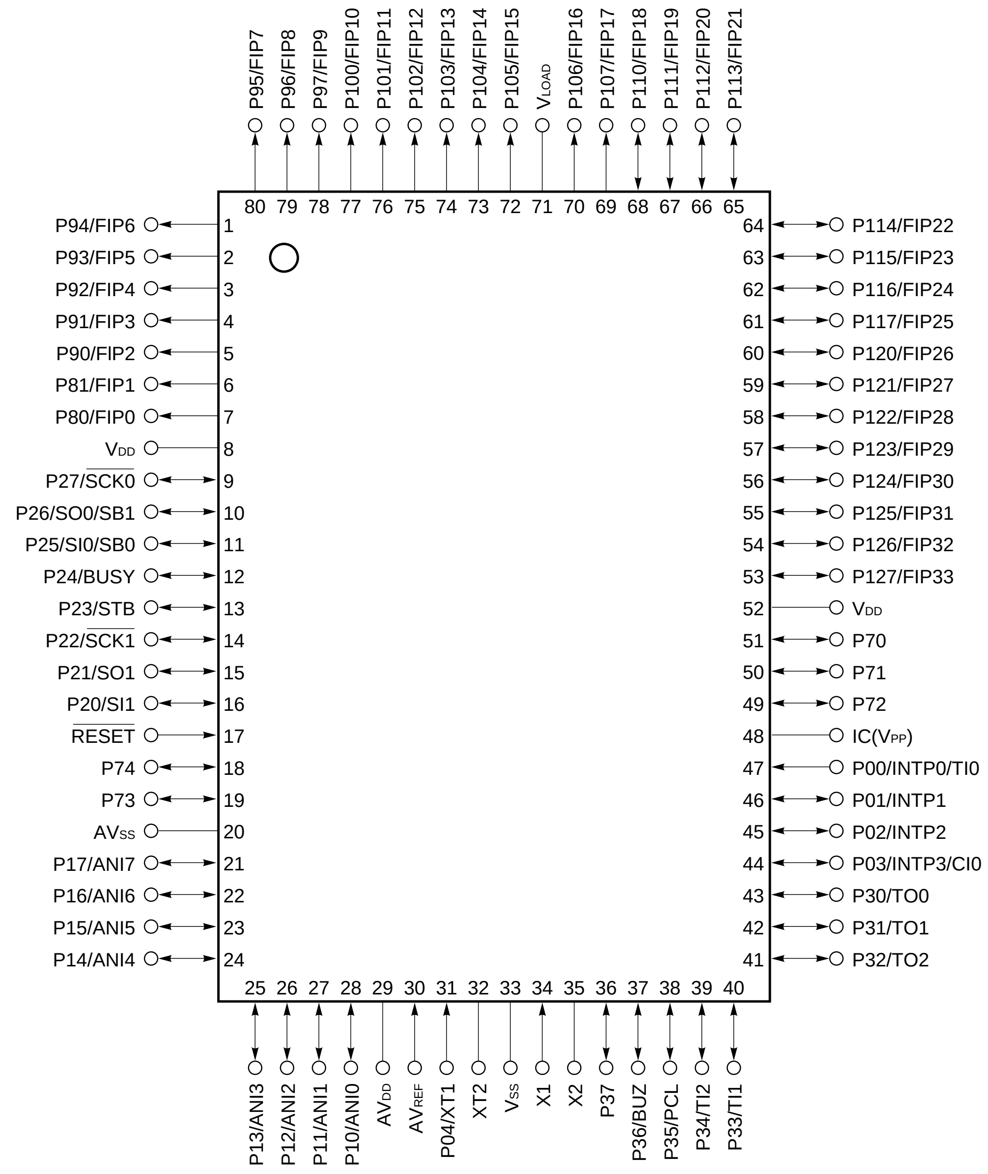
Remark xxx is ROM code suffix.

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 operating mode

- **80-pin plastic QFP (14 × 20 mm)**
  μPD78042FGF-xxx-3B9, μPD78043FGF-xxx-3B9,
  μPD78044FGF-xxx-3B9, μPD78045FGF-xxx-3B9,
  μPF78P048AGF-3B9,
- **80-pin ceramic WQFN**
  μPD78P048AKL-S



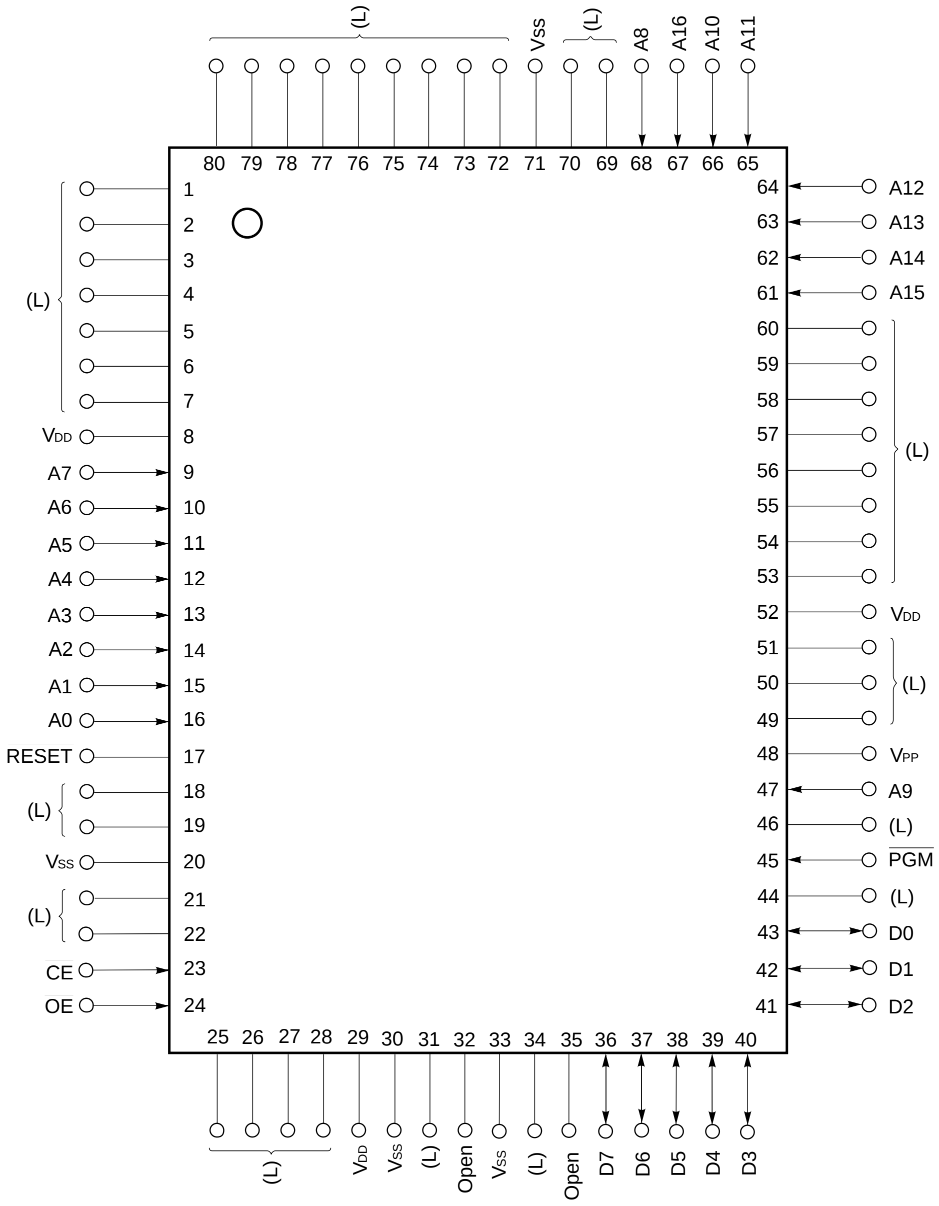
Cautions
1. **Be sure to connect IC (Internally Connected) pin to V_{SS} directly.**
2. **Connect AV_{DD} pin to V_{DD}.**
3. **Connect AV_{SS} pin to V_{SS}.**

Remark Pin connection in parentheses is intended for μPD78P048A.

ANI0 to ANI7	:	Analog Input
AV$_{\mathrm{DD}}$	:	Analog Power Supply
AV$_{\mathrm{REF}}$	:	Analog Reference Voltage
AV$_{\mathrm{SS}}$	:	Analog Ground
BUSY	:	Busy
BUZ	:	Buzzer Clock
CI0	:	Counter Input
FIP0 to FIP33	:	Fluorescent Indicator Panel
IC	:	Internally Connected
INTP0 to INTP3	:	Interrupt from Peripherals
P00 to P04	:	Port0
P10 to P17	:	Port1
P20 to P27	:	Port2
P30 to P37	:	Port3
P70 to P74	:	Port7
P80, P81	:	Port8
P90 to P97	:	Port9
P100 to P107	:	Port10
P110 to P117	:	Port11
P120 to P127	:	Port12
PCL	:	Programmable Clock
$\overline{\text{RESET}}$	:	Reset
SB0, SB1	:	Serial Bus
$\overline{\text{SCK0}}$, $\overline{\text{SCK1}}$	:	Serial Clock
SI0, SI1	:	Serial Input
SO0, SO1	:	Serial Output
STB	:	Strobe
TI0 to TI2	:	Timer Input
TO0 to TO2	:	Timer Output
V$_{\mathrm{DD}}$	:	Power Supply
V$_{\mathrm{LOAD}}$	:	Negative Power Supply
V$_{\mathrm{PP}}$	:	Programming Power Supply
V$_{\mathrm{SS}}$	:	Ground
X1, X2	:	Crystal (Main System Clock)
XT1, XT2	:	Crystal (Subsystem Clock)

(2) PROM programming mode

- **80-pin plastic QFP (14 × 20 mm)**
  μPD78P048AGF-3B9
- **80-pin ceramic WQFN**
  μPD78P048AKL-S



Cautions 1. (L) : Connect individually to V_{SS} via a pull-down resistor.

2. V_{SS} : Connect to the ground.

3. $\overline{\text{RESET}}$: Set to the low level.

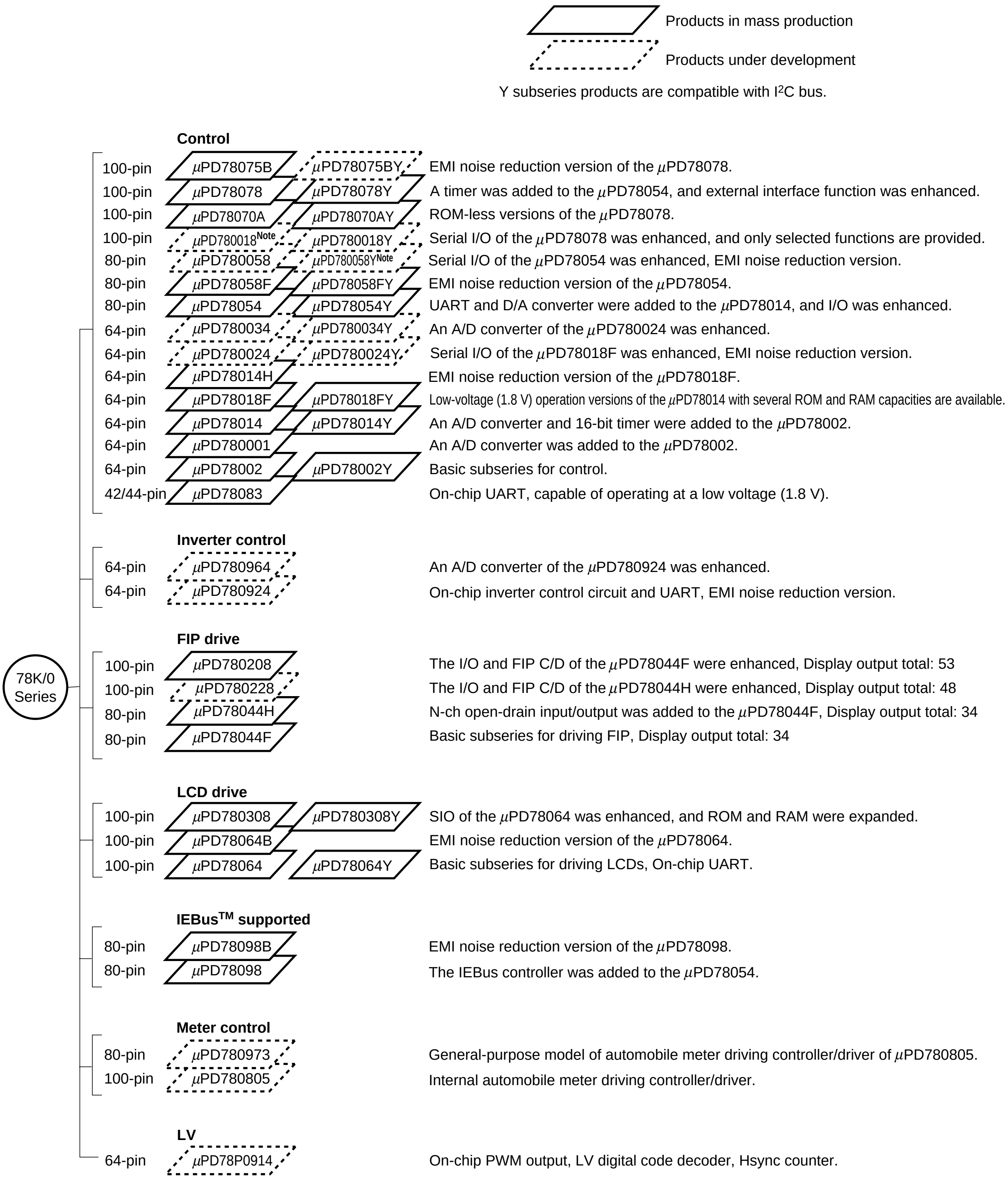
4. Open : Do not connect anything.

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

★

1.6 78K/0 Series Expansion

The 78K/0 Series consists of the following subseries. The part numbers enclosed with slanted rectangles are the subseries names.



Y subseries products are compatible with I²C bus.

Control

100-pin	μPD78075B	μPD78075BY	EMI noise reduction version of the μPD78078.
100-pin	μPD78078	μPD78078Y	A timer was added to the μPD78054, and external interface function was enhanced.
100-pin	μPD78070A	μPD78070AY	ROM-less versions of the μPD78078.
100-pin	μPD780018[Note]	μPD780018Y	Serial I/O of the μPD78078 was enhanced, and only selected functions are provided.
80-pin	μPD780058	μPD780058Y[Note]	Serial I/O of the μPD78054 was enhanced, EMI noise reduction version.
80-pin	μPD78058F	μPD78058FY	EMI noise reduction version of the μPD78054.
80-pin	μPD78054	μPD78054Y	UART and D/A converter were added to the μPD78014, and I/O was enhanced.
64-pin	μPD780034	μPD780034Y	An A/D converter of the μPD780024 was enhanced.
64-pin	μPD780024	μPD780024Y	Serial I/O of the μPD78018F was enhanced, EMI noise reduction version.
64-pin	μPD78014H		EMI noise reduction version of the μPD78018F.
64-pin	μPD78018F	μPD78018FY	Low-voltage (1.8 V) operation versions of the μPD78014 with several ROM and RAM capacities are available.
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 a low voltage (1.8 V).

Inverter control

64-pin	μPD780964	An A/D converter of the μPD780924 was enhanced.
64-pin	μPD780924	On-chip inverter control circuit and UART, EMI noise reduction version.

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	N-ch open-drain input/output 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	SIO of the μPD78064 was enhanced, and ROM and RAM were expanded.
100-pin	μPD78064B		EMI noise reduction version of the μPD78064.
100-pin	μPD78064	μPD78064Y	Basic subseries for driving LCDs, On-chip UART.

IEBus™ supported

80-pin	μPD78098B	EMI noise reduction version of the μPD78098.
80-pin	μPD78098	The IEBus controller was added to the μPD78054.

Meter control

80-pin	μPD780973	General-purpose model of automobile meter driving controller/driver of μPD780805.
100-pin	μPD780805	Internal automobile meter driving controller/driver.

LV

64-pin	μPD78P0914	On-chip PWM output, LV digital code decoder, Hsync counter.

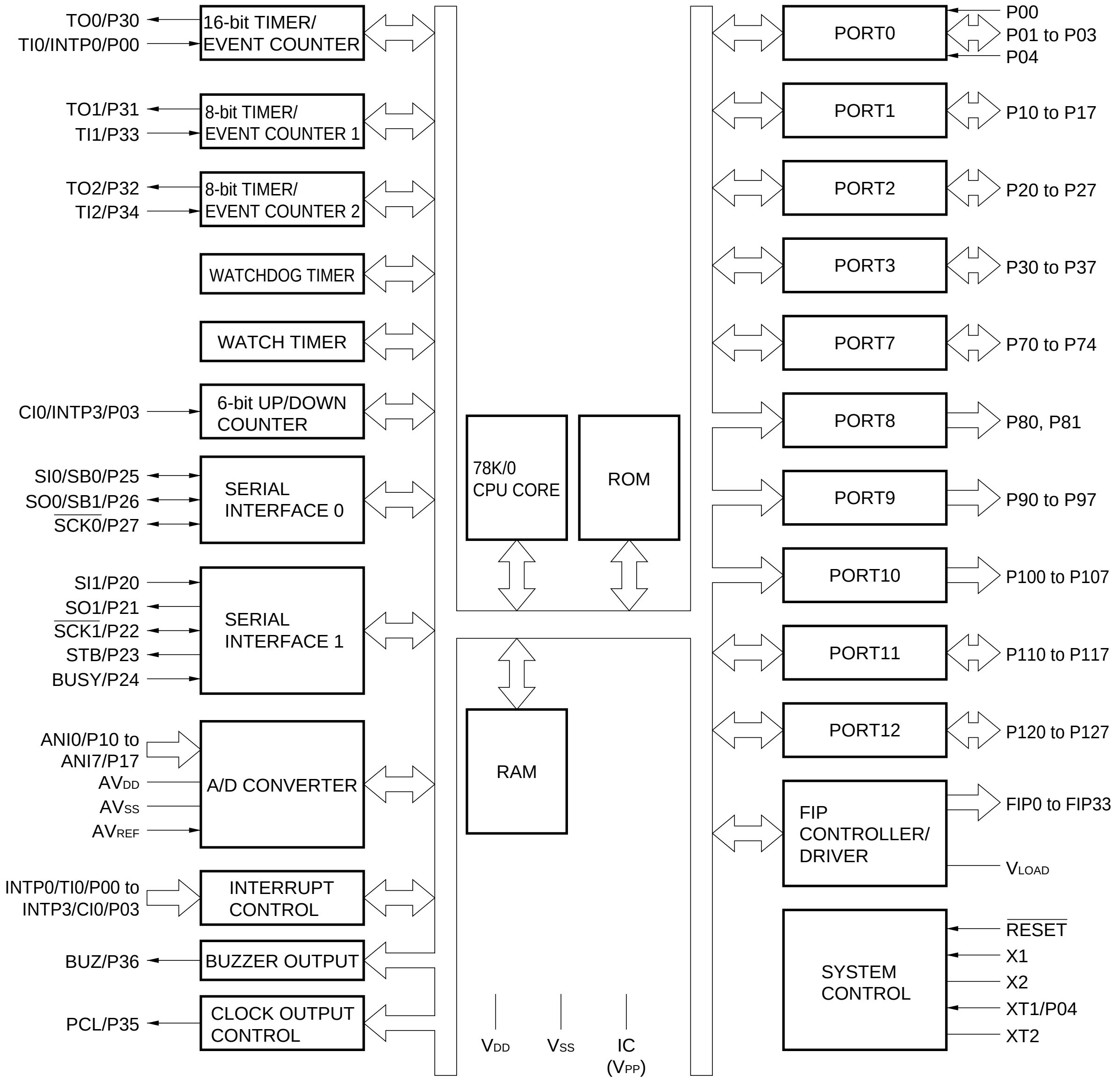
Note Under planning

The major functional differences among the subseries are shown below.

Function / Subseries Name		ROM Capacity	Timer				8-bit A/D	10-bit A/D	8-bit D/A	Serial Interface	I/O	VDD MIN. Value	External Expansion
			8-bit	16-bit	Watch	WDT							
Control	μPD78075B	32 K to 40 K	4 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	88	1.8 V	Available
	μPD78078	48 K to 60 K											
	μPD78070A	—									61	2.7 V	
	μPD780018	48 K to 60 K							–	2 ch (time-division 3-wire:1 ch)	88		
	μPD780058	24 K to 60 K	2 ch						2 ch	3 ch (time-division UART: 1 ch)	68	1.8 V	
	μPD78058F	48 K to 60 K								3 ch (UART: 1 ch)	69	2.7 V	
	μPD78054	16 K to 60 K										2.0 V	
	μPD780034	8 K to 32 K					–	8 ch	–	3 ch (UART: 1 ch, time-division 3-wire: 1 ch)	51	1.8 V	
	μPD780024						8 ch	–					
	μPD78014H									2 ch	53		
	μPD78018F	8 K to 60 K											
	μPD78014	8 K to 32 K										2.7 V	
	μPD780001	8 K		–	–					1 ch	39		–
	μPD78002	8 K to 16 K			1 ch		–				53		Available
	μPD78083				–		8 ch			1 ch (UART: 1 ch)	33	1.8 V	–
Inverter control	μPD780964	8 K to 32 K	3 ch	**Note**	–	1 ch	–	8 ch	–	2 ch (UART: 2 ch)	47	2.7 V	Available
	μPD780924						8 ch	–					
FIP drive	μPD780208	32 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	2 ch	74	2.7 V	–
	μPD780228	48 K to 60 K	3 ch	–	–					1 ch	72	4.5 V	
	μPD78044H	32 K to 48 K	2 ch	1 ch	1 ch						68	2.7 V	
	μPD78044F	16 K to 40 K								2 ch			
LCD drive	μPD780308	48 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	–	3 ch (time-division UART: 1 ch)	57	2.0 V	–
	μPD78064B	32 K								2 ch (UART: 1 ch)			
	μPD78064	16 K to 32 K											
IEBus supported	μPD78098B	40 K to 60 K	2 ch	1 ch	1 ch	1 ch	8 ch	–	2 ch	3 ch (UART: 1 ch)	69	2.7 V	Available
	μPD78098	32 K to 60 K											
Meter control	μPD780973	24 K to 32 K	3 ch	1 ch	1 ch	1 ch	5 ch	–	–	2 ch (UART: 1 ch)	56	4.5 V	–
	μPD780805	40 K to 60 K	2 ch				8 ch				39	2.7 V	
LV	μPD78P0914	32 K	6 ch	–	–	1 ch	8 ch	–	–	2 ch	54	4.5 V	Available

Note 10-bit timer: 1 channel

1.7 Block Diagram



Remarks 1. The ROM and RAM capacities vary depending on the part number.

2. Pin names in parentheses apply to the μPD78P048A only.

1.8 Outline of Function

Item		Part Number	μPD78042F	μPD78043F	μPD78044F	μPD78045F	μPD78P048A
Internal memory	ROM		Mask ROM				One-time PROM/EPROM
			16 Kbytes	24 Kbytes	32 Kbytes	40 Kbytes	60 Kbytes[Note 1]
	High-speed RAM 512 bytes		1024 bytes		1024 bytes		1024 bytes[Note 2]
	Expansion RAM		—				1024 bytes[Note 3]
	Buffer RAM		64 bytes				
	FIP display RAM		48 bytes				
General register			8 bits × 8 × 4 banks				
Minumum instruction execution time	With main system clock selected		0.4 μs/0.8 μs/1.6 μs/3.2 μs/6.4 μs (when operated at 5.0 MHz)				
	With subsystem clock selected		122 μs (when operated at 32.768 kHz)				
Instruction set			• 16-bit operation • Multiply/divide (8-bit × 8-bit, 16-bit / 8-bit) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, and other related operations				
I/O port (Includes FIP dual-function pins)			Total : 68 I/O pins • CMOS input : 2 pins • CMOS input/output : 27 pins • N-ch open-drain I/O : 5 pins • P-ch open-drain I/O : 16 pins • P-ch open-drain output : 18 pins				
FIP controller/driver			Total of display output : 34 outputs • Segments : 9 to 24 pins • Digits : 2 to 16 pins				
A/D converter			• 8-bit resolution × 8 channels • Power supply voltage : AV_{DD} = 4.0 to 5.5 V				
Serial interface			• 3-wire serial I/O/SBI/2-wire serial I/O mode selection possible : 1 channel • 3-wire serial I/O mode (Maximum 64-byte on-chip automatic transmit/receive function) : 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 • 6-bit up/down counter : 1 channel				
Timer output			3 outputs: (14-bit PWM generation possible from one output)				
Clock output			19.5 kHz, 39.1 kHz, 78.1 kHz, 156 kHz, 313 kHz, 625 kHz (@ 5.0 MHz with main system clock) 32.768 kHz (@ 32.768 kHz with subsystem clock)				

Notes 1. 16 K, 24 K, 32 K, 40 K or 60 Kbytes can be selected by the memory size switching register (IMS).

2. 512 or 1024 bytes can be selected by IMS.

3. 0 or 1024 bytes can be selected by the internal expansion RAM switching register (IXS).

Item / Part Number		μPD78042F	μPD78043F	μPD78044F	μPD78045F	μPD78P048A
Buzzer output		1.2 kHz, 2.4 kHz, 4.9 kHz (@ 5.0 MHz with main system clock)				
Vectored interrupt source	Maskable	Internal : 10 External : 4				
	Non-maskable	Internal: 1				
	Software	1				
Test input		Internal : 1				
Power supply voltage		V_{DD} = 2.7 to 5.5 V				
Package		• 80-pin plastic QFP (14 × 20 mm) • 80-pin ceramic WQFN: μPD78P048A only				

1.9 Mask Options

The mask ROM versions (μPD78042F, μPD78043F, μPD78044F, μPD78045F) have mask options. By specifying the mask options when ordering, the pull-up resistors and pull-down resistors listed in Table 1-1 can be incorporated. When these resistors are necessary, the number of external components and mounting space can be saved by utilizing the mask options.

Table 1-1. Mask Options in Mask ROM Versions

Pin Name	Mask Option
P30/TO0 to P32/TO2, P33/TI1, P34/TI2, P35/PCL, P36/BUZ, P37	On-chip pull-down resistor specifiable bit-wise.
P70 to P74	On-chip pull-up resistor specifiable bit-wise.
P80/FIP0, P81/FIP1	On-chip pull-down resistor specifiable bit-wise. Connection to V_{LOAD} or V_{SS} specifiable in 2-bit units from P80.
P90/FIP2 to P97/FIP9, P100/FIP10 to P107/FIP17, P110/FIP18 to P117/FIP25, P120/FIP26 to P127/FIP33	On-chip pull-down resistor specifiable bit-wise. Connection to V_{LOAD} or V_{SS} specifiable in 4-bit units from P90.

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

2.1.1 Normal operating mode pins

(1) Port pins (1/3)

Pin Name	Input/ Output	Function		After Reset	Alternate Function Pin
P00	Input	Port 0. 5-bit input/output port.	Input only	Input	INTP0/TI0
P01	Input/ output		Input/output specifiable bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software.	Input	INTP1
P02					INTP2
P03					INTP3/CI0
P04[Note 1]	Input		Input only	Input	XT1
P10 to P17	Input/ output	Port 1. 8-bit input/output port. Input/output specifiable bit-wise. If used as input port, an on-chip pull-up resistor can be connected by software.[Note 2]		Input	ANI0 to ANI7
P20	Input/ output	Port 2. 8-bit input/output port. Input/output specifiable bit-wise. If used as an input port, an on-chip 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. When the P04/XT1 pin is used as an input port, set the bit 6 (FRC) of the processor clock control register (PCC) to 1 (do not use the feedback resistor contained in the subsystem clock oscillator).

2. When pins P10/ANI0 to P17/ANI7 are used as an analog input of the A/D converter, set the port 1 to the input mode. In this case, its on-chip pull-up resistor will be automatically disabled.

(1) Port pins (2/3)

Pin Name	Input/ Output	Function	After Reset	Alternate Function Pin
P30	Input/ output	Port 3. 8-bit input/output port. LED can be driven directly. Input/output specifiable bit-wise. If used as an input port, an on-chip pull-up resistor can be connected by software. Mask ROM version can specify pull-down resistors bit-wise with the mask option.	Input	TO0
P31				TO1
P32				TO2
P33				TI1
P34				TI2
P35				PCL
P36				BUZ
P37				—
P70 to P74	Input/ output	Port 7. N-ch open drain 5-bit I/O port. Input/output specifiable bit-wise. LED can be driven directly. Mask ROM version can specify pull-up resistors bit-wise with the mask option.	Input	—
P80, P81	Output	Port 8. P-ch open drain 2-bit high breakdown voltage output port. LED can be driven directly. Mask ROM version can specify pull-down resistors bit-wise with the mask option (connection to V_{LOAD} or V_{SS} specifiable in 2-bit units). The μPD78P048A has on-chip pull-down resistors (connected to V_{LOAD}).	Output	FIP0, FIP1
P90 to P97	Output	Port 9. P-ch open drain 8-bit high breakdown voltage output port. LED can be driven directly. Mask ROM version can specify pull-down resistors bit-wise with the mask option (connection to V_{LOAD} or V_{SS} specifiable in 4-bit units). The μPD78P048A has on-chip pull-down resistors (connected to V_{LOAD}).	Output	FIP2 to FIP9
P100 to P107	Output	Port 10. P-ch open drain 8-bit high breakdown voltage output port. LED can be driven directly by P100 to P105. Mask ROM version can specify pull-down resistors bit-wise with the mask option (connection to V_{LOAD} or V_{SS} specifiable in 4-bit units). The μPD78P048A has on-chip pull-down resistors for P100 to P105 (connected to V_{LOAD}).	Output	FIP10 to FIP7

(1) Port pins (3/3)

Pin Name	Input/ Output	Function	After Reset	Alternate Function Pin
P110 to P117	Input/ output	Port 11. P-ch open-drain 8-bit high breakdown voltage input/output port. LED can be driven directly. Input/output specifiable bit-wise. Mask ROM version can specify a pull-down resistor bit-wise with the mask option (connection to V_{LOAD} or V_{SS} specifiable in 4-bit units).	Input	FIP18 to FIP25
P120 to P127	Input/ output	Port 12. P-ch open-drain 8-bit high breakdown voltage input/output port. LED can be driven directly. Input/output specifiable bit-wise. Mask ROM version can specify pull-down resistors bit-wise with the mask option (connection to V_{LOAD} or V_{SS} specifiable in 4-bit units).	Input	FIP26 to FIP33

(2) Non-Port Pins (1/2)

Pin Name	Input/ Output	Function	After Reset	Alternate Function Pin
INTP0	Input	External interrupt request inputs with specifiable valid edges (rising edge, falling edge, both rising and falling edges).	Input	P00/TI0
INTP1				P01
INTP2				P02
INTP3		External interrupt input with falling edge detection.		P03/CI0
SI0	Input	Serial interface serial data input.	Input	P25/SB0
SI1				P20
SO0	Output	Serial interface serial data output.	Input	P26/SB1
SO1				P21
SB0	Input/ output	Serial interface serial data input/output.	Input	P25/SI0
SB1				P26/SO0
$\overline{\text{SCK0}}$	Input/ output	Serial interface serial clock input/output.	Input	P27
$\overline{\text{SCK1}}$				P22
STB	Output	Serial interface automatic transmit/receive strobe output.	Input	P23
BUSY	Input	Serial interface automatic transmit/receive busy input.	Input	P24
TI0	Input	Input of external count clock to 16-bit timer (TM0).	Input	P00/INTP0
TI1		Input of external count clock to 8-bit timer (TM1).		P33
TI2		Input of external count clock to 8-bit timer (TM2).		P34
TO0	Output	16-bit timer (TM0) output (alternate function as 14-bit PWM output).	Input	P30
TO1		8-bit timer (TM1) output.		P31
TO2		8-bit timer (TM2) output.		P32
CI0	Input	6-bit up/down counter clock input.	Input	P03/INTP3
PCL	Output	Clock output (for main system clock and subsystem clock trimming).	Input	P35
BUZ	Output	Buzzer output.	Input	P36
FIP0, FIP1	Output	High breakdown voltage and high current output for FIP controller/ driver digit output.	Output	P80, P81
FIP2 to FIP9				P90 to P97
FIP10 to FIP15	Output	High breakdown voltage and high current output for FIP controller/ driver digit/segment output.	Output	P100 to P105
FIP16, FIP17	Output	High breakdown voltage output for FIP controller/driver segment output.	Output	P106, P107
FIP18 to FIP25			Input	P110 to P117
FIP26 to FIP33				P120 to P127
V_{LOAD}	—	Pull-down resistor connection for FIP controller/driver.	—	—
ANI0 to ANI7	Input	A/D converter analog input.	Input	P10 to P17
AV_{REF}	Input	A/D converter reference voltage input.	—	—

(2) Non-Port Pins (2/2)

Pin Name	Input/ Output	Function	After Reset	Alternate Function Pin
AV_{DD}	—	A/D converter analog power supply. Connected to V_{DD}.	—	—
AV_{SS}	—	A/D converter ground potential. Connected to V_{SS}.	—	—
$\overline{\text{RESET}}$	Input	System reset input.	—	—
X1	Input	Crystal resonator connection for main system clock oscillation.	—	—
X2	—		—	—
XT1	Input	Crystal resonator connection for subsystem clock oscillation.	Input	P04
XT2	—		—	—
V_{DD}	—	Positive power supply.	—	—
V_{PP}	—	Applies high voltage for writing/verifying program. Directly connect to V_{SS} in normal operation mode.	—	—
V_{SS}	—	Ground potential.	—	—
IC	—	Internally connected. Directly connected to V_{SS}.	—	—

2.1.2 PROM programming mode pins (μPD78P048A only)

Pin Name	Input/ Output	Function
$\overline{\text{RESET}}$	Input	PROM programming mode setting. When +5 V or +12.5 V is applied to the V_{PP} pin or a low level voltage is applied to the $\overline{\text{RESET}}$ pin, the PROM programming mode is set.
V_{PP}	Input	High-voltage application for PROM programming mode setting and program write/verify.
A0 to A16	Input	Address bus.
D0 to D7	Input/ output	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 potential.

2.2 Description of Pin Functions

2.2.1 P00 to P04 (Port 0)

These are 5-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt request input, an external count clock input to the timer, a capture trigger signal input and crystal resonator connection for subsystem clock oscillation.

The following operating modes can be specified bit-wise.

(1) Port mode

P00 and P04 function as input-only ports and P01 to P03 function as input/output ports.

P01 to P03 can be specified for input or output ports bit-wise with port mode register 0 (PM0). When they are used as input ports, a pull-up resistor can be used for them with an on-chip pull-up resistor option register (PUO).

(2) Control mode

In this mode, these ports function as an external interrupt request input, an external count clock input to the timer, and crystal connection for subsystem clock oscillation.

(a) INTP0 to INTP3

INTP0 to INTP2 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges). INTP0 become a 16-bit timer/event counter capture trigger signal input pin with a valid edge input. INTP3 become a falling edge detection external interrupt request input pin.

(b) TI0, CI0

TI0 is a pin for external count clock input to 16-bit timer/event counter, and CI0 is a pin for external count clock input to 6-bit up/down counter.

(c) XT1

Crystal connect pin for subsystem clock oscillation

2.2.2 P10 to P17 (Port 1)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as an A/D converter analog input.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with a port mode register 1 (PM1). If used as an input port, a pull-up resistor can be used for these ports with an on-chip pull-up resistor option register (PUO).

(2) Control mode

These ports function as A/D converter analog input pins (ANI0 to ANI7). The on-chip pull-up resistor is automatically disabled when the pins are specified for analog input.

2.2.3 P20 to P27 (Port 2)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as data input/output to/from the serial interface, clock input/output, automatic transmit/receive busy input, and strobe output functions.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with a port mode register 2 (PM2). When they are used as input ports, a pull-up resistor can be used to them with an on-chip pull-up resistor option register (PUO).

(2) Control mode

These ports function as serial interface data input/output, clock input/output, automatic transmit/receive busy input, and strobe output functions.

(a) SI0, SI1, SO0, SO1

Serial interface serial data input/output pins

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

Serial interface serial clock input/output pins

(c) SB0 and SB1

NEC standard serial bus interface input/output pins

(d) BUSY

Serial interface automatic transmit/receive busy input pins

(e) STB

Serial interface automatic transmit/receive strobe output pins

Caution If port 2 is used as a serial interface, the I/O and output latches must be set according to its functions. For the setting method, refer to Figure 14-3, "Serial Operating Mode Register 0 Format", and Figure 15-3, "Serial Operating Mode Register 1 Format".

2.2.4 P30 to P37 (Port 3)

These are 8-bit input/output ports. Beside serving as input/output ports, they function as timer input/output, clock output and buzzer output.

Mask ROM version can contain pull-down resistors with the mask option.

Port 3 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. They can be specified bit-wise as input or output ports with port mode register 3 (PM3). When they are used as input ports, a pull-up resistor can be used with an on-chip pull-up resistor option register (PUO).

(2) Control mode

These ports function as timer input/output, clock output, and buzzer output.

(a) TI1 and TI2

Pin for external count clock input to the 8-bit timer/event counter.

(b) TO0 to TO2

Timer output pins

(c) PCL

Clock output pin

(d) BUZ

Buzzer output pin

2.2.5 P70 to P74 (Port 7)

These are 5-bit input/output ports. They can be specified bit-wise for input or output ports by port mode register 7 (PM7).

Port 7 can drive LEDs directly.

P70 to P74 are N-ch open-drain outputs.

Mask ROM version can contain pull-up resistors with the mask option.

2.2.6 P80 and P81 (Port 8)

These are 2-bit output dedicated ports. Besides serving as output port, they function as an digit output for FIP controller/driver.

Port 8 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 2-bit output dedicated ports. P80 and P81 are P-ch open-drain. Mask ROM version can contain pull-down resistors with the mask option. The μPD78P048A contains pull-down resistors.

(2) Control mode

These ports function as digit output pins (FIP0, FIP1) of the FIP controller/driver.

2.2.7 P90 to P97 (Port 9)

These are 8-bit output dedicated ports. Besides serving as output ports, they function as a digit output for FIP controller/driver.

They can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit output dedicated ports.

P90 to P97 are P-ch open-drain outputs. Mask ROM version can contain pull-up resistors with the mask option. The μPD78P048A contains pull-down resistors.

(2) Control mode

These ports function as pins for digit output of the FIP controller/driver (FIP2 to FIP9).

2.2.8 P100 to P107 (Port 10)

These are 8-bit output dedicated ports. In addition to their use as an output port, they also have FIP controller/driver digit/segment outputs and segment output functions.

P100 to P105 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

Port 10 functions as an 8-bit output dedicated port. P100 to P107 are P-ch open-drain outputs. Mask ROM version can contain pull-down resistors with the mask option.

The μPD78P048A contains pull-down resistors in P100 to P105.

(2) Control mode

P100 to P105 function as digit/segment output pins for FIP controller/driver (FIP10 to FIP15). P106 and P107 function as segment output pins for FIP controller/driver (FIP16, FIP17).

2.2.9 P110 to P117 (Port 11)

These are 8-bit input/output ports. Besides serving as input/output ports, they function as a segment output for FIP controller/driver.

P110 to P117 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. P110 to P117 are P-ch open-drain outputs. Mask ROM version can contain pull-down resistors with the mask option.

(2) Control mode

These ports function as segment output pins for FIP controller/driver (FIP18 to FIP25).

2.2.10 P120 to P127 (Port 12)

These are 8-bit input/output ports. Besides serving as input/output ports, they are used for segment output pins for FIP controller/driver.

P120 to P127 can drive LEDs directly.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 8-bit input/output ports. P120 to P127 are P-ch open-drain. Mask ROM version can contain pull-down resistors with the mask option.

(2) Control mode

These ports function as segment output pins for FIP controller/driver (FIP26 to FIP33).

2.2.11 AV_{REF}

The A/D converter's reference voltage should be input from this pin.

2.2.12 AV_{DD}

This pin supplies power for A/D converter operations dedicatedly. Always apply to this pin the same voltage as the V_{DD} pin voltage even if the A/D converter is not used.

2.2.13 AV$_{SS}$

This pin supplies the ground level for A/D converter operations dedicatedly. Always apply to this pin the same voltage as the V$_{SS}$ pin voltage even if the A/D converter is not used.

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

This is an active-low system reset input pin.

2.2.15 X1 and X2

These are crystal resonator connection pins for main system clock oscillation. For external clock supply, input it to X1 and its inverted signal to X2.

2.2.16 XT1 and XT2

These are crystal resonator connection pins for subsystem clock oscillation.

For external clock supply, input it to XT1 and its inverted signal to XT2.

2.2.17 V$_{DD}$

Positive power should be supplied from this pin.

2.2.18 V$_{SS}$

Ground potential should be supplied from this pin.

2.2.19 V$_{PP}$ (μPD78P048A only)

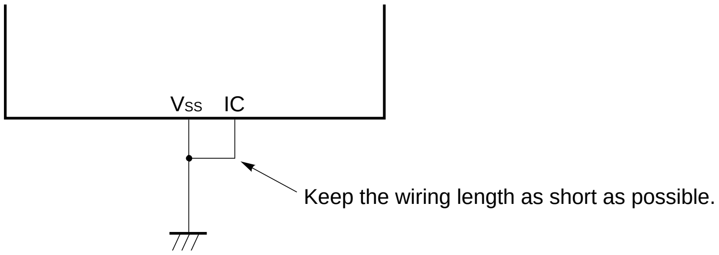
A high-voltage should be applied to this pin during PROM programming mode setting and in program write/verify mode. Connect directly to V$_{SS}$ in normal operating mode.

2.2.20 IC (Mask ROM version only)

The IC (Internally Connected) pin sets a test mode in which the μPD78042F, 78043F, 78044F and 78045F are tested before shipment. In normal operation mode, connect the IC pin directly to V$_{SS}$ with as short a wiring length as possible.

If there is a potential difference between the IC and V$_{SS}$ pins because the wiring length between the IC and V$_{SS}$ pin is too long, or external noise is superimposed on the IC pin, your program may not run correctly.

• Directly connect the IC pin to the V$_{SS}$.



2.3 Input/Output Circuit and Recommended Connection of Unused Pins

Table 2-1 shows the input/output circuit types of pins and the recommended connections for unused pins. Refer to Figure 2-1 for the configuration of the input/output circuit of each type.

Table 2-1. Pin Input/Output Circuit Types (1/2)

Pin Name	Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
P00/INTP0/TI0	2	Input	Connect to V_{SS}.
P01/INTP1	8-A	Input/output	Connect to V_{SS} via a resistor independently.
P02/INTP2			
P03/INTP3/CI0			
P04/XT1	16	Input	Connect to V_{DD} or V_{SS}.
P10/ANI0 to P17/ANI7	11	Input/output	Connect to V_{SS} or V_{DD} via a resistor independently.
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/$\overline{\text{SCK0}}$			
Mask ROM version			
P30/TO0	5-C	Input/output	Connect to V_{DD} or V_{SS} via a resistor independently.
P31/TO1			
P32/TO2			
P33/TI1	8-B		
P34/TI2			
P35/PCL	5-C		
P36/BUZ			
P37			

Table 2-1. Pin Input/Output Circuit Types (2/2)

Pin Name		Input/Output Circuit Type	Input/Output	Recommended Connection of Unused Pins
μPD78P048A				
	P30/TO0	5-A	Input/output	Connect to V_{DD} or V_{SS} via a resistor independently.
	P31/TO1			
	P32/TO2			
	P33/TI1	8-A		
	P34/TI2			
	P35/PCL	5-A		
	P36/BUZ			
	P37			
Mask ROM version				
	P70 to P74	13-B	Input/output	Connect to V_{DD} or V_{SS} via a resistor independently.
	P80/FIP0, P81/FIP1	14-A	Output	Open
	P90/FIP2 to P97/FIP9			
	P100/FIP10 to P107/FIP17			
	P110/FIP18 to P117/FIP25	15-C	Input/output	Connect to V_{DD} or V_{SS} via a resistor independently.
	P120/FIP26 to P127/FIP33			
	IC	—	—	Connect directly to V_{SS}.
μPD78P048A				
	P70 to P74	13-D	Input/output	Connect to V_{DD} or V_{SS} via a resistor independently.
	P80/FIP0, P81/FIP1	14	Output	Open
	P90/FIP2 to P97/FIP9			
	P100/FIP10 to P105/FIP15			
	P106/FIP16, P107/FIP17	14-B	Output	Open
	P110/FIP18 to P117/FIP25	15-B	Input/output	Connect to V_{DD} or V_{SS} via a resistor independently.
	P120/FIP26 to P127/FIP33			
	V_{PP}	—	—	Connect directly to V_{SS}.
$\overline{RESET}$		2	Input	—
XT2		16	—	Open
AV_{REF}		—		Connect to V_{SS}.
AV_{DD}				Connect to V_{DD}.
AV_{SS}				Connect to V_{SS}.
V_{LOAD}				

Figure 2-1. Pin Input/Output Circuit List (1/3)

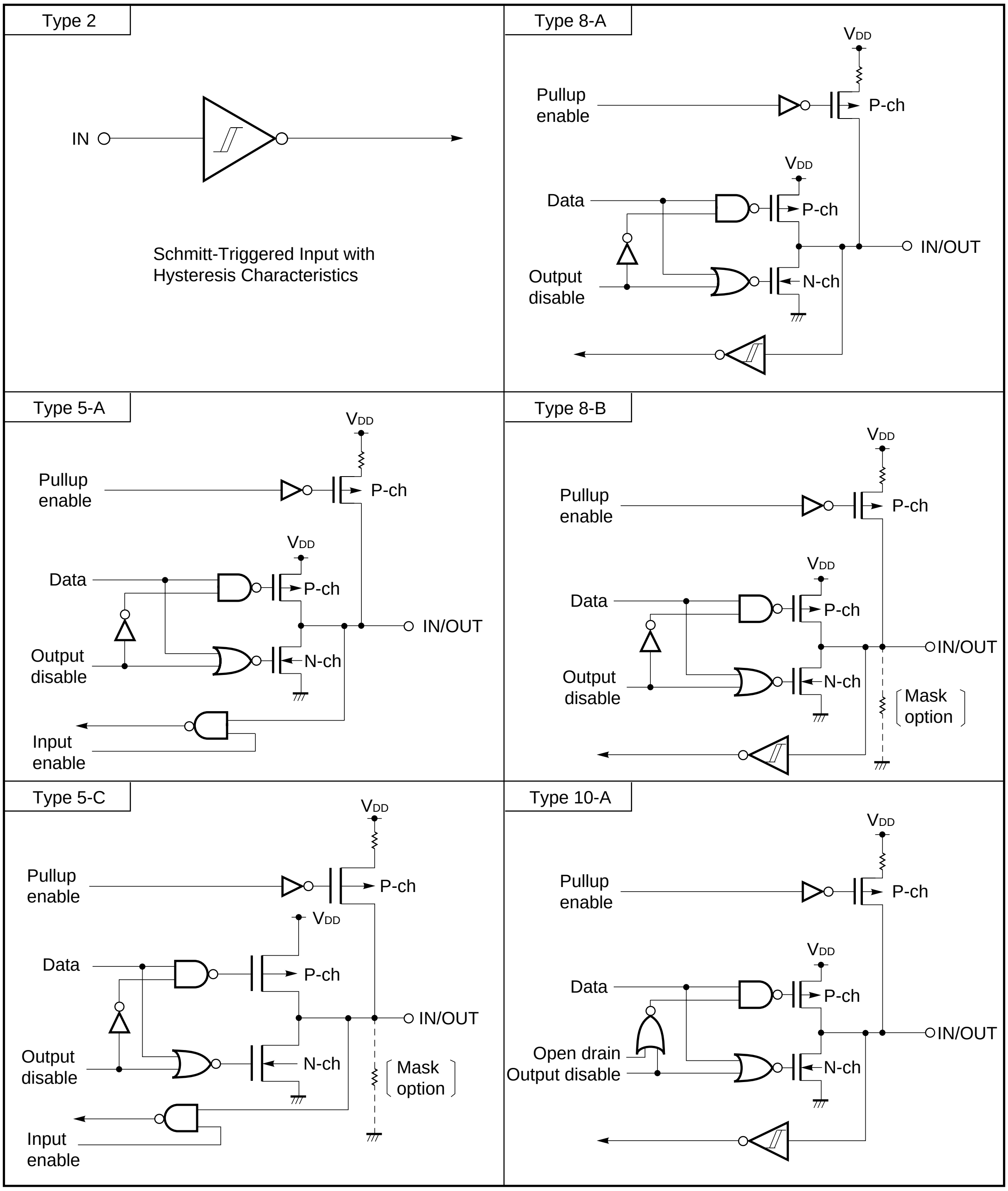


Figure 2-1. Pin Input/Output Circuit List (2/3)

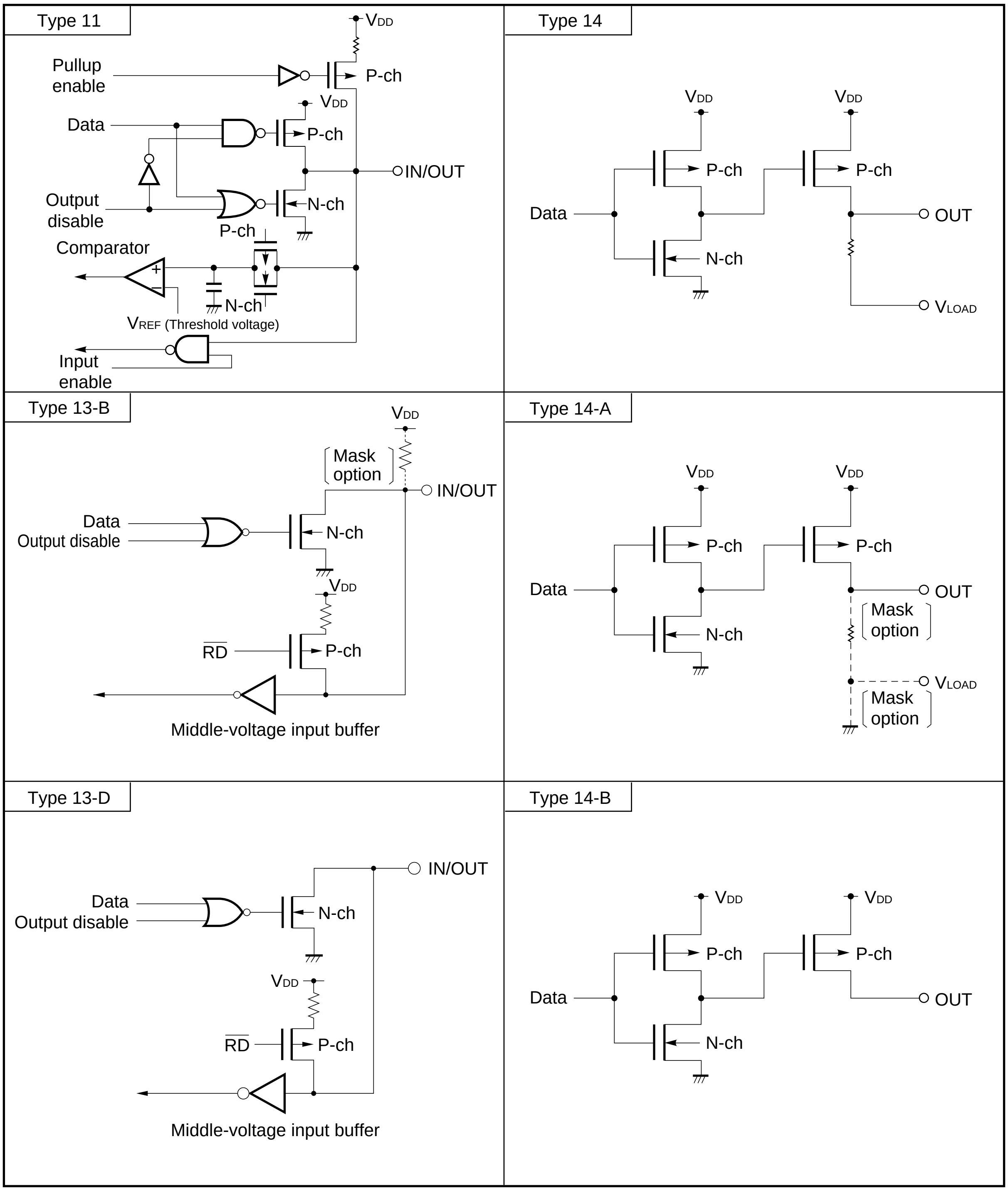
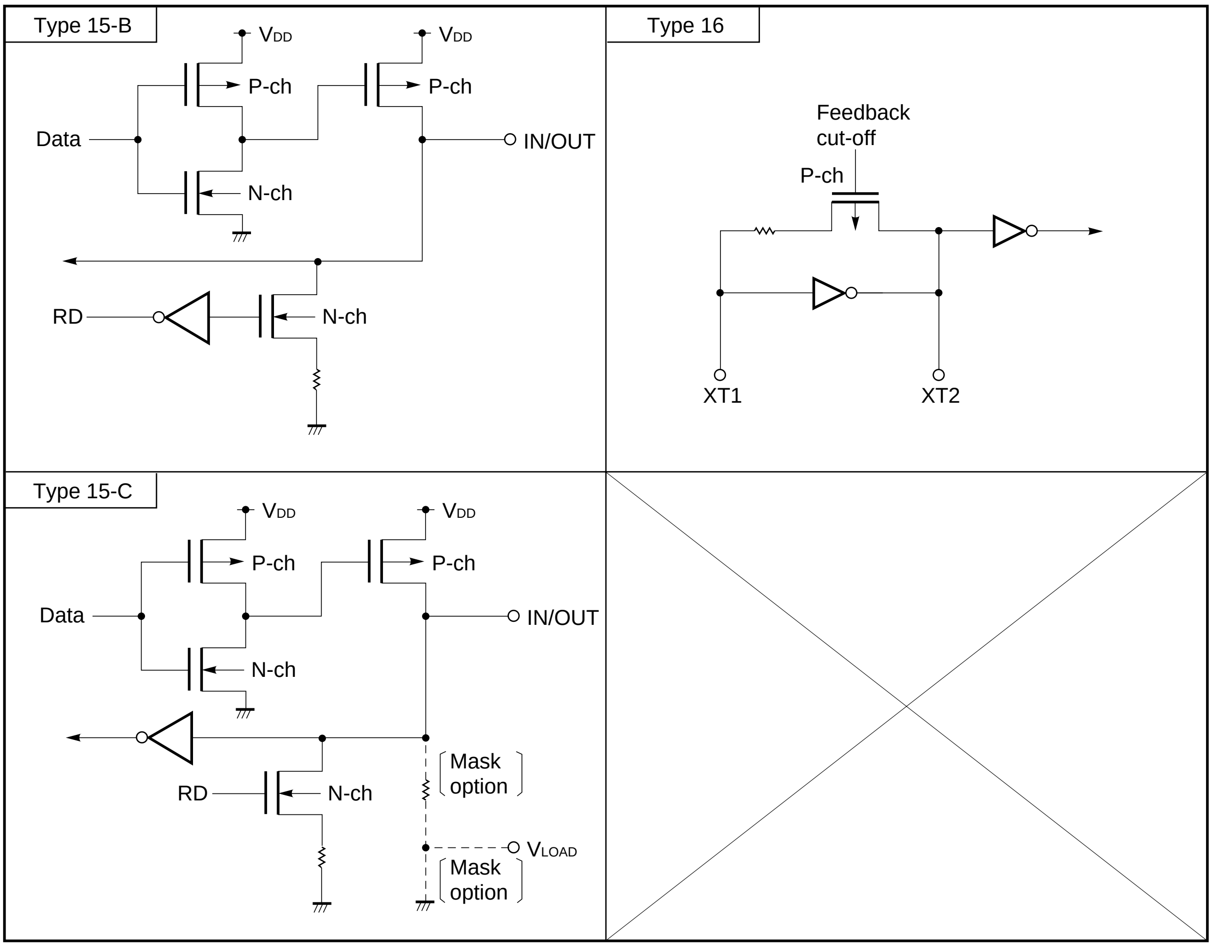


Figure 2-1. Pin Input/Output Circuit List (3/3)



[MEMO]

CHAPTER 3   CPU ARCHITECTURE

3.1 Memory Spaces

Each μPD78044F Subseries device accesses a memory space of 64 Kbytes. Figures 3-1 to 3-5 show memory maps.

Figure 3-1. Memory Map (μPD78042F)

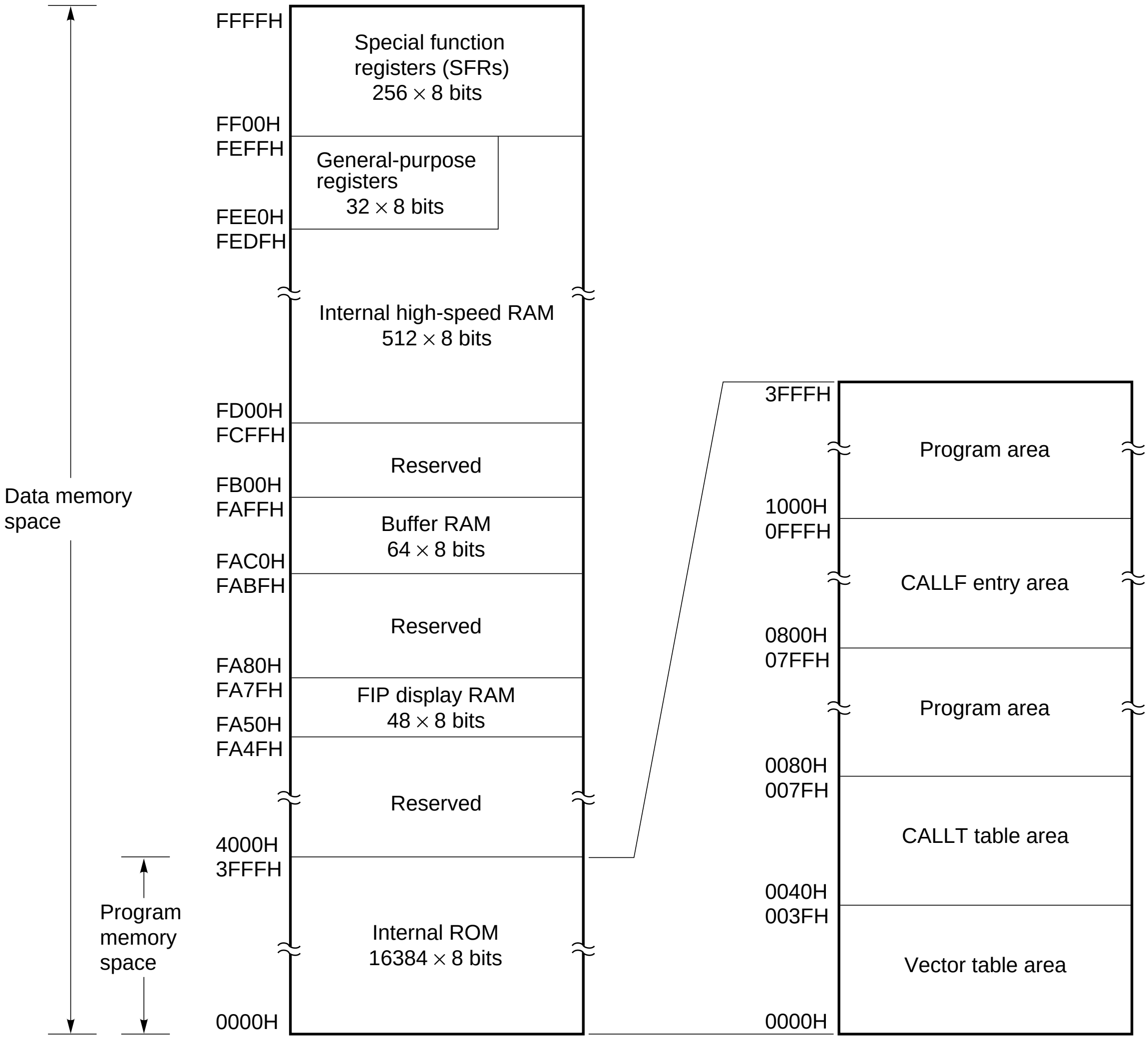


Figure 3-2. Memory Map (μPD78043F)

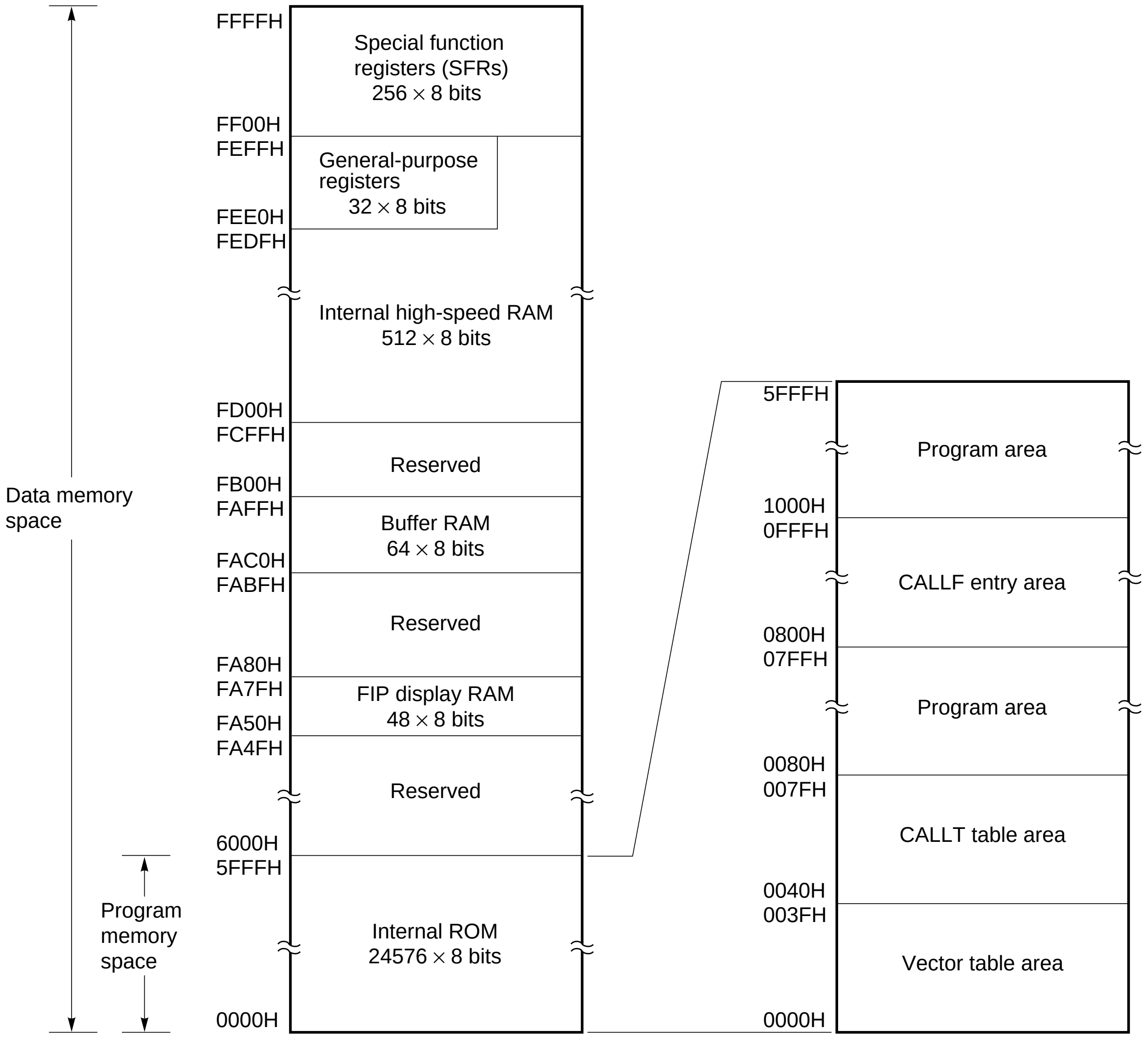


Figure 3-3. Memory Map (μPD78044F)

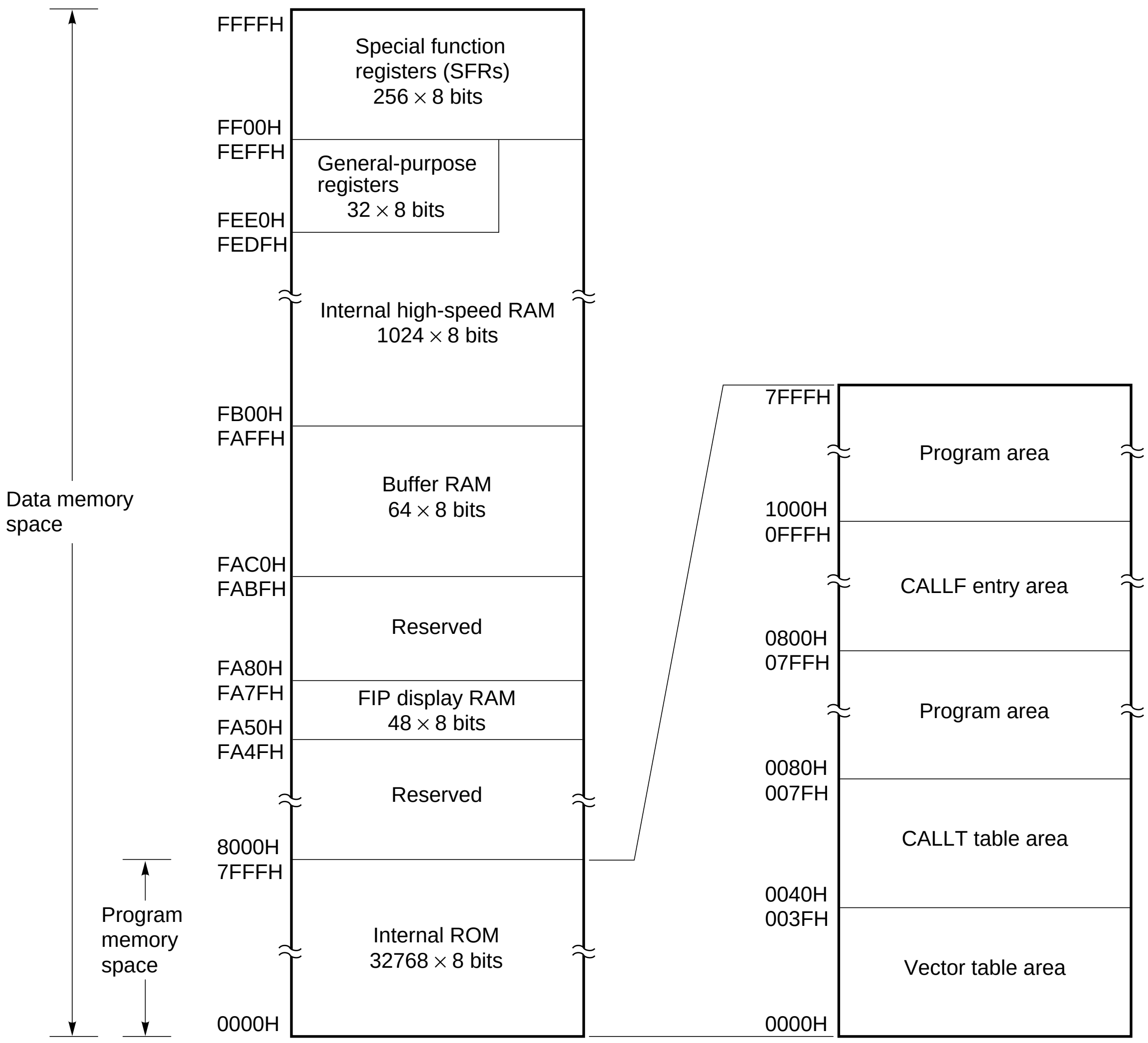


Figure 3-4. Memory Map (μPD78045F)

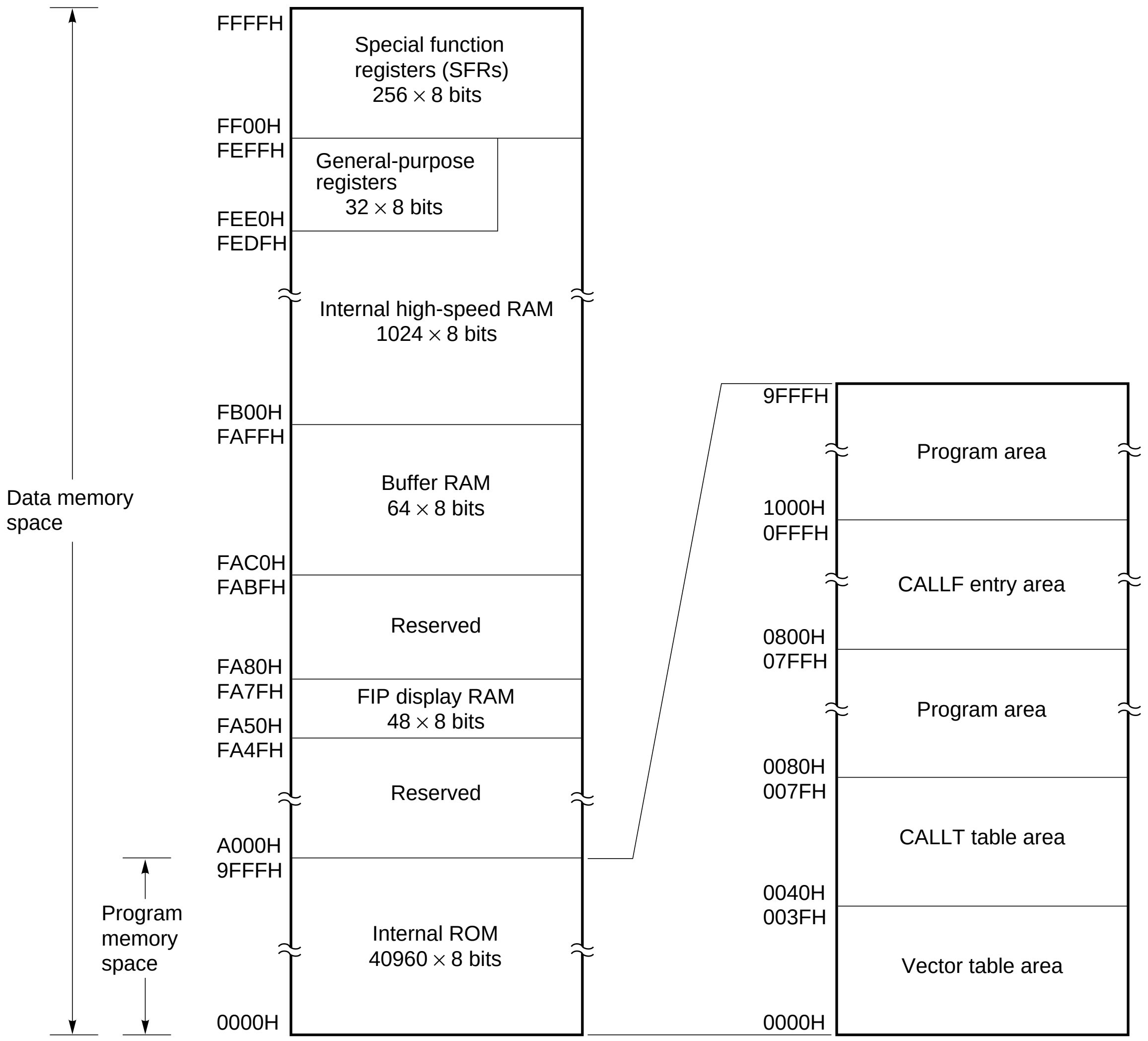
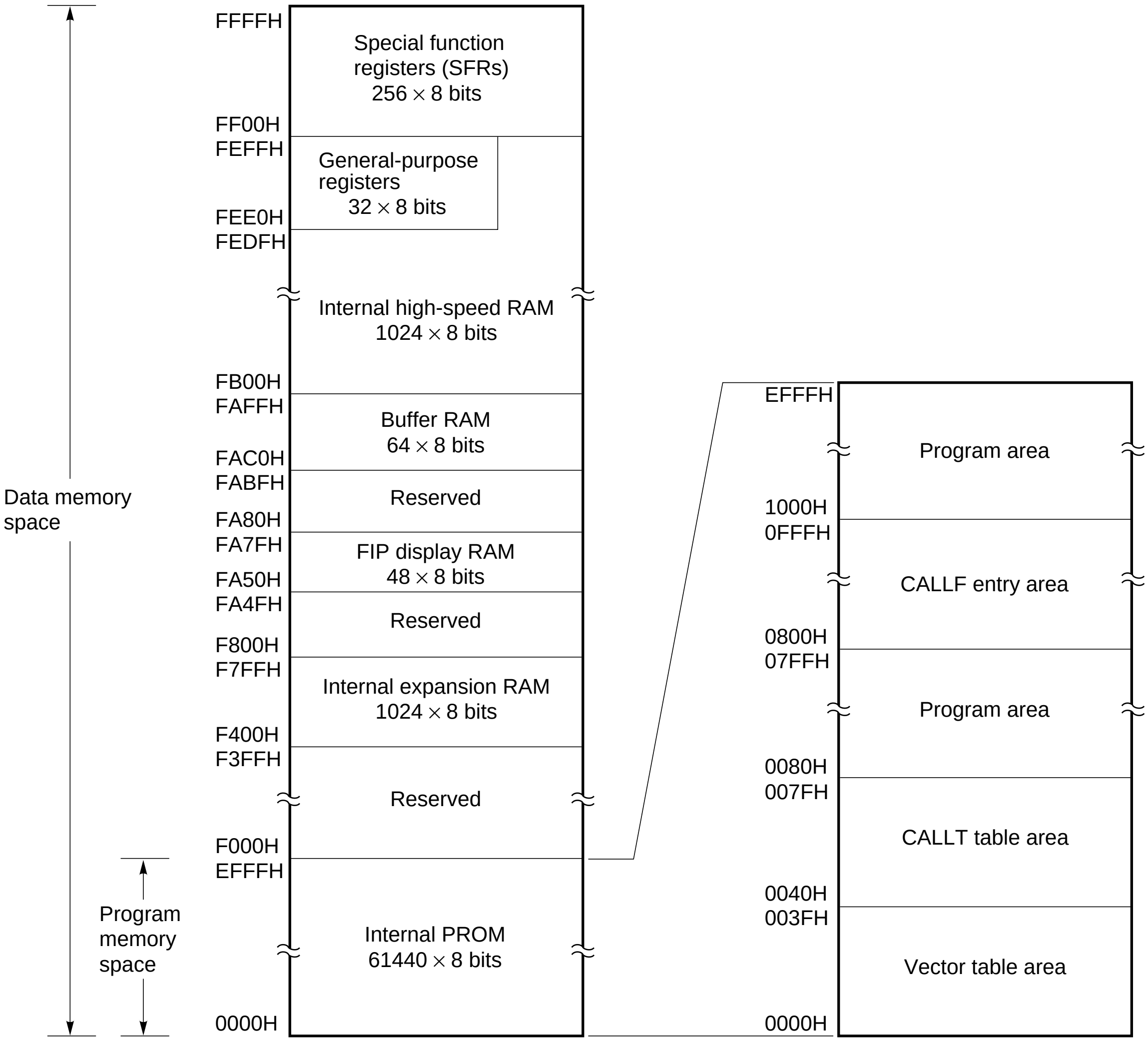


Figure 3-5. Memory Map (μPD78P048A)



3.1.1 Internal program memory space

Internal program memory store programs and table data. Normally, they are addressed with a program counter (PC).

The μPD78044F Subseries contains internal ROM (or PROM) in each product having the capacity shown below.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Configuration	Capacity
μPD78042F	Mask ROM	16384 × 8 bits
μPD78043F	Mask ROM	24576 × 8 bits
μPD78044F	Mask ROM	32768 × 8 bits
μPD78045F	Mask ROM	40960 × 8 bits
μPD78P048A	PROM	61440 × 8 bits

The following areas are allocated in the internal program memory space.

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as vector table area. The RESET input and program start addresses for branch upon generation of each interrupt request are stored in the vector table area. Of the 16-bit address, the low-order 8 bits are stored at even addresses and the high-order 8 bits are stored at odd addresses.

Table 3-2. Vector Table

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

(2) CALLT instruction table area

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

(3) CALLF instruction entry area

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

3.1.2 Internal data memory space

The μPD78044F Subseries units incorporate the following RAMs.

(1) Internal high-speed RAM

The μPD78044F Subseries units incorporate the following capacity of internal high-speed RAM in each product.

Table 3-3. Internal High-Speed RAM Capacities

Part Number	Internal High-Speed RAM Capacity
μPD78042F	512 × 8 bits
μPD78043F	512 × 8 bits
μPD78044F	1024 × 8 bits
μPD78045F	1024 × 8 bits
μPD78P048A	1024 × 8 bits

4 banks of general registers, each bank consisting of eight 8-bit registers are allocated in the 32-byte area FEE0H to FEFFH.

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

(2) Internal Expansion RAM

Internal expansion RAM is allocated to the 1024-byte area from F400H to F7FFH of the μPD78P048A.

(3) Buffer RAM

Buffer RAM is allocated to the 64-byte area from FAC0H to FAFFH. The buffer RAM is used to store the transmit/receive data of serial interface channel 1 (3-wire serial I/O mode with automatic transmit/receive function). When the 3-wire serial I/O mode with automatic transmit/receive function is not used, the buffer RAM can also be used as normal RAM.

(4) FIP display RAM

FIP display RAM is allocated to the 48-byte area from FA50H to FA7FH. FIP display RAM can also be used as normal RAM.

3.1.3 Special function register (SFR) area

An on-chip peripheral hardware special-function register (SFR) is allocated in the area FF00H to FFFFH (Refer to **Table 3-4 Special Function Register List** of **3.2.3 Special function register (SFR)**).

Caution Do not access addresses where the SFR is not assigned.

3.1.4 Data memory addressing

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

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

To address a memory area to be manipulated when an instruction is executed, the μPD78044F Subseries has many addressing modes to improve the operability. Especially, in the areas to which the data memory is assigned (addresses FD00H to FFFFH for the μPD78042F and 78043F, and FB00H to FFFFH for the μPD78044F, 78045F, and 78P048A), the special function registers (SFRs) and general-purpose registers can be addressed in accordance with their function.

Data memory addressing is shown in Figures 3-6 to 3-10.

For details of each addressing, refer to **3.4 Operand Address Addressing**.

Figure 3-6. Data Memory Addressing (μPD78042F)

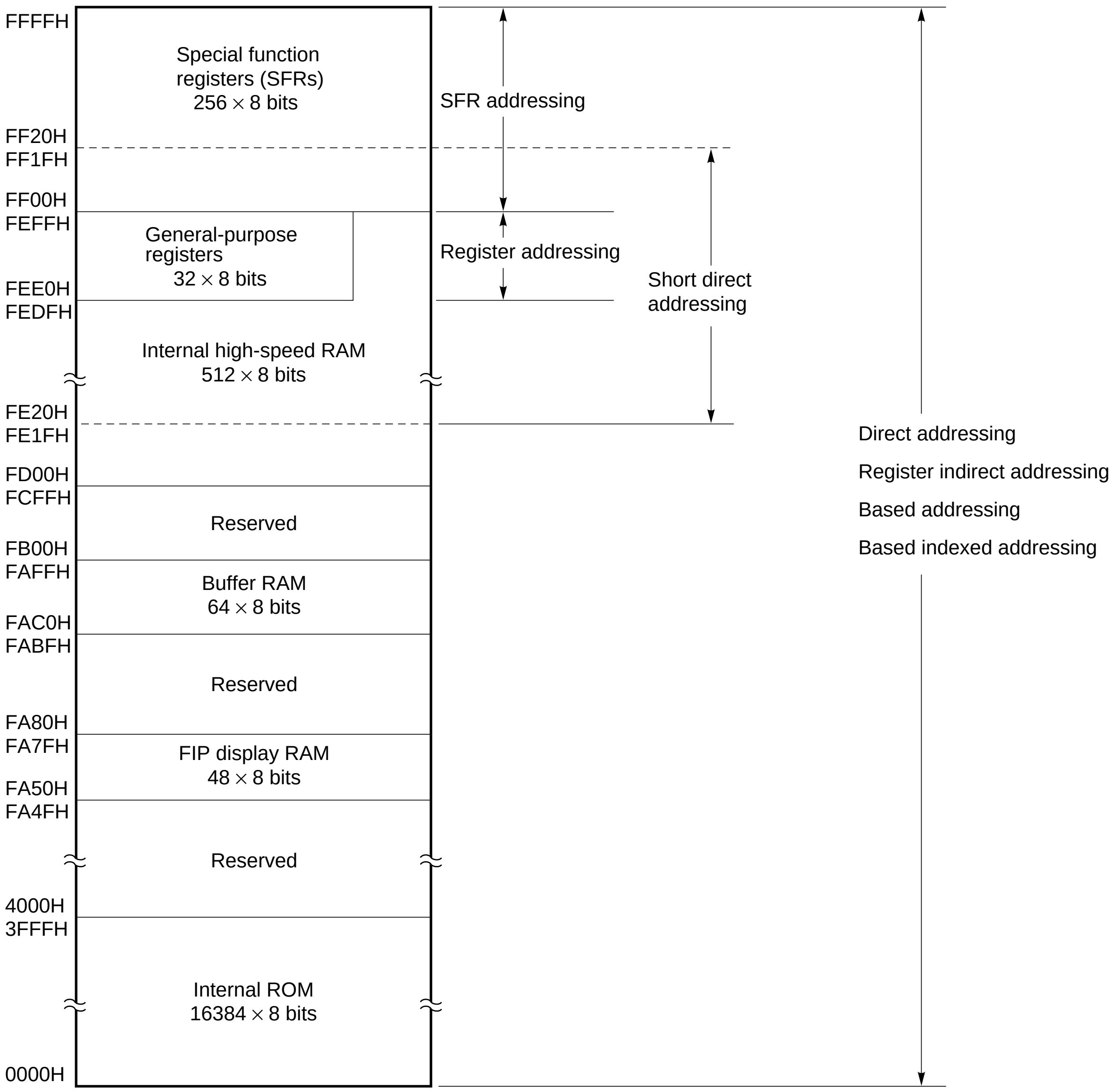


Figure 3-7. Data Memory Addressing (μPD78043F)

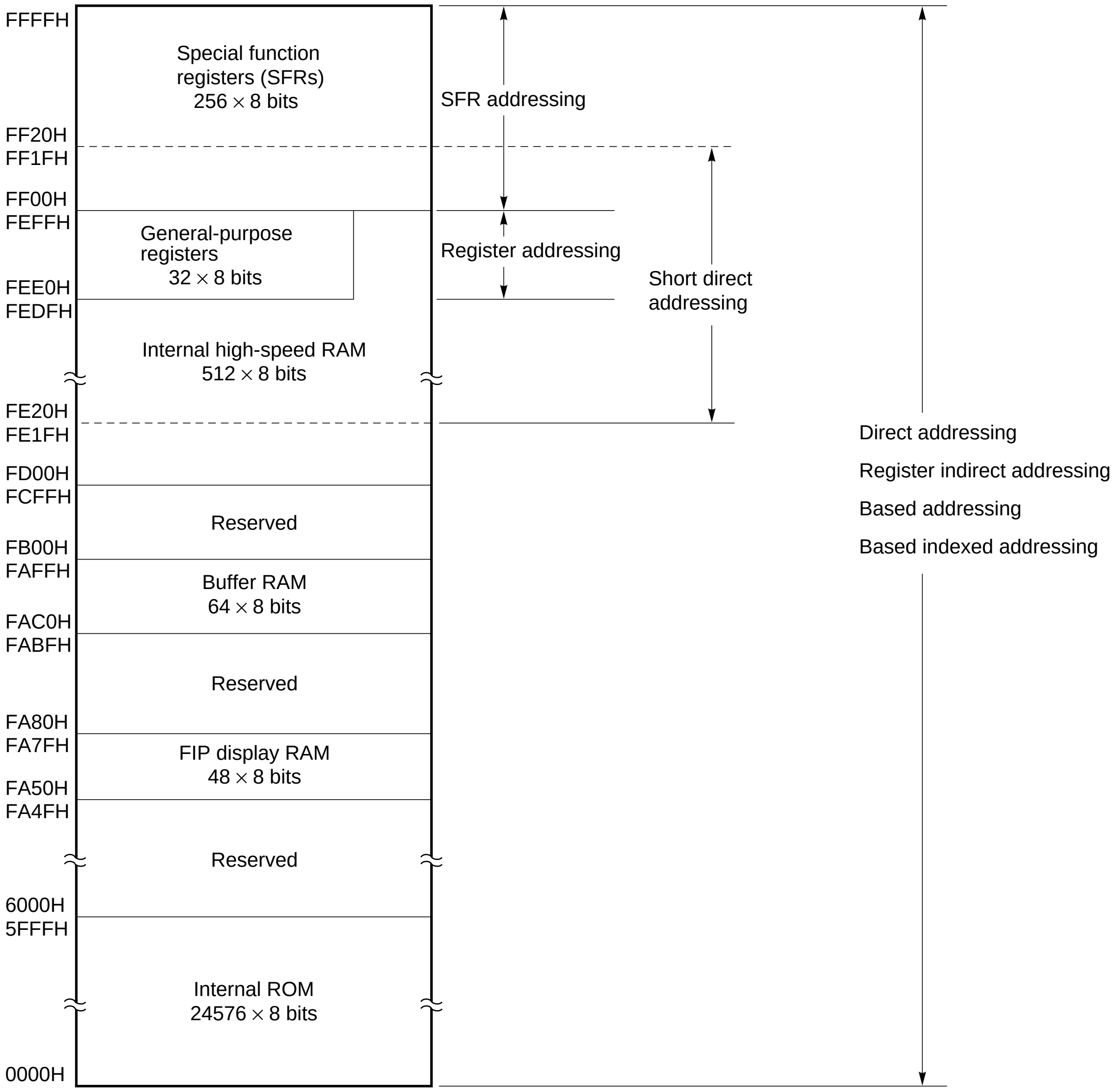


Figure 3-8. Data Memory Addressing (μPD78044F)

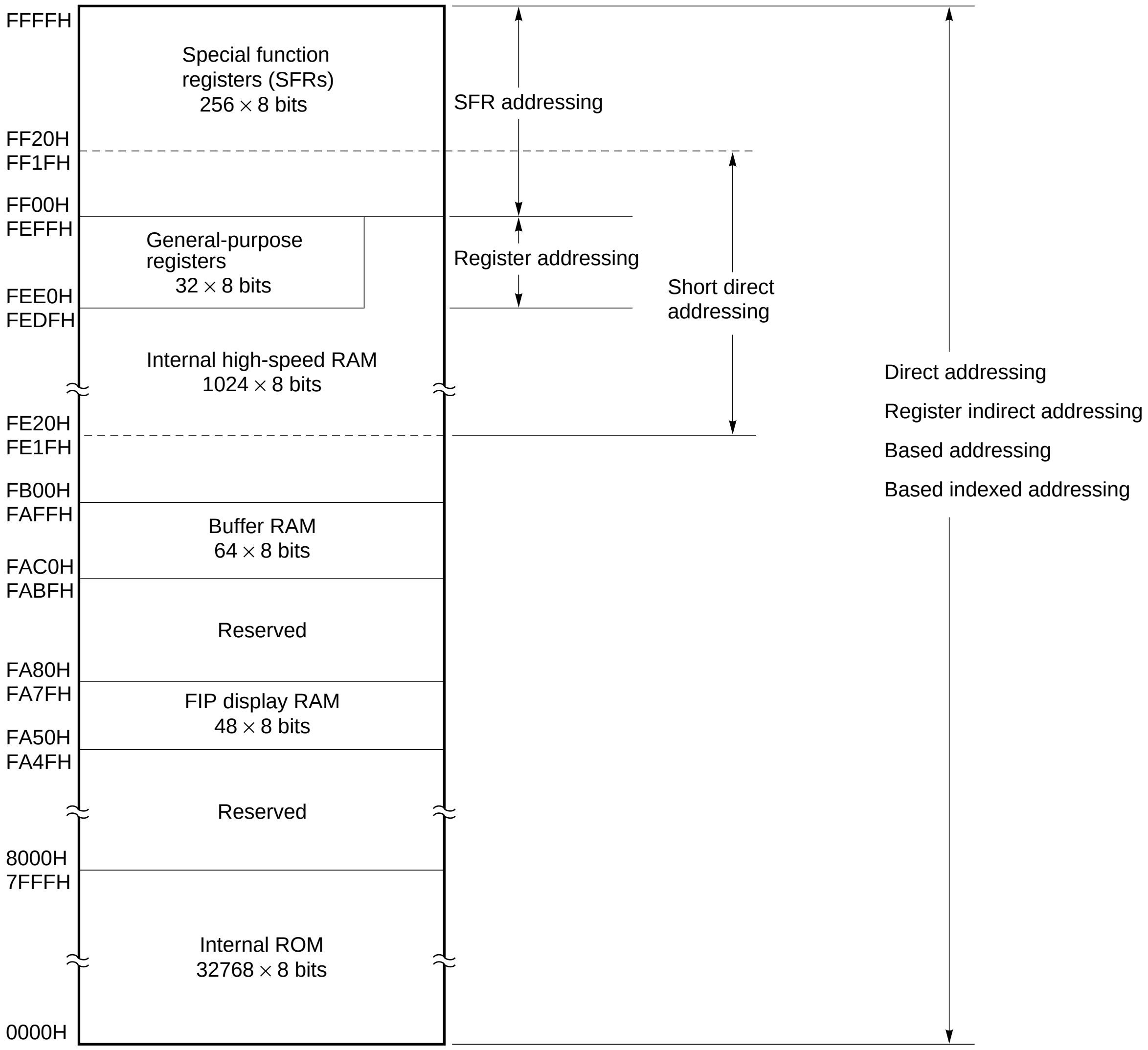


Figure 3-9. Data Memory Addressing (μPD78045F)

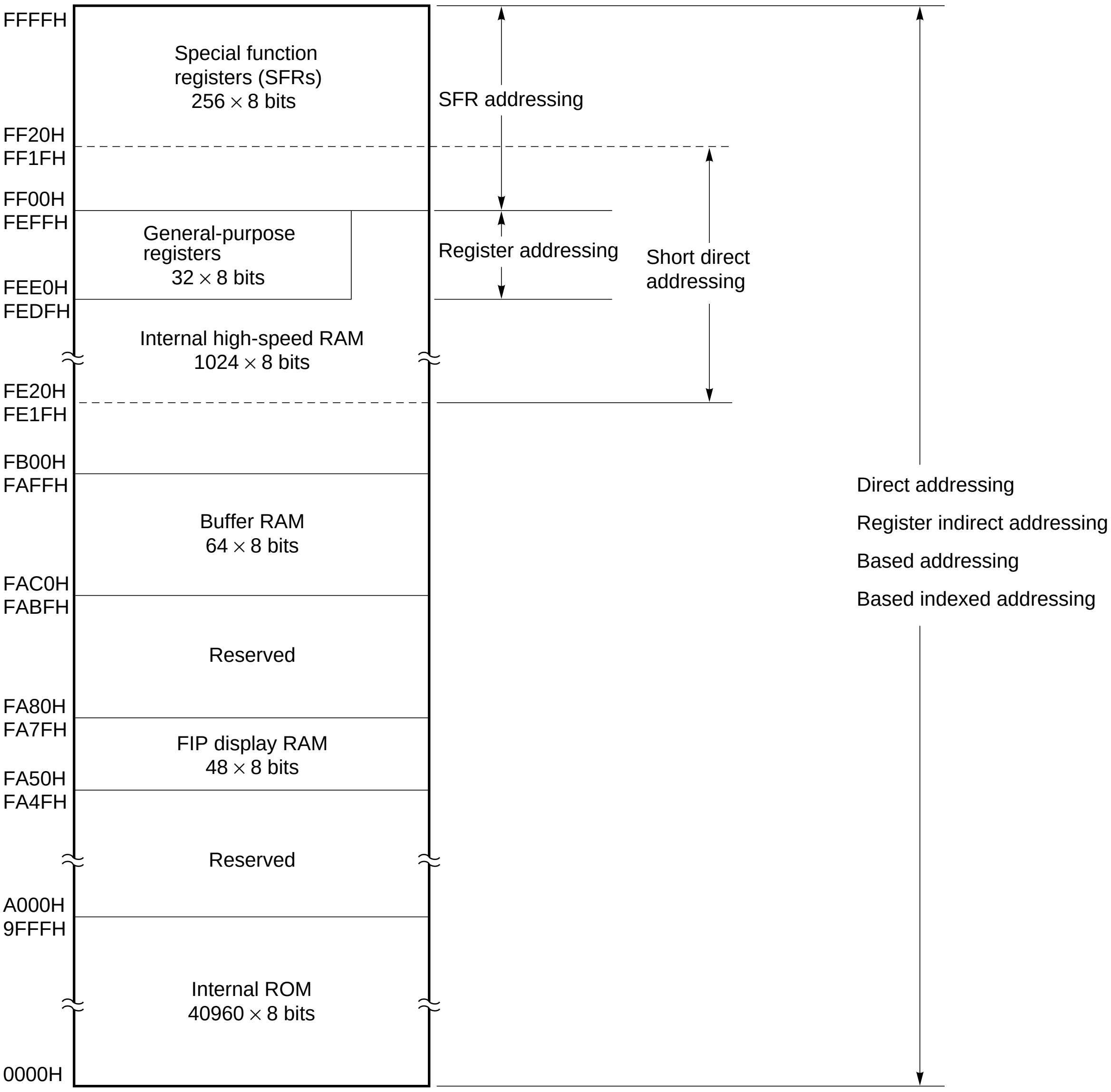
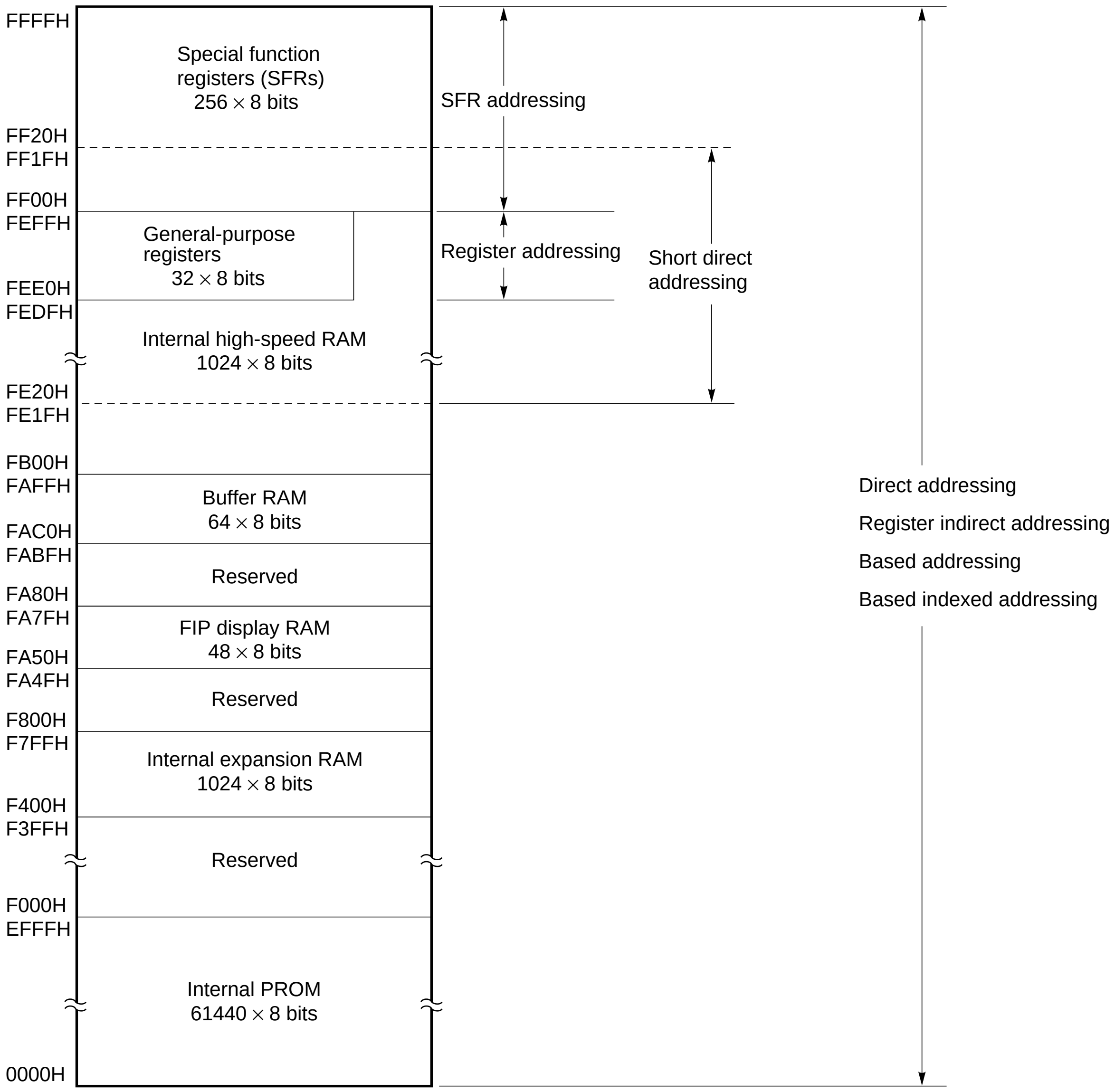


Figure 3-10. Data Memory Addressing (μPD78P048A)



3.2 Processor Registers

The μPD78044F Subseries units incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 16-bit register which holds the address information of the next program to be executed. In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

$\overline{\text{RESET}}$ input sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-11. 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 various flags to be set/reset by instruction execution. Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are automatically reset upon execution of the RETB, RETI and POP PSW instructions.

$\overline{\text{RESET}}$ input sets the PSW to 02H.

Figure 3-12. Program Status Word Configuration

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledge operations of CPU.

All interrupts, except the non-maskable interrupt, are disabled (DI status) when IE = 0.

The interrupts are enabled when IE = 1. At this time, acknowledging an interrupt request is controlled with an inservice priority flag (ISP), an interrupt mask flag for various interrupt sources and a priority specification flag.

The IE is reset (to 0) upon DI instruction execution or interrupt request acknowledgment and is set (to 1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation result is zero, this flag is set (to 1). It is reset (to 0) in all other cases.

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

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

In these flags, the 2-bit information which indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (to 1). It is reset (to 0) in all other cases.

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts.

When ISP = 0, the vector interrupt assigned a low priority by the priority specify flag registers (PR0L and PR0H) (refer to **17.3 (3) Priority specify flag registers (PR0L and PR0H)**) is disabled and cannot be acknowledged. Whether an interrupt request is actually acknowledged is controlled with the interrupt enable flag (IE).

(f) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit manipulation instruction execution.

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal high-speed RAM area can be set as the stack area.

Each product of internal high-speed RAM is as follows.

μPD78042F, 78043F: FD00H to FEFFH

μPD78044F, 78045F, 78P048A: FB00H to FEFFH

Figure 3-13. Stack Pointer Configuration

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

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

Each stack operation saves/resets data as shown in Figures 3-14 and 3-15.

Caution Since $\overline{\text{RESET}}$ input makes SP contents indeterminate, be sure to initialize the SP before instruction execution.

Figure 3-14. Data to be Saved to Stack Memory

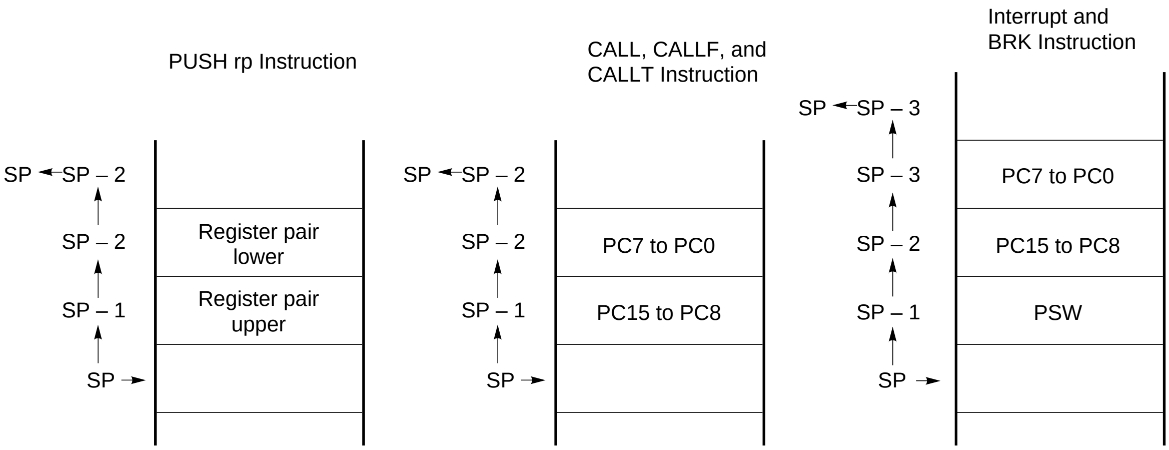
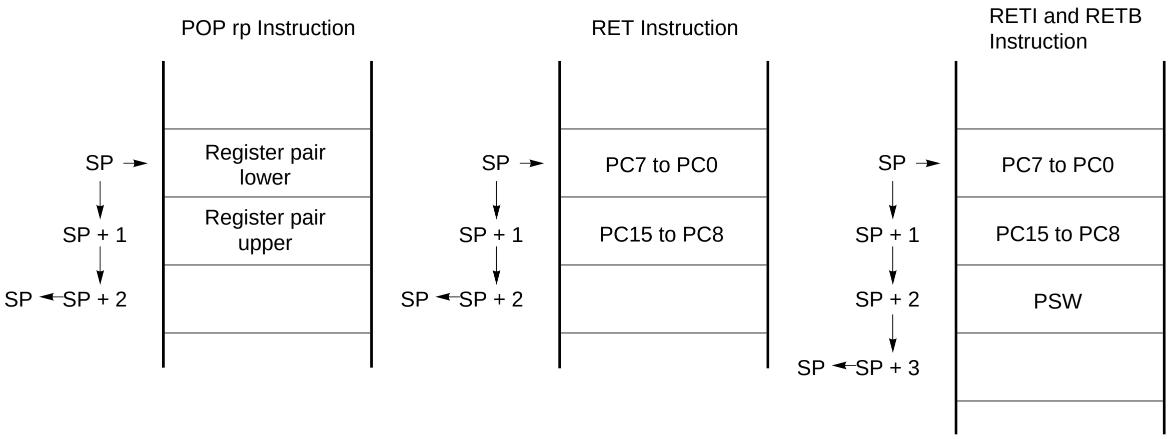


Figure 3-15. Data to be Reset from Stack Memory



3.2.2 General registers

A general register is mapped at particular addresses (FEE0H to FEFFH) of the data memory. It consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L and H).

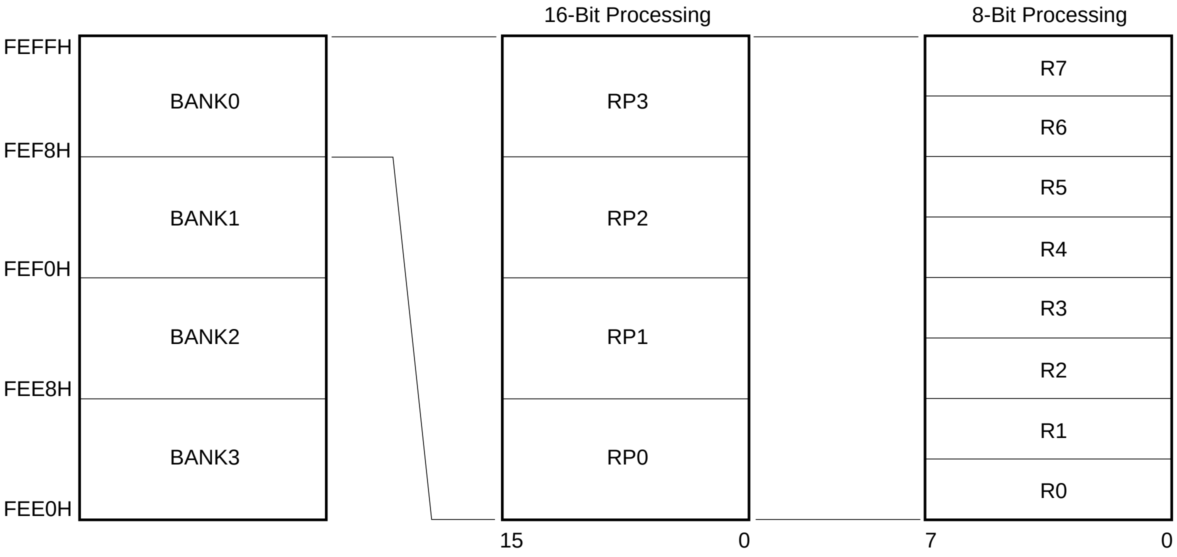
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE and HL).

They can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE and HL) and absolute names (R0 to R7 and RP0 to RP3).

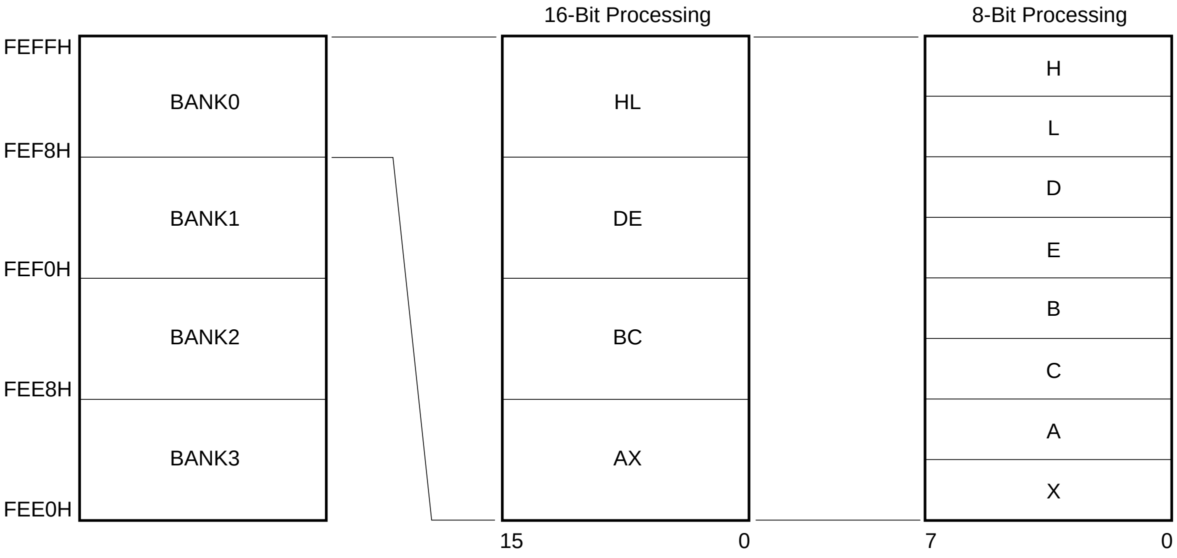
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interruption for each bank.

Figure 3-16. General Register Configuration

(a) Absolute Name



(b) Function Name



3.2.3 Special function register (SFR)

Unlike a general register, each special function register has special functions. It is allocated in the FF00H to FFFFH area.

The special function register can be manipulated, like the general register, with the operation, transfer and bit manipulation instructions. Manipulatable bit units, 1, 8 and 16, depend on the special function register type.

Each manipulation bit unit can be specified as follows:

- **1-bit manipulation**
  Describe the symbol reserved with assembler for the 1-bit manipulation instruction operand (sfr.bit).
  This manipulation can also be specified with an address.

- **8-bit manipulation**
  Describe the symbol reserved with assembler for the 8-bit manipulation instruction operand (sfr).
  This manipulation can also be specified with an address.

- **16-bit manipulation**
  Describe the symbol reserved with assembler for the 16-bit manipulation instruction operand (sfrp).
  When addressing an address, describe an even address.

Table 3-4 gives a list of special function registers. The meaning of items in the table is as follows.

- **Symbol**
  Indicates symbols that specify the addresses of the special function registers. RA78K/0 uses these symbols as the reserved words, and CC78K/0 has defined them in the header file named "sfrbit.h". Symbols can be used as instruction operands if RA78K/0, ID78K0, or CC78K/0 is used.

- **R/W**
  Indicates whether the corresponding special function register can be read or written.

R/W	:	Read/write enable
R	:	Read only
W	:	Write only

- **Manipulatable bit units**
  "√" indicates the manipulatable bit units, 1, 8, and 16.
  "–" indicates a bit unit for which manipulation is not possible.

- **After reset**
  Indicates each register status upon $\overline{\text{RESET}}$ input.

Table 3-4. Special Function Register List (1/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0	R/W	√	√	–	00H
FF01H	Port 1	P1		√	√	–	
FF02H	Port 2	P2		√	√	–	
FF03H	Port 3	P3		√	√	–	
FF07H	Port 7	P7		√	√	–	
FF08H	Port 8	P8	W	√	√	–	
FF09H	Port 9	P9		√	√	–	
FF0AH	Port 10	P10		√	√	–	
FF0BH	Port 11	P11	R/W	√	√	–	
FF0CH	Port 12	P12		√	√	–	
FF10H FF11H	16-bit compare register	CR00		–	–	√	Undefined
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		√	√	–	
FF27H	Port mode register 7	PM7		√	√	–	1FH
FF2BH	Port mode register 11	PM11		√	√	–	FFH
FF2CH	Port mode register 12	PM12		√	√	–	

Table 3-4. Special Function Register List (2/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF40H	Timer clock select register 0	TCL0		R/W	√	√	–	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			√	√	–	
FF4FH	8-bit timer output control register	TOC1			√	√	–	
FF60H	Serial operating mode register 0	CSIM0			√	√	–	
FF61H	Serial bus interface control register	SBIC			√	√	–	
FF62H	Slave address register	SVA			–	√	–	Undefined
FF63H	Interrupt timing specify register	SINT			√	√	–	00H
FF68H	Serial operating mode register 1	CSIM1			√	√	–	
FF69H	Automatic data transmit/receive control register	ADTC			√	√	–	
FF6AH	Automatic data transmit/receive address pointer	ADTP			–	√	–	
FF6BH	Automatic data transmit/receive interval specify register	ADTI						
FF80H	A/D converter mode register	ADM			√	√	–	01H
FF84H	A/D converter input select register	ADIS			–	√	–	00H
FFA0H	Display mode register 0	DSPM0			△Note	√	–	
FFA1H	Display mode register 1	DSPM1			–	√	–	
FFA8H	6-bit up/down counter mode register	UDM			√	√	–	
FFA9H	6-bit up/down counter	UDC			–	√	–	
FFAAH	6-bit up/down counter compare register	UDCC			–	√	–	
FFE0H	Interrupt request flag register 0L	IF0	IF0L		√	√	√	
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 order specify flag register 0L	PR0	PR0L		√	√	√	
FFE9H	Priority order specify flag register 0H		PR0H		√	√		
FFECH	External interrupt mode register	INTM0			–	√	–	00H

Note Only bit 7 enables for read-only operation.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FFF0H	Memory size switching register	IMS	R/W	–	√	–	**Note**
FFF4H	Internal expansion RAM size switching register	IXS	W	–	√	–	**Note**
FFF7H	Pull-up resistor option register	PUO	R/W	√	√	–	00H
FFF9H	Watchdog timer mode register	WDTM		√	√	–	
FFFAH	Oscillation stabilization time select register	OSTS		–	√	–	04H
FFFBH	Processor clock control register	PCC		√	√	–	

Note The value when reset memory size switching register (IMS) and internal expansion RAM size register (IXS) depends on products.

If using the mask ROM version, do not set any value other than that when reset to IMS and IXS.

	μPD78042F	μPD78043F	μPD78044F	μPD78045F	μPD78P048A
IMS	44H	46H	C8H	CAH	CFH
IXS	None				0AH

3.3 Instruction Address Addressing

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

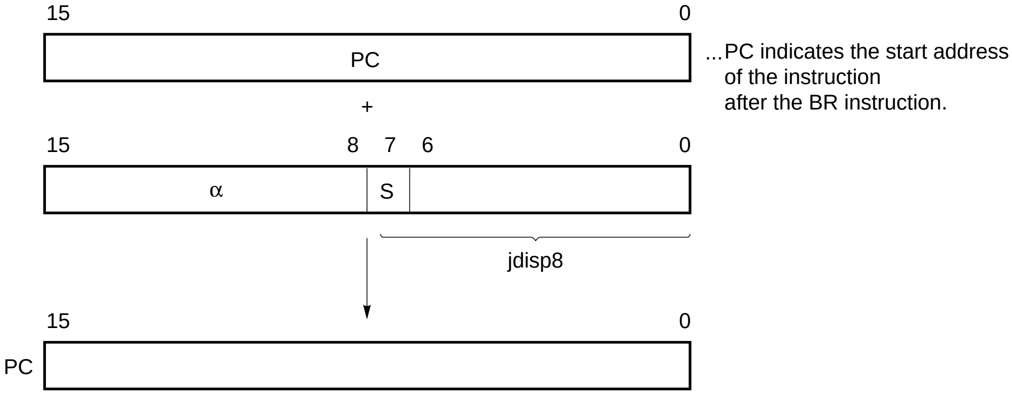
3.3.1 Relative addressing

[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (–128 to +127) and bit 7 becomes a sign bit. In other words, execution branches in a relative range of –128 to +127 from the start address of the subsequent instruction in the relative addressing mode.

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

[Illustration]


15
0
PC
...PC indicates the start address of the instruction after the 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.

3.3.2 Immediate addressing

[Function]

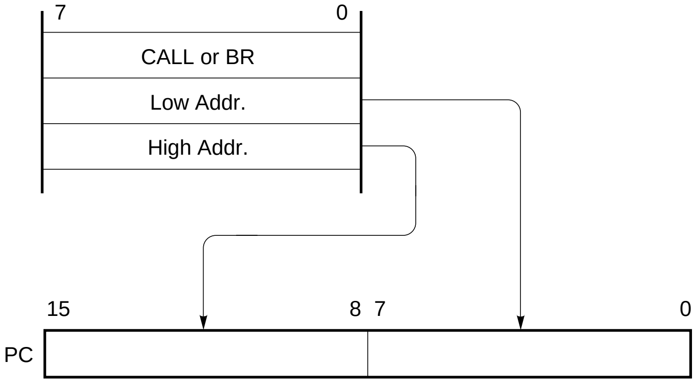
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

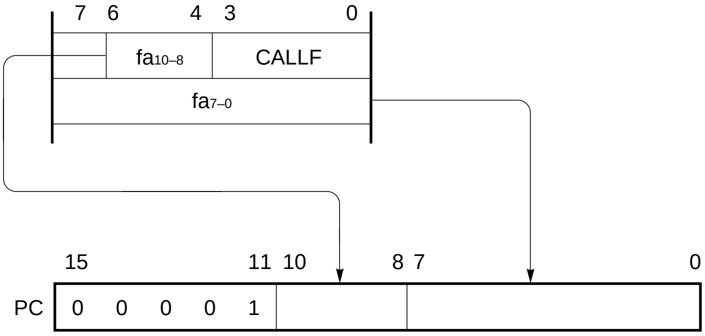
The CALL !addr16 and BR !addr16 instructions can branch in the entire memory space. The CALL !addr11 instruction branches in an area of 0800H to 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



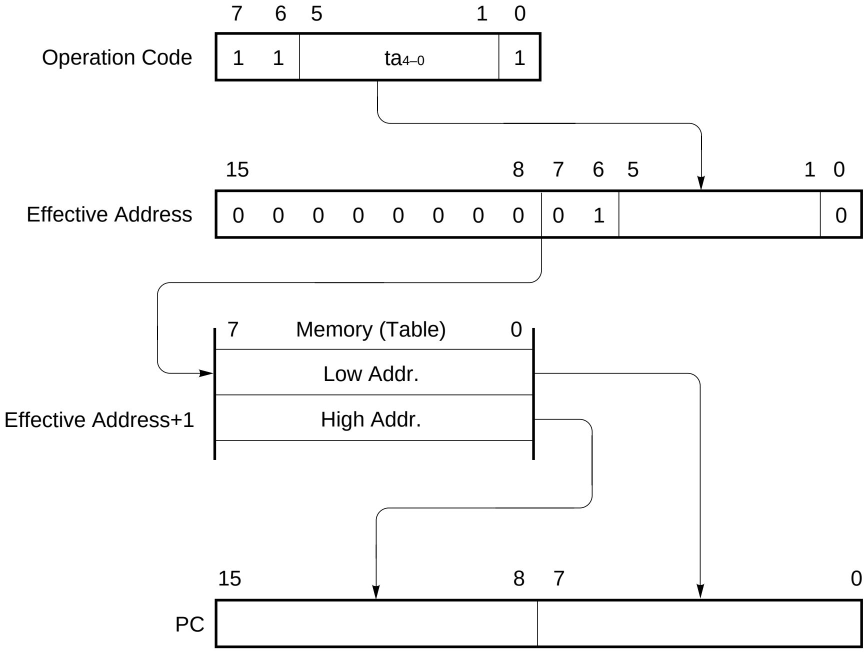
3.3.3 Table indirect addressing

[Function]

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

Table indirect addressing is performed when the CALLT [addr5] instruction is executed. This instruction can reference addresses stored to the memory table of 40H to 7FH and can branch in the entire memory space.

[Illustration]



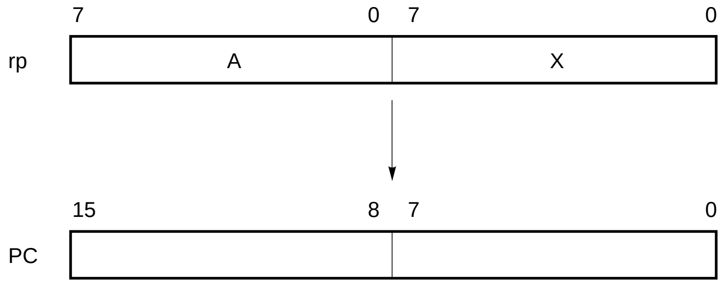
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Implied addressing

[Function]

The register which functions as an accumulator (A and AX) in the general register is automatically (implicitly) addressed.

Of the μPD78044F Subseries instruction words, the following instructions employ implied addressing.

Instruction	Register to be Specified by Implied Addressing
MULU	A register for multiplicand and AX register for product storage.
DIVUW	AX register for dividend and quotient storage.
ADJBA/ADJBS	A register for storage of numeric values subject to decimal adjustment.
ROR4/ROL4	A register for storage of digit data which undergoes digit rotation.

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

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

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

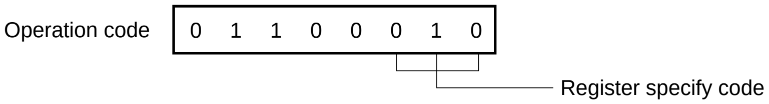
[Operand format]

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

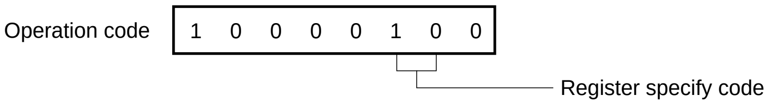
'r' and 'rp' can be described with function names (X, A, C, B, E, D, L, H, AX, BC, DE and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

This addressing is to directly address a memory area indicated by the immediate data in the instruction word.

[Operand format]

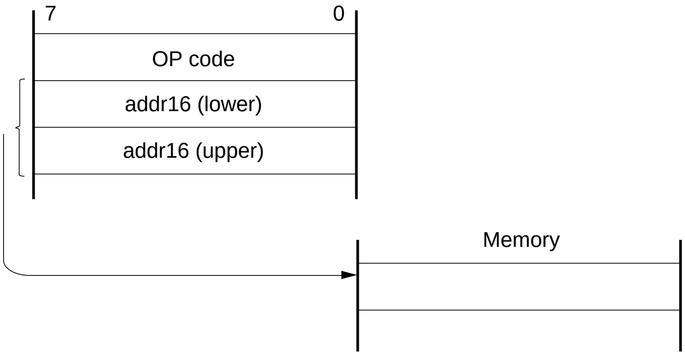
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

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

This addressing is applied to the 256-byte space FE20H to FF1FH for a fixed space. An internal high-speed RAM and a special function register (SFR) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the entire SFR area. To this area, ports which are frequently accessed in a program and a compare register of the timer/event counter and a capture register of the timer/event counter are mapped and these SFRs can be manipulated with a small number of bytes and clocks.

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

[Operand format]

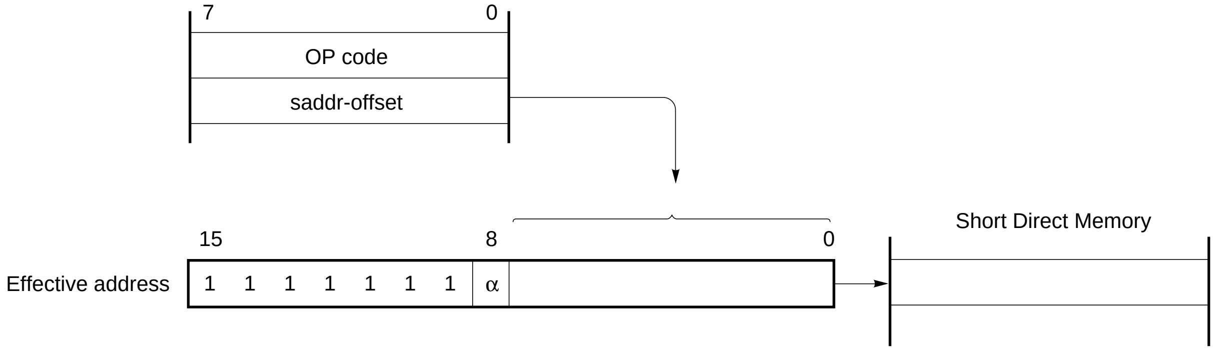
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[Description example]

MOV 0FE30H, #50H; when setting saddr to FE30H and immediate data to 50H

Operation 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)

[Illustration]



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

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

3.4.5 Special function register (SFR) addressing

[Function]

The memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word.

This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFR mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

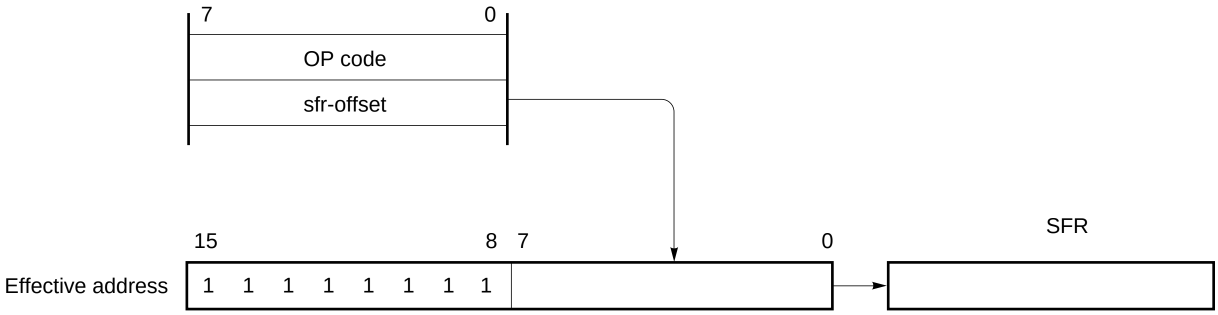
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

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

[Illustration]



3.4.6 Register indirect addressing

[Function]

This addressing is to address a memory area with the contents of a register pair specified as an operand. The register pair to be accessed is specified by the register bank select flag (RBS0 and RBS1) and the register pair specify code in an instruction code. This addressing can be carried out for all the memory spaces.

[Operand format]

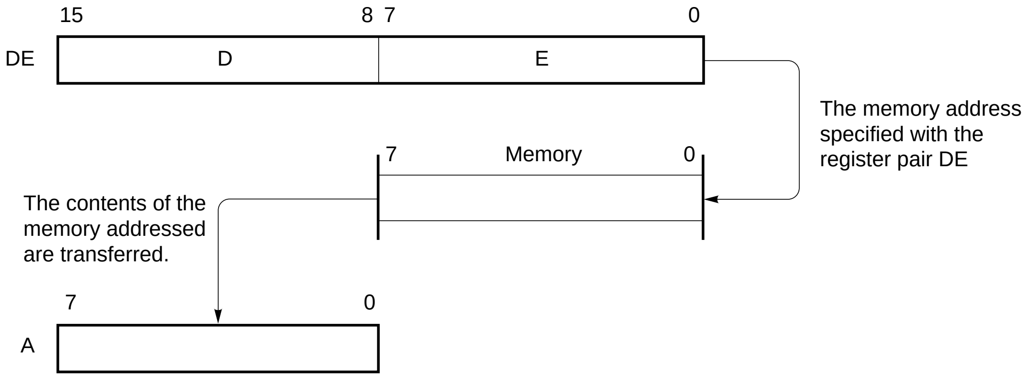
Identifier	Description
–	[DE], [HL]

[Description example]

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

Operation code	1	0	0	0	0	1	0	1

[Illustration]


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

3.4.7 Based addressing

[Function]

This addressing is to address a memory area by using the result of adding 8-bit immediate data to the contents of the HL register pair which is used 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). Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

Identifier	Description
—	[HL+byte]

[Description example]

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

Operation code	1	0	1	0	1	1	1	0
	0	0	0	1	0	0	0	0

3.4.8 Based indexed addressing

[Function]

This addressing is to address a memory area by using the result of adding the contents of the B or C register specified in the instruction word to the contents of the HL register pair which serves as a base register. The HL and B or C register to be accessed are in the register bank specified by the register bank select flags (RBS0 and RBS1). The addition is performed by expanding the contents of the B or C register as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

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

[Description example]

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

Operation code	1	0	1	0	1	0	1	1

3.4.9 Stack addressing

[Function]

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

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

Stack addressing enables addressing the internal high-speed RAM area only.

[Description example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD78044F Subseries units incorporate two input ports, 18 output ports and 48 input/output ports. Figure 4-1 shows the port configuration. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as built-in hardware input/output pins.

Figure 4-1. Port Types

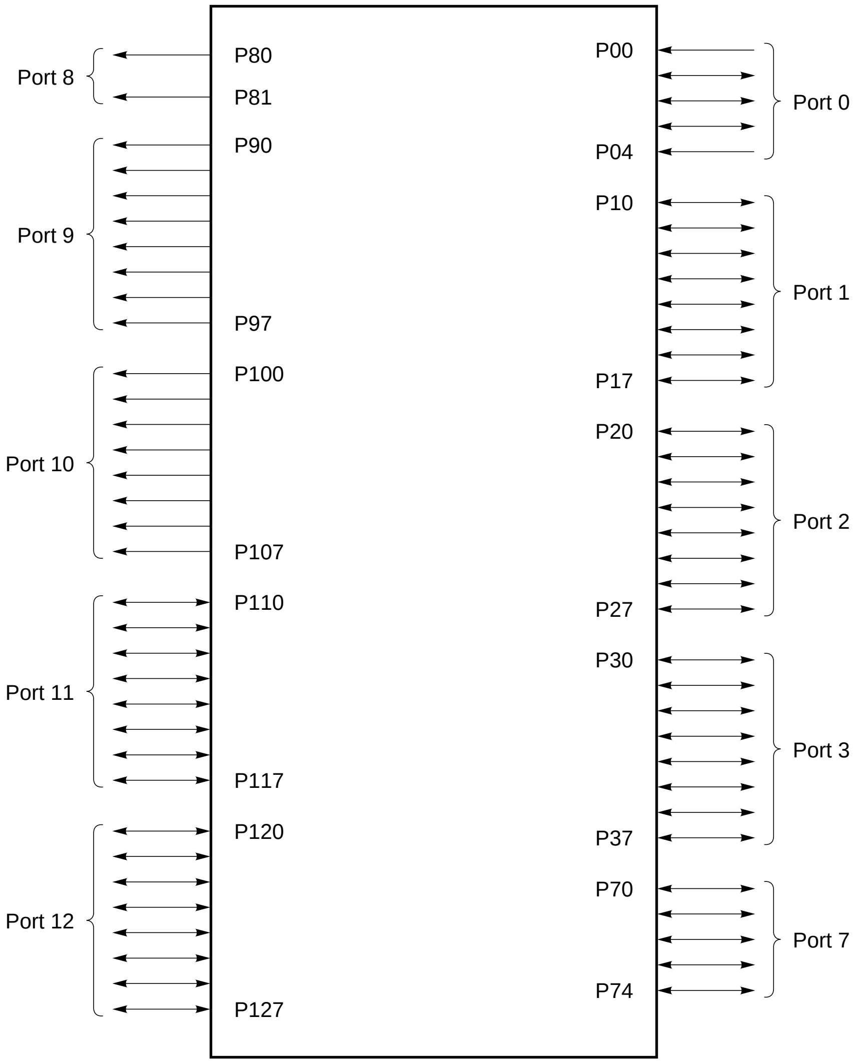


Table 4-1. Port Functions (1/2)

Pin Name	Function		Alternate Function Pin
P00	Port 0. 5-bit input/output port.	Input only.	INTP0/TI0
P01		Input/output specifiable bit-wise. If used as an input port, the on-chip pull-up resistors can be used.	INTP1
P02			INTP2
P03			INTP3/CI0
P04		Input only.	XT1
P10 to P17	Port 1. 8-bit input/output port. Input/output specifiable bit-wise. If used as an input port, the on-chip pull-up resistors can be used.		ANI0 to ANI7
P20	Port 2. 8-bit input/output port. Input/output specifiable bit-wise. If used as an input port, the on-chip pull-up resistors can be used.		SI1
P21			SO1
P22			$\overline{\text{SCK1}}$
P23			STB
P24			BUSY
P25			SI0/SB0
P26			SO0/SB1
P27			$\overline{\text{SCK0}}$
P30	Port 3. 8-bit input/output port. Input/output specifiable bit-wise. LED can be driven directly. If used as an input port, the on-chip pull-up resistors can be used. Only mask ROM version can specify pull-down registers bit-wise with the mask option.		TO0
P31			TO1
P32			TO2
P33			TI1
P34			TI2
P35			PCL
P36			BUZ
P37			–
P70 to P74	Port 7. N-ch open-drain 5-bit input/output port. Input/output specifiable bit-wise. LED can be driven directly. Only mask ROM version can specify a pull-up resistor bit-wise with the mask option.		–

Table 4-1. Port Functions (2/2)

Pin Name	Function	Alternate Function Pin
P80, P81	Port 8. P-ch open-drain 2 bits high breakdown voltage output ports. LEDs can be driven directly. Mask ROM version can specify pull-down resistors bit-wise with the mask option (can be specified to V_{LOAD} or V_{SS} in 2-bit units). μPD78P048A contains a pull-down resistor (connected to V_{LOAD}).	FIP0, FIP1
P90 to P97	Port 9. P-ch open-drain 8-bit high breakdown voltage output ports. LEDs can be driven directly. Mask ROM version can specify pull-down resistors bit-wise with the mask option (can be specified to V_{LOAD} or V_{SS} in 4-bit units). μPD78P048A contains a pull-down resistor (connected to V_{LOAD}).	FIP2 to FIP9
P100 to P107	Port 10. P-ch open-drain 8-bit high breakdown voltage output ports. P100 to P105 can drive LEDs directly. Mask ROM version can specify pull-down resistors bit-wise with the mask option (can be specified to V_{LOAD} or V_{SS} in 4-bit units). μPD78P048A contains pull-down resistors in P100 to P105 (connected to V_{LOAD}).	FIP10 to FIP17
P110 to P117	Port 11. P-ch open-drain 8-bit high breakdown voltage input/output ports. Can specify input/output bit-wise. P110 to P117 can drive LEDs directly. Only mask ROM version can specify pull-down resistors bit-wise with the mask option (can be specified to V_{LOAD} or V_{SS} in 4-bit units).	FIP18 to FIP25
P120 to P127	Port 12. P-ch open-drain 8-bit high breakdown voltage input/output ports. Can specify input/output bit-wise. P120 to P127 can drive LEDs directly. Mask ROM version can specify pull-down resistors bit-wise with the mask option (can be specified to V_{LOAD} or V_{SS} in 4-bit units).	FIP26 to FIP33

4.2 Port Configuration

A port consists of the following hardware.

Table 4-2. Port Configuration

Item	Configuration
Control register	Port mode register (PMm: m = 0, 1, 2, 3, 7, 11, 12) Pull-up resistor option register (PUO)
Port	Total: 68 ports (2 inputs, 18 outputs, 48 inputs/outputs)
Pull-up resistor	• Mask ROM versions Total: 32 (software specifiable: 27, mask option: 5) • μPD78P048A ... Total: 27
Pull-down resistor	• Mask ROM version ... Total: 42 (mask option: 42) • μPD78P048A ... Total: 16

4.2.1 Port 0

Port 0 is a 5-bit input/output port with output latch. P01 to P03 pins can specify the input mode/output mode in 1-bit units with the port mode register 0 (PM0). P00 and P04 pins are input-only ports. When P01 to P03 pins are used as input ports, on-chip pull-up resistors can be specified for them in 3-bit units with the pull-up resistor option register (PUO).

Alternate functions include external interrupt request input, external count clock input to the timer and crystal connection for subsystem clock oscillation.

$\overline{\text{RESET}}$ input sets port 0 to input mode.

Figures 4-2 and 4-3 show block diagrams of port 0.

Caution Because port 0 also serves for external interrupt request input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. Thus, when the output mode is used, set the interrupt mask flag to 1.

Figure 4-2. P00 and P04 Block Diagram

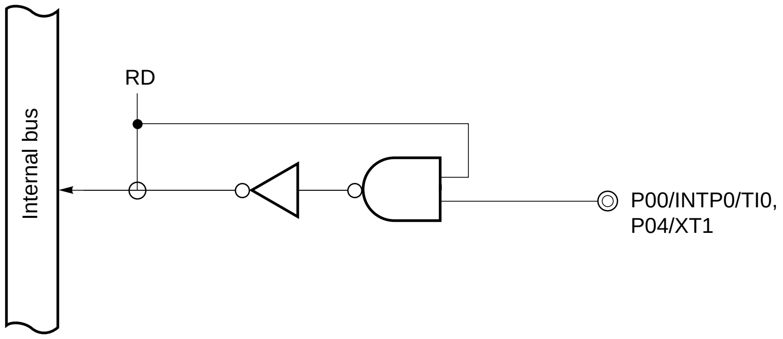
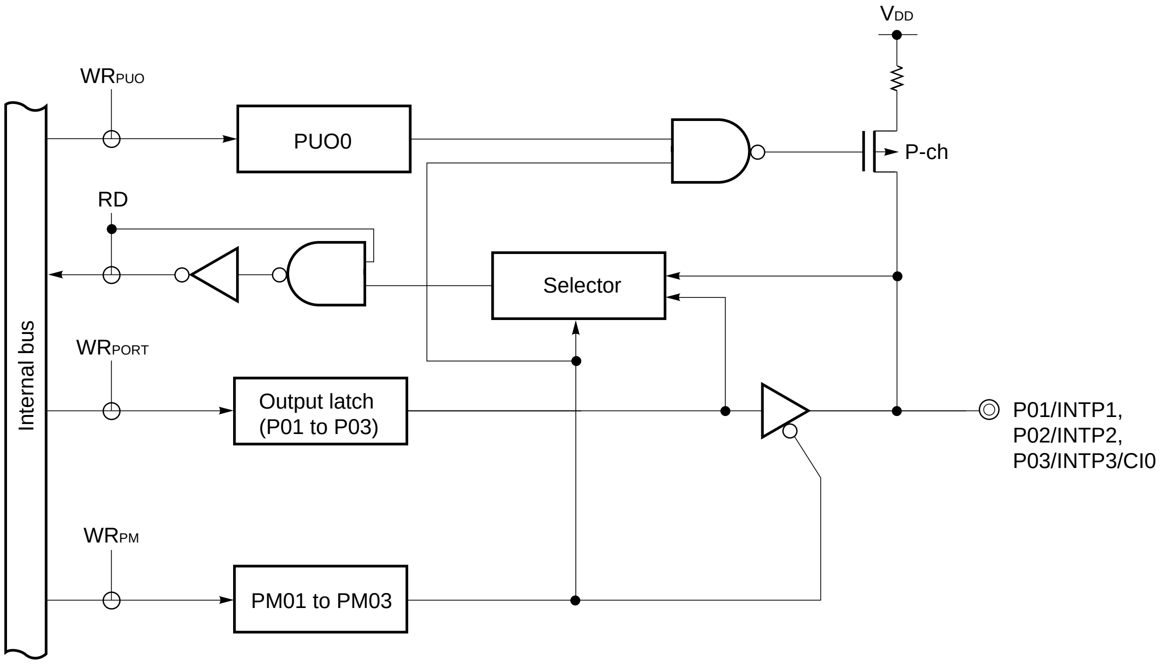


Figure 4-3. P01 to P03 Block Diagram



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 0 read signal

WR : Port 0 write signal

4.2.2 Port 1

Port 1 is an 8-bit input/output port with output latch. P10 to P17 pins can specify the input mode/output mode in 1-bit units with a port mode register 1 (PM1). When P10 to P17 pins are used as input ports, on-chip pull-up resistors can be specified for them in 8-bit units with the pull-up resistor option register (PUO).

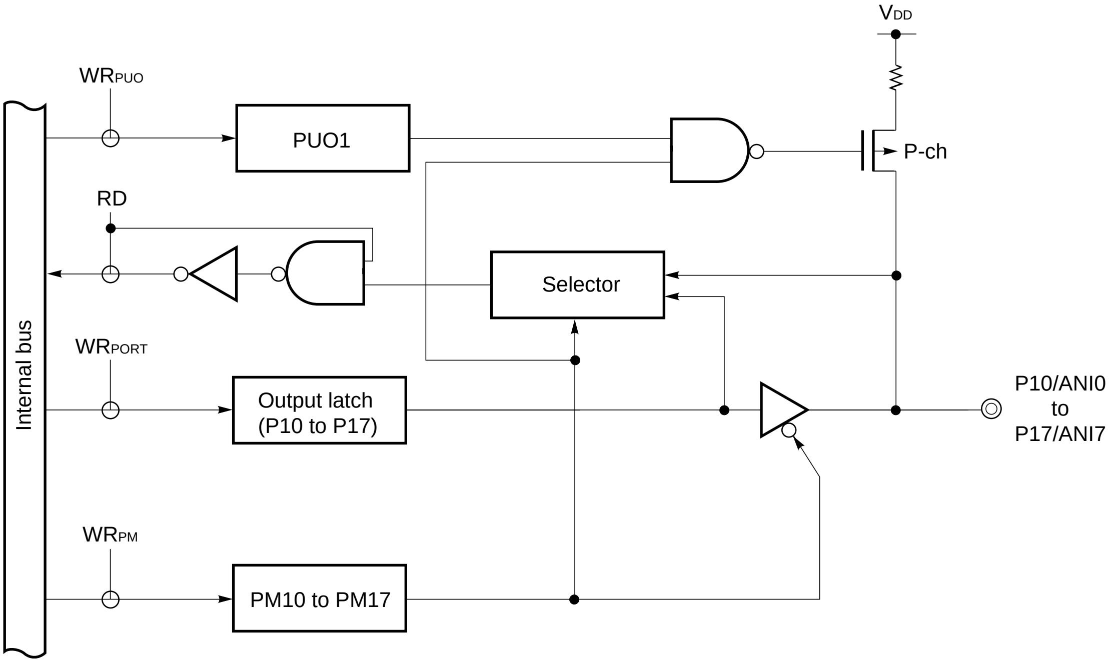
Alternate functions include an A/D converter analog input.

$\overline{\text{RESET}}$ input sets port 1 to input mode.

Figure 4-4 shows a block diagram of port 1.

Caution A pull-up resistor cannot be used for pins used as A/D converter analog input.

Figure 4-4. P10 to P17 Block Diagram



PUO : Pull-up resistor option register
PM : Port mode register
RD : Port 1 read signal
WR : Port 1 write signal

4.2.3 Port 2

Port 2 is an 8-bit input/output port with output latch. P20 to P27 pins can specify the input mode/output mode in 1-bit units with the port mode register 2 (PM2). When P20 to P27 pins are used as input ports, on-chip pull-up resistors can be specified for them in 8-bit units with the pull-up resistor option register (PUO).

Alternate functions include serial interface data input/output, clock input/output, automatic transmit/receive busy input, and strobe output.

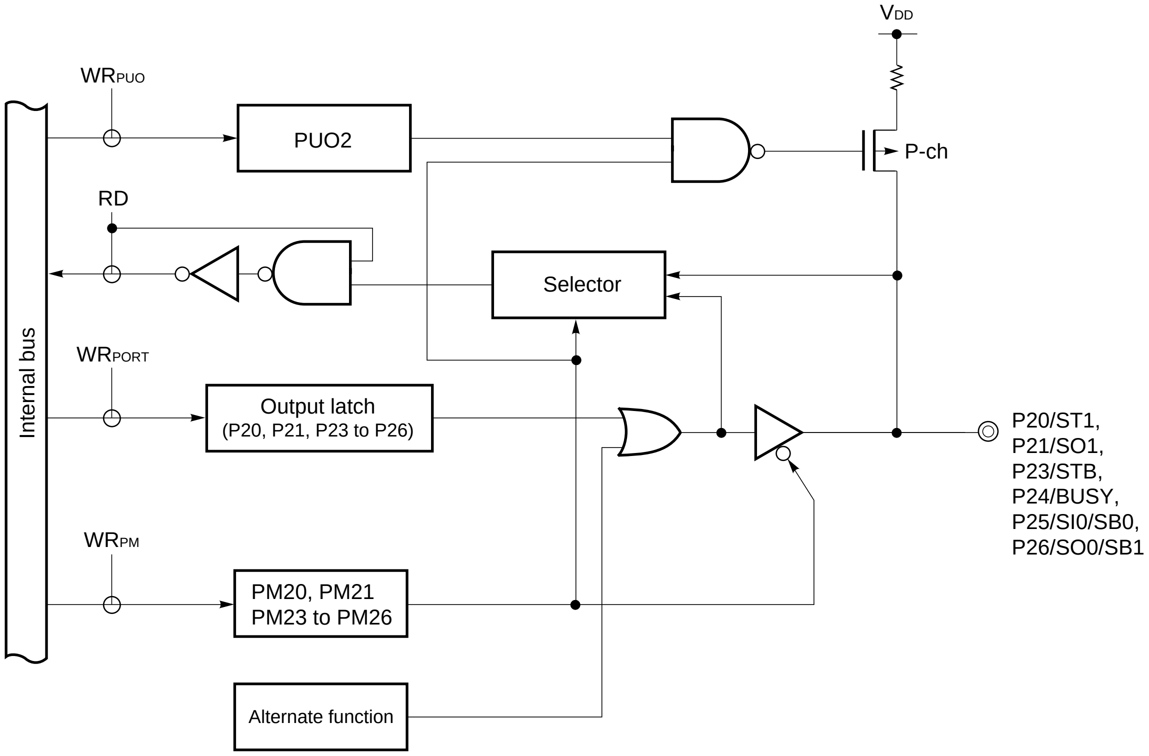
$\overline{\text{RESET}}$ input sets port 2 to input mode.

Figures 4-5 and 4-6 show a block diagram of port 2.

Cautions 1. If used as serial interface, set the input/output latch according to each function. Refer to Figure 14-3 Serial Operating Mode Register 0 Format and Figure 15-3 Serial Operating Mode Register 1 Format for setting.

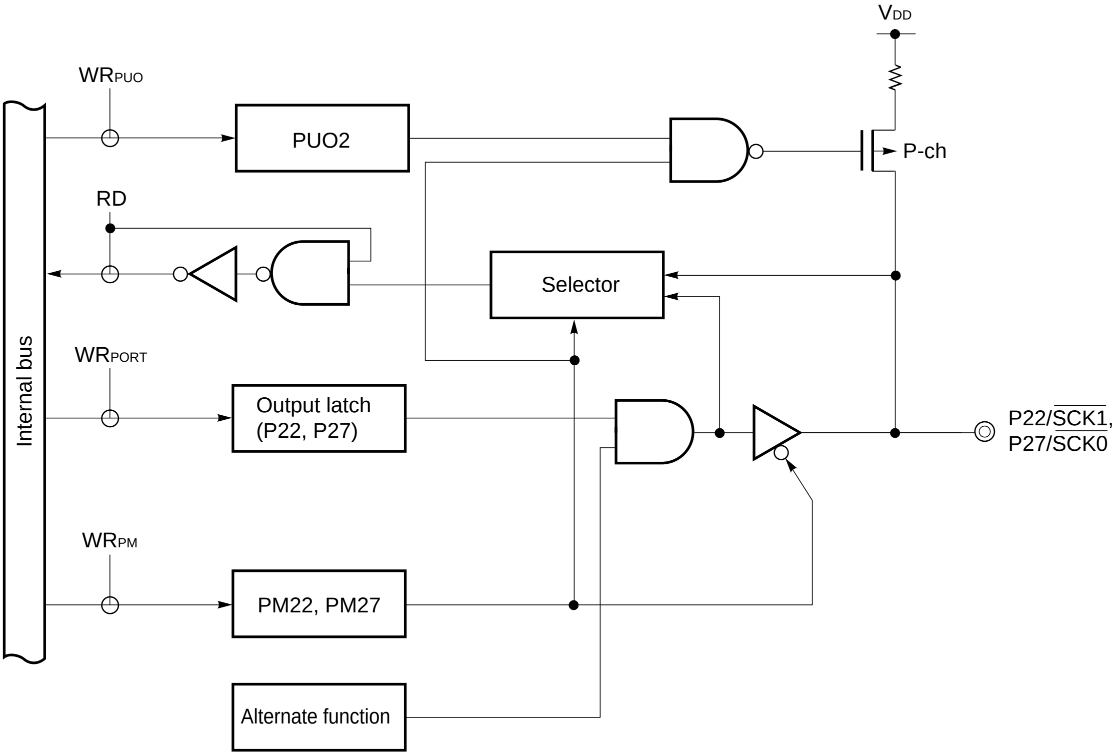
2. To read the status of a pin in the SBI mode, set the PM2n bit of PM2 to 1 (n = 5 or 6) (refer to 14.4.3 (10) Judging busy status of slave).

Figure 4-5. P20, P21, P23 to P26 Block Diagram



PUO : Pull-up resistor option register
PM : Port mode register
RD : Port 2 read signal
WR : Port 2 write signal

Figure 4-6. P22 and P27 Block Diagram



PUO : Pull-up resistor option register
PM : Port mode register
RD : Port 2 read signal
WR : Port 2 write signal

4.2.4 Port 3

Port 3 is an 8-bit input/output port with output latch. P30 to P37 pins can specify the input mode/output mode in 1-bit units with the port mode register 3 (PM3). When P30 to P37 pins are used as input ports, on-chip pull-up resistors can be specified for them in 8-bit units with the pull-up resistor option register (PUO).

Mask ROM version can contain pull-down resistors bit-wise with the mask option. The μPD78P048A does not contain pull-down resistors.

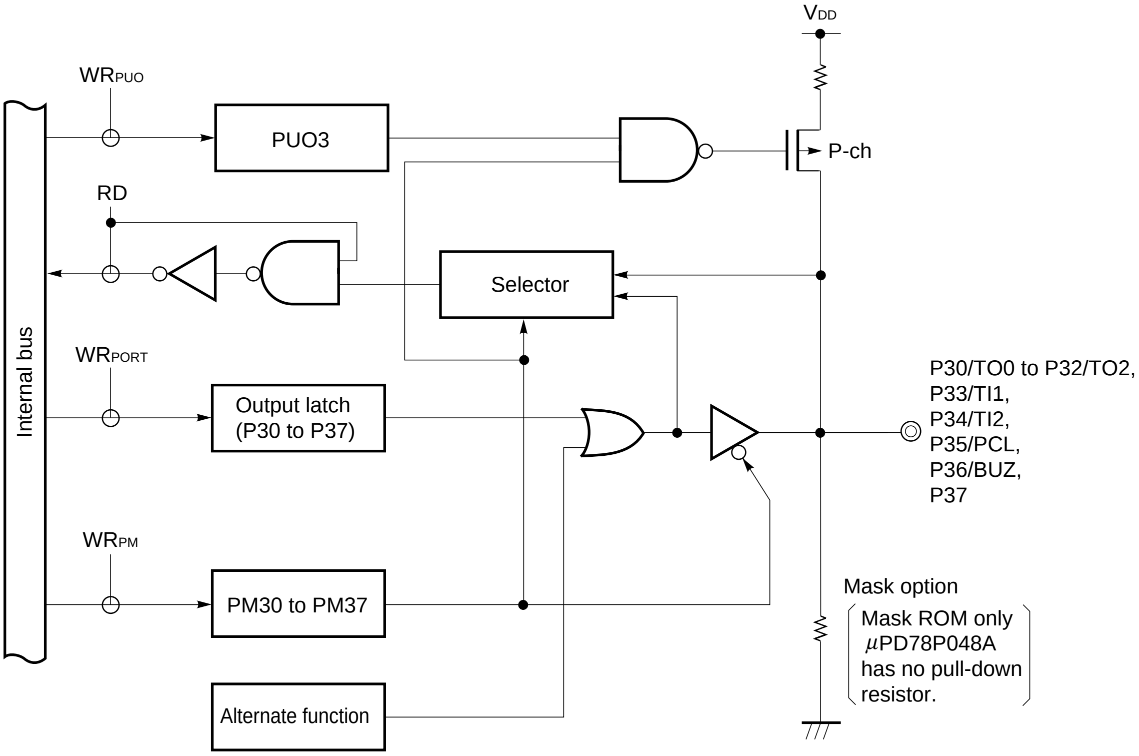
Port 3 can drive LEDs directly.

Alternate functions include timer input/output, clock output and buzzer output.

$\overline{\text{RESET}}$ input sets port 3 to input mode.

Figure 4-7 shows a block diagram of port 3.

Figure 4-7. P30 to P37 Block Diagram



PUO : Pull-up resistor option register

PM : Port mode register

RD : Port 3 read signal

WR : Port 3 write signal

4.2.5 Port 7

Port 7 is a 5-bit input/output port with output latch. Pins from 70 to 74 can specify I/O mode bit-wise with the port mode register 7 (PM7). Mask ROM version can contain pull-up resistors bit-wise with the mask option. The μPD78P048A does not contain pull-up resistors.

Port 7 can drive LEDs directly.

$\overline{\text{RESET}}$ input sets port 7 to input mode.

Figure 4-8 shows a block diagram of port 7.

Caution The low-level input leak current flowing to the P70 to P74 pins varies depending on the following conditions.

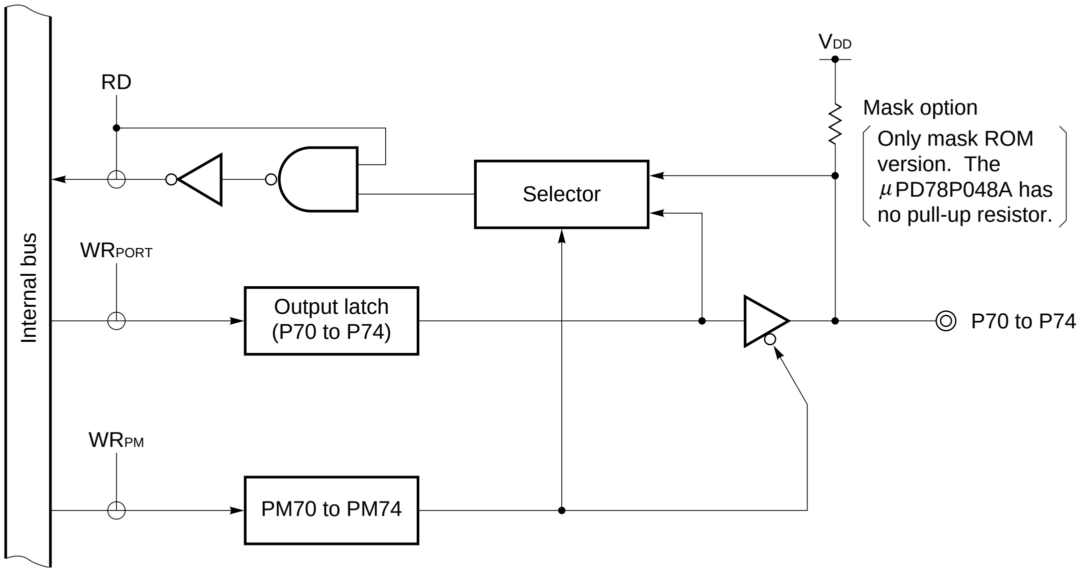
[For Mask ROM version]

- **When a pull-up resistor is connected:**
  - **–3 μA (max.) regardless of operational conditions**
- **When a pull-up resistor is not connected:**
  - **–150 μA (max.) during 1.5 clock cycles after a read instruction execution to port 7 (P7) or port mode register 7 (PM7)**
  - **–3 μA (max.) in other conditions**

[For PROM version]

- **–150 μA (max.) during 1.5 clock cycles after a read instruction execution to port 7 (P7) or port mode register 7 (PM7)**
- **–3 μA (max.) in other conditions**

Figure 4-8. P70 to P74 Block Diagram


VDD
RD
Mask option
Only mask ROM version. The μPD78P048A has no pull-up resistor.
Selector
Internal bus
WRPORT
Output latch (P70 to P74)
P70 to P74
WRPM
PM70 to PM74

PM	:	Port mode register
RD	:	Port 7 read signal
WR	:	Port 7 write signal

4.2.6 Port 8

Port 8 is a 2-bit output only port. Mask ROM version can specify pull-down resistors bit-wise with the mask option. Pull-down resistor connection to V_{LOAD} or V_{SS} can be specified in 2-bit units. The μPD78P048A contains a pull-down resistor which is connected to V_{LOAD}.

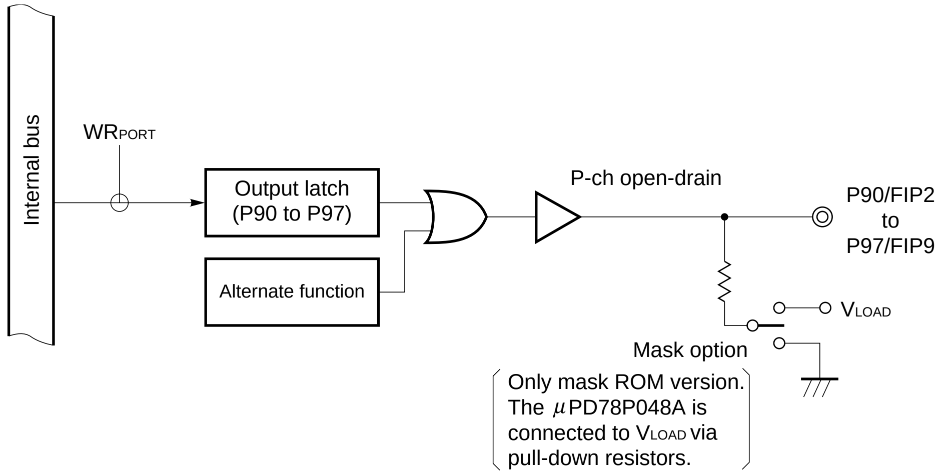
Port 8 can drive LEDs directly.

In addition, FIP controller/driver digit output is provided as an alternate function.

$\overline{\text{RESET}}$ input sets port 8 to output mode.

Figure 4-9 shows a block diagram of port 8.

Figure 4-9. P80 and P81 Block Diagram



Internal bus
WRPORT
Output latch (P90 to P97)
Alternate function
P-ch open-drain
P90/FIP2 to P97/FIP9
VLOAD
Mask option
Only mask ROM version. The µPD78P048A is connected to VLOAD via pull-down resistors.

WR : Port 8 write signal

4.2.7 Port 9

Port 9 is an 8-bit output only port. Mask ROM version can specify pull-down resistors bit-wise with the mask option. Pull-down resistor connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P048A contains a pull-down resistor which is connected to V_{LOAD}.

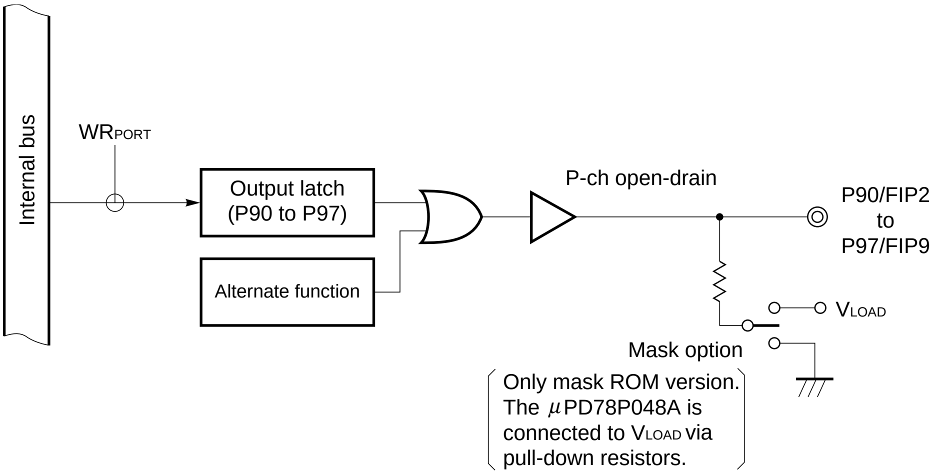
Port 9 can drive LEDs directly.

In addition, FIP controller/driver digit output is provided as an alternate function.

$\overline{\text{RESET}}$ input sets port 9 to output mode.

Figure 4-10 shows a block diagram of port 9.

Figure 4-10. P90 to P97 Block Diagram


Internal bus
WRPORT
Output latch (P90 to P97)
Alternate function
P-ch open-drain
P90/FIP2 to P97/FIP9
VLOAD
Mask option
Only mask ROM version. The μPD78P048A is connected to VLOAD via pull-down resistors.

WR : Port 9 write signal

4.2.8 Port 10

Port 10 is an 8-bit output only port. Mask ROM version can specify pull-down resistors bit-wise with the mask option. Pull-down resistors connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P048A contains pull-down resistor which are connected to V_{LOAD} from P100 to P105.

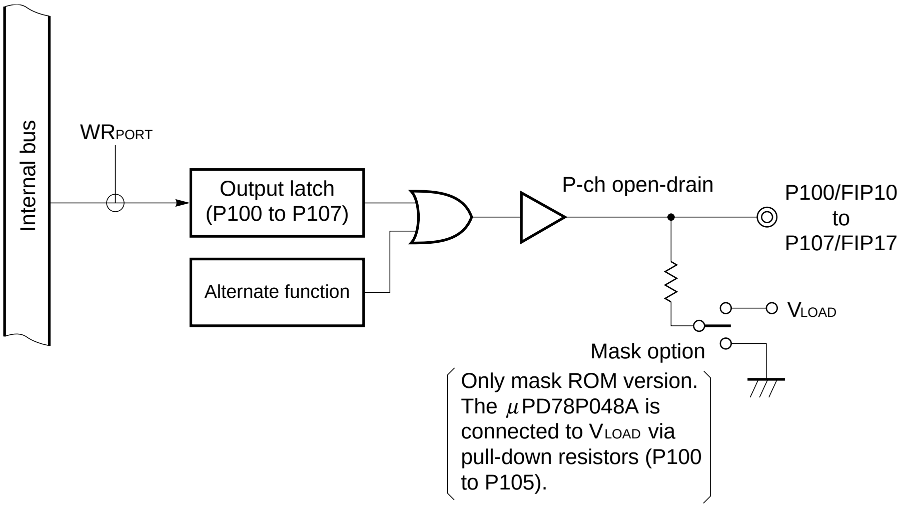
From P100 to P105 can drive LEDs directly.

In addition, FIP controller/driver segment/digit output is provided as an alternate function.

$\overline{\text{RESET}}$ input sets port 10 to output mode.

Figure 4-11 shows a block diagram of port 10.

Figure 4-11. P100 to P107 Block Diagram


Internal bus
WRPORT
Output latch (P100 to P107)
Alternate function
P-ch open-drain
P100/FIP10 to P107/FIP17
VLOAD
Mask option
Only mask ROM version. The μPD78P048A is connected to VLOAD via pull-down resistors (P100 to P105).

WR : Port 10 write signal

4.2.9 Port 11

Port 11 is an 8-bit input/output port with output latch. P110 to P117 pins can specify input/output mode bit-wise with the port mode register 11 (PM11). Mask ROM version can specify pull-down resistors bit-wise with the mask option. Pull-down resistors connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P048A does not contain pull-down resistors.

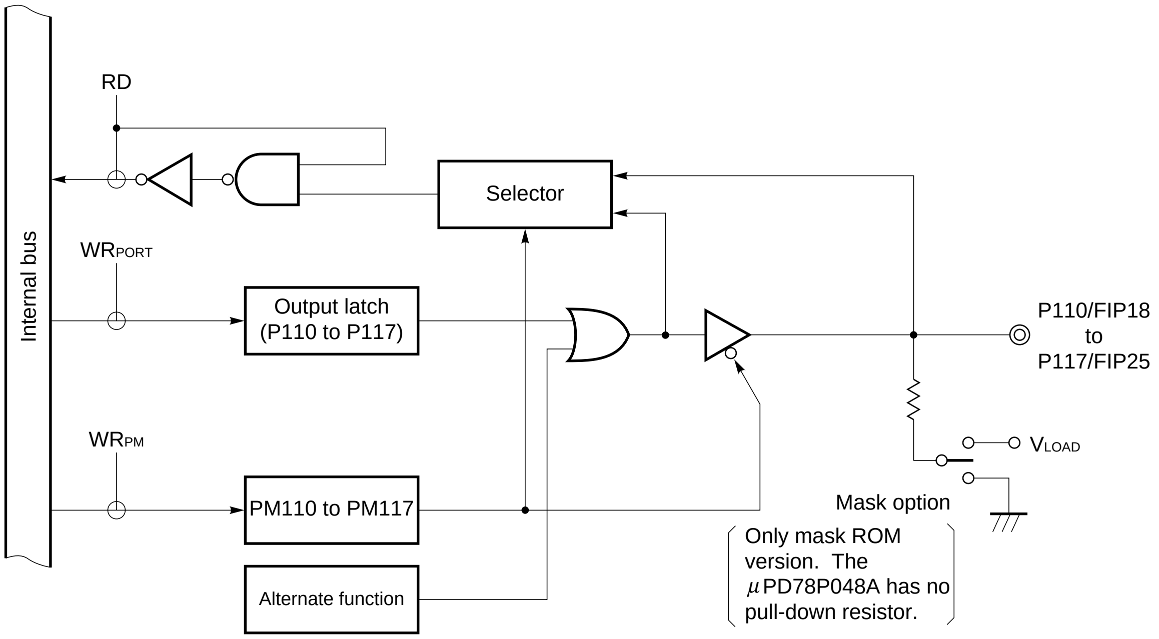
Port 11 can drive LEDs directly.

In addition, FIP controller/driver segment output is provided as an alternate function.

$\overline{\text{RESET}}$ input sets port 11 to input mode.

Figure 4-12 shows a block diagram of port 11.

Figure 4-12. P110 to P117 Block Diagram



PM	:	Port mode register
RD	:	Port 11 read signal
WR	:	Port 11 write signal

4.2.10 Port 12

This is an 8-bit input/output port with output latches. P120 to P127 pins can specify input mode/output mode bit-wise by means of port mode register 12 (PM12).

Mask ROM version can specify pull-down resistors bit-wise with the mask option. Pull-down resistors connection to V_{LOAD} or V_{SS} can be specified in 4-bit units. The μPD78P048A does not contain pull-down resistors.

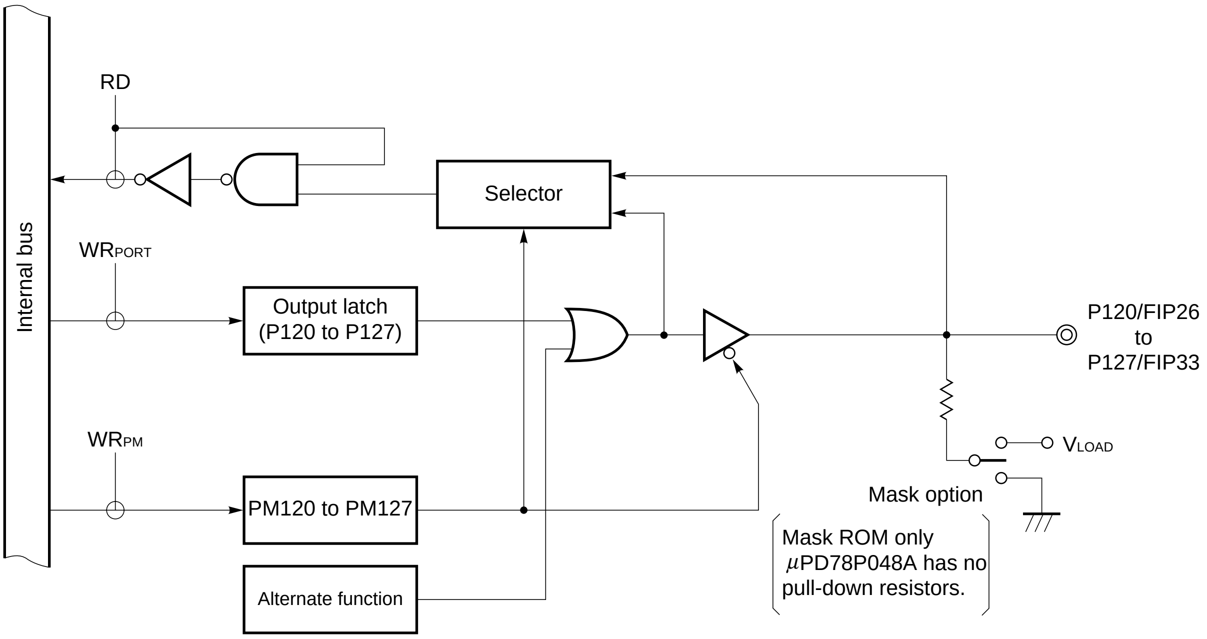
Port 12 can drive LEDs directly.

In addition, FIP controller/driver segment output is provided as an alternate function.

$\overline{\text{RESET}}$ input sets the input mode.

The port 12 block diagram is shown in Figure 4-13.

Figure 4-13. P120 to P127 Block Diagram



PM	:	Port mode register
RD	:	Port 12 read signal
WR	:	Port 12 write signal

4.3 Port Function Control Registers

The following two types of registers control the ports.

- Port mode registers (PM0, PM1, PM2, PM3, PM7, PM11, PM12)
- Pull-up resistor option register (PUO)

(1) Port mode registers (PM0, PM1, PM2, PM3, PM7, PM11, PM12)

These registers are used to set port input/output in 1-bit units.
PM0, PM1, PM2, PM3, PM7, PM11, and PM12 are independently set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets PM0 and PM7 to 1FH, other registers to FFH.

When a port pin is used as its alternate function pin, set the port mode register and the output latch according to Table 4-3.

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

2. Pins P80 and P81, P90 to P97 and P100 to P107 are output only pins.

3. As port 0 has an alternate function as external interrupt request input, when the port function output mode is specified and the output level is changed, the interrupt request flag is set. When the output mode is used, therefore, the interrupt mask flag should be set to 1 beforehand.

Table 4-3. Port Mode Register and Output Latch Setting when Alternate Function is Used

Pin Name	Alternate Function		PMxx	Pxx	Pin Name	Alternate Function		PMxx	Pxx
	Function Name	Input/ Output				Function Name	Input/ Output		
P00	INTP0	Input	1 (fixed)	None	P30 to P32	TO0 to TO2	Output	0	0
	TI0	Input	1 (fixed)	None	P33, P34	TI1, TI2	Input	1	×
P01, P02	INTP1, INTP2	Input	1	×	P35	PCL	Output	0	0
P03	INTP3	Input	1	×	P36	BUZ	Output	0	0
	CI0	Input	1	×	P110 to P117	FIP18 to FIP25	Output	0	0
P04[Note]	XT1	Input	1 (fixed)	None	P120 to P127	FIP26 to FIP33	Output	0	0
★ P10 to P17[Note]	ANI0 to ANI7	Input	1	×					

Note If a read is executed to these ports in the alternate function mode, the read data will be undefined values.

Caution When Port 2 is used as serial interface pin, input/output and the output latch should be set according to functions. For setting, refer to Figure 14-3. Serial Operating Mode Register 0 Format and Figure 15-3. Serial Operating Mode Register 1 Format.

Remark × : don't care
PMxx : Port mode register
Pxx : Port output latch

Figure 4-14. Port Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
PM0	0	0	0	1	PM03	PM02	PM01	1	FF20H	1FH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM7	0	0	0	PM74	PM73	PM72	PM71	PM70	FF27H	1FH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FF2BH	FFH	R/W
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn Pin Input/Output Mode Selection (m = 0, 1, 2, 3, 7, 11, 12 : n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

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

The PUO register enables or disables the on-chip pull-up resistor for each port pin. To enable the on-chip pull-up resistor of a port pin, the pin must be in the input mode and the corresponding bit in the PUO register must be set to 1. For any pin specified to the output mode or used as an analog input pin, the on-chip pull-up resistors cannot be used, regardless of the PUO register setting.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Cautions 1. P00 and P04 pins do not incorporate a pull-up resistor.

2. When port 1 is used as analog input for A/D converter, an on-chip pull-up resistor cannot be used even if 1 is set in PUO1.

Figure 4-15. Pull-Up Resistor Option Register Format

Symbol	7	6	5	4	③	②	①	⓪	Address	After Reset	R/W
PUO	0	0	0	0	PUO3	PUO2	PUO1	PUO0	FFF7H	00H	R/W

PUOm	Pm Internal Pull-up Resistor Selection (m = 0, 1, 2, 3)
0	Internal pull-up resistor not used
1	Internal pull-up resistor used

4.4 Port Function Operations

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

4.4.1 Writing to input/output port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.

Once data is written to the output latch, it is retained until data is written to the output latch again.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is OFF, the pin status does not change.

Once data is written to the output latch, it is retained until data is written to the output latch again.

Caution In the case of 1-bit memory manipulation instruction, although a single bit is manipulated, the port is accessed in 8-bit units. Therefore, on a port with a mixture of input and output pins, the output latch contents for pins specified as input are undefined except for the manipulated bit.

4.4.2 Reading from input/output port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin status is read by a transfer instruction.

The output latch contents do not change.

4.4.3 Operations on input/output port

(1) In output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.

Once data is written to the output latch, it is retained until data is written to the output latch again.

(2) In input mode

The output latch contents become undefined. However, the pin status does not change because the output buffer is turned off.

Caution In the case of 1-bit memory manipulation instruction, although a single bit is manipulated, the port is accessed in 8-bit units. Therefore, on a port with a mixture of input and output pins, the output latch contents for pins specified as input are undefined except for the manipulated bit.

4.5 Selection of Mask Option

The following mask option is provided in mask ROM version. The μPD78P048A has no mask option.

Table 4-4. Comparison between Mask ROM Version and the μPD78P048A

Pin Name	Mask option of mask ROM version	μPD78P048A
P30/TO0 to P32/TO2, P33/T11, P34/T12, P35/PCL, P36/BUZ, P37	Can incorporate on-chip pull-down resistors bit-wise.	Does not have on-chip pull-down resistors.
P70 to P74	Can incorporate on-chip pull-up resistors bit-wise.	Does not have on-chip pull-up resistors.
P80/FIP0, P81/FIP1	Can incorporate on-chip pull-down resistors bit-wise. Can specify to be connected to V_{LOAD} or V_{SS} in 2-bit units from P80.	Has on-chip pull-down resistors. (Connected to V_{LOAD}.)
P90/FIP2 to P97/FIP9	Can incorporate on-chip pull-down resistors bit-wise. Can specify to be connected to V_{LOAD} or V_{SS} in 4-bit units from P90.	Has on-chip pull-down resistors. (Connected to V_{LOAD}.)
P100/FIP10 to P107/FIP17	Can incorporate on-chip pull-down resistors bit-wise. Can specify to be connected to V_{LOAD} or V_{SS} in 4-bit units from P100.	P100/FIP10 to P105/FIP15 pins have on-chip pull-down resistors (Connected to V_{LOAD}). P106/FIP16 and P107/FIP17 pins do not have on-chip pull-down resistors.
P110/FIP18 to P117/FIP25, P120/FIP26 to P127/FIP33	Can incorporate on-chip pull-down resistors bit-wise. Can specify to be connected to V_{LOAD} or V_{SS} in 4-bit units from P110.	Does not have on-chip pull-down resistors.

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

The clock generator generates the clock to be supplied to the CPU and peripheral hardware. The following two types of system clock oscillators are available.

(1) Main system clock oscillator

This circuit oscillates at frequencies of 1 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).

(2) Subsystem clock oscillator

The circuit oscillates at a frequency of 32.768 kHz. Oscillation cannot be stopped. If the subsystem clock oscillator is not used, the internal feedback resistor can be disabled by the processor clock control register (PCC). This enables to decrease power consumption in the STOP mode.

The noise eliminator is operating automatically to reduce the effect of switching noise when FIP is displayed.

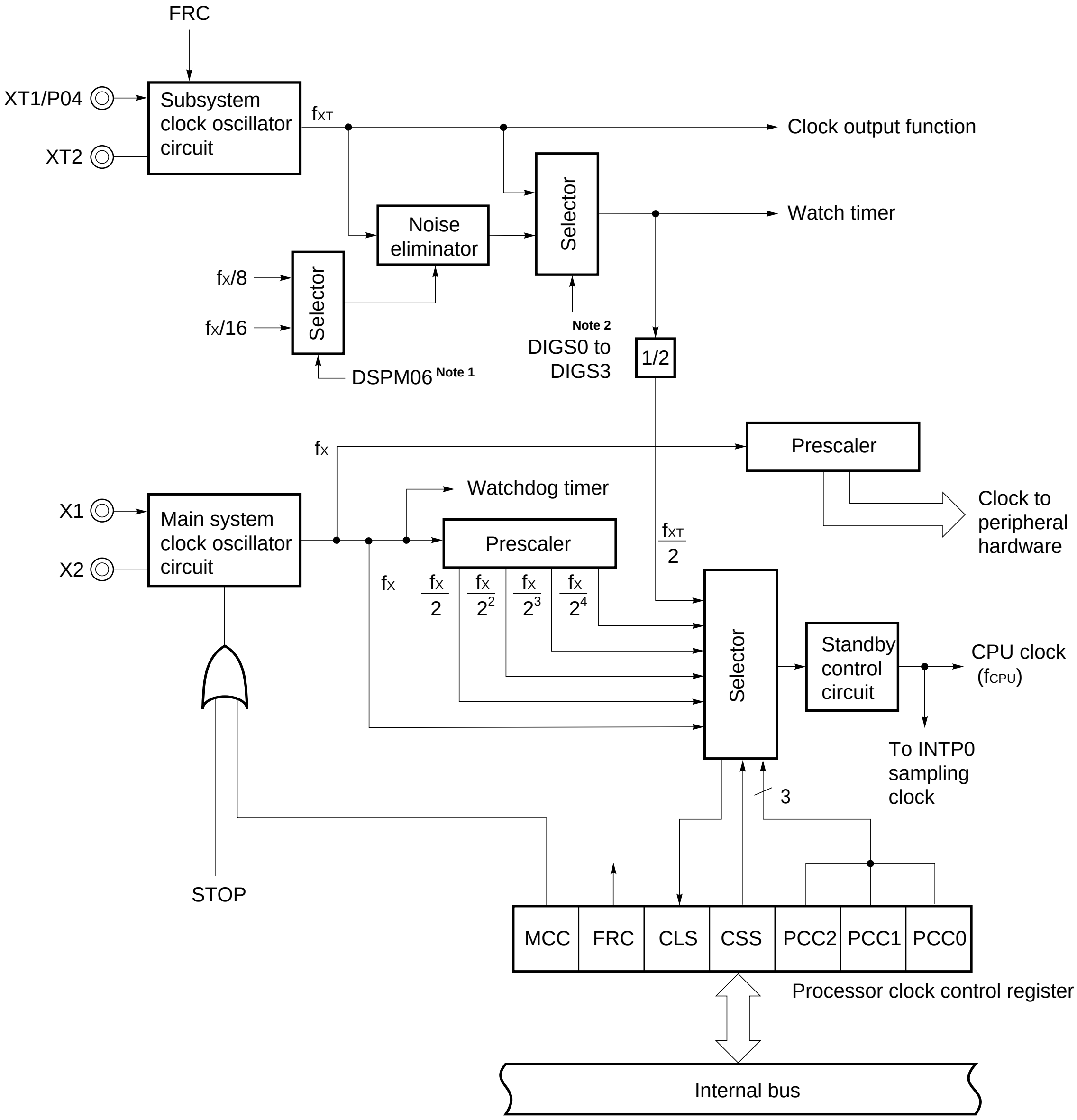
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

Item	Configuration
Control register	Processor clock control register (PCC) Display mode register 0 (DSPM0) Display mode register 1 (DSPM1)
Oscillator	Main system clock oscillator Subsystem clock oscillator

Figure 5-1. Clock Generator Block Diagram



Notes 1. Bit 6 of display mode register 0 (DSPM0)

2. Bits 4 to 7 of display mode register 1 (DSPM1)

5.3 Clock Generator Control Register

The clock generator is controlled by the following three registers:

- Processor clock control register (PCC)
- Display mode register 0 (DSPM0)
- Display mode register 1 (DSPM1)

(1) Processor clock control register (PCC)

The PCC sets CPU clock selection, the ratio of division, main system clock oscillator operation/stop and subsystem clock oscillator internal feedback resistor use/disable.

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

$\overline{\text{RESET}}$ input sets the PCC to 04H.

Figure 5-2. Feedback Resistor of Subsystem Clock

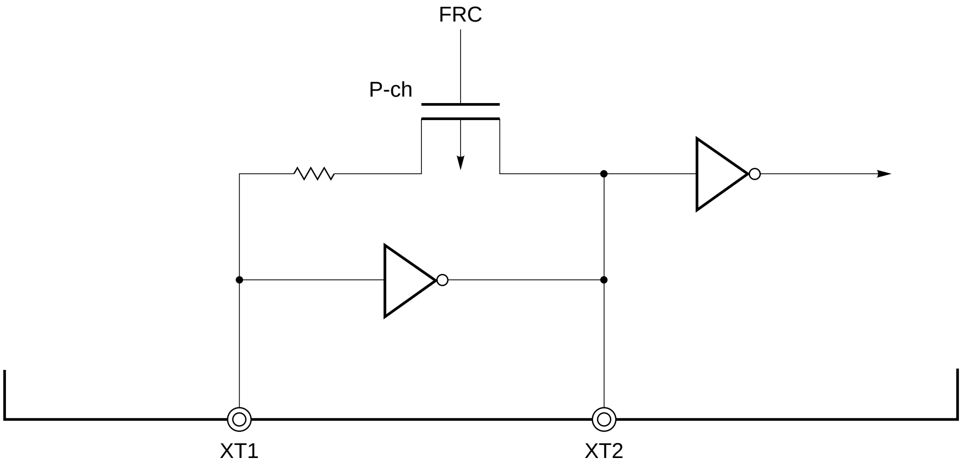


Figure 5-3. Processor Clock Control Register Format

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

R/W

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) selection Note 2
0	0	0	0	f_X
	0	0	1	$f_X/2$
	0	1	0	$f_X/2^2$
	0	1	1	$f_X/2^3$
	1	0	0	$f_X/2^4$
1	0	0	0	$f_{XT}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

R

CLS	CPU Clock Status
0	Main system clock
1	Subsystem clock

R/W

FRC	Subsystem Clock Feedback Resistor Selection
0	Internal feedback resistor used
1	Internal feedback resistor not used

R/W

MCC	Main System Clock Oscillation Control Note 3
0	Oscillation possible
1	Oscillation stopped

Notes **1.** Bit 5 is Read Only.

2. FIP can be displayed only when CSS is 0 and PCC2 to PCC0 are 000 or 001.

3. When the CPU is operating on the subsystem clock, MCC should be used to stop the main system clock oscillation. A STOP instruction should not be used.

Caution Bit 3 must be set to 0.

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

2. f_{XT} : Subsystem clock oscillation frequency

The fastest instruction of the μPD78044F Subseries is executed in two CPU clocks. Therefore, the relation between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in Table 5-2.

Table 5-2. Relation between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$
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
$f_{XT}/2$	122 μs

f_X = 5.0 MHz, f_{XT} = 32.768 kHz
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

(2) Display mode register 0 (DSPM0)

This register sets the mode for noise elimination circuit of the subsystem clock.
DSPM0 is set with 8-bit memory manipulation instruction.
Only bit 7 (KSF) can be read with 1-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears DSPM0 to 00H.

Remark In addition to the function mentioned above, DSPM0 can also set the number of display segments and display key scan timing.

Figure 5-4. Display Mode Register 0 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM0	KSF	DSPM06	0	0	SEGS3	SEGS2	SEGS1	SEGS0	FFA0H	00H	R/W [Note 1]

SEGS3	SEGS2	SEGS1	SEGS0	Display segment
0	0	0	0	9 segments
0	0	0	1	10 segments
0	0	1	0	11 segments
0	0	1	1	12 segments
0	1	0	0	13 segments
0	1	0	1	14 segments
0	1	1	0	15 segments
0	1	1	1	16 segments
1	0	0	0	17 segments
1	0	0	1	18 segments
1	0	1	0	19 segments
1	0	1	1	20 segments
1	1	0	0	21 segments
1	1	0	1	22 segments
1	1	1	0	23 segments
1	1	1	1	24 segments

DSPM06	Subsystem Clock Noise Reduction Circuit Mode setting [Note 2]
0	2.5 MHz < $f_X \leq$ 5.0 MHz
1	1.25 MHz < $f_X \leq$ 2.5 MHz

KSF	Timing Status
0	Display timing
1	Key scan timing

Notes 1. Bit 7 (KSF) is read only.

2. Set the value following the main system clock frequency (fx) to be used. The noise elimination circuit is effective when FIP is driven.

Remark f_X: Main system clock oscillation frequency

(3) Display mode register 1 (DSPM1)

Register to set display operating/stopping.

DSPM1 is set by 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears DSPM1 to 00H.

Remark DSPM1 has functions which set display digits, cut width of the digital signal and display cycle except display operating/stopping.

Figure 5-5. Display Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM1	DIGS3	DIGS2	DIGS1	DIGS0	DIMS3	DIMS2	DIMS1	DIMS0	FFA1H	00H	R/W

DIMS0	Display Cycle Selection
0	1024/f_X is 1 display cycle. (1 display cycle = 204.8 μs: when operated at 5.0 MHz)
1	2048/f_X is 1 display cycle. (1 display cycle = 409.6 μs: when operated at 5.0 MHz)

DIMS3	DIMS2	DIMS1	Digit signal cut width
0	0	0	1/16
0	0	1	2/16
0	1	0	4/16
0	1	1	6/16
1	0	0	8/16
1	0	1	10/16
1	1	0	12/16
1	1	1	14/16

DIGS3	DIGS2	DIGS1	DIGS0	Display digit
0	0	0	0	Display stopped (Static display)[Note]
0	0	0	1	2 digits
0	0	1	0	3 digits
0	0	1	1	4 digits
0	1	0	0	5 digits
0	1	0	1	6 digits
0	1	1	0	7 digits
0	1	1	1	8 digits
1	0	0	0	9 digits
1	0	0	1	10 digits
1	0	1	0	11 digits
1	0	1	1	12 digits
1	1	0	0	13 digits
1	1	0	1	14 digits
1	1	1	0	15 digits
1	1	1	1	16 digits

Note When setting to display stopped, static display can be set by operating port output latch.

Remark f_X: Main system clock oscillation frequency

5.4 System Clock Oscillator

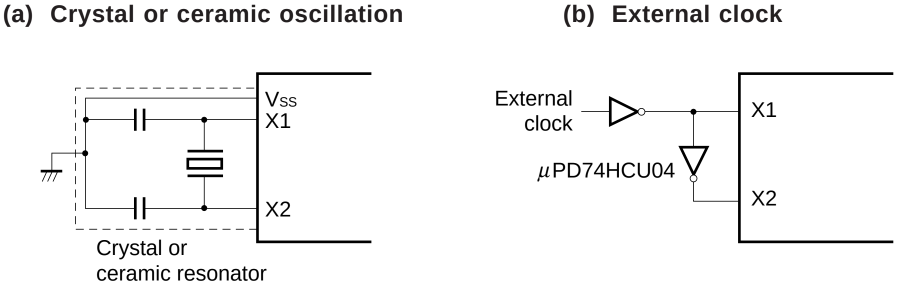
5.4.1 Main system clock oscillator

The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator (standard: 5.0 MHz) connected to the X1 and X2 pins.

External clocks can be input to the main system clock oscillator. In this case, input a clock signal to the X1 pin and an antiphase clock signal to the X2 pin.

Figure 5-6 shows an external circuit of the main system clock oscillator.

Figure 5-6. External Circuit of Main System Clock Oscillator



(a) Crystal or ceramic oscillation

(b) External clock

★ **Caution The STOP instruction cannot be executed and bit 7 (MCC) of the processor clock control register (PCC) cannot be set to 1 while an external clock is being input. This is because the X2 pin is pulled up to V_{DD}.**

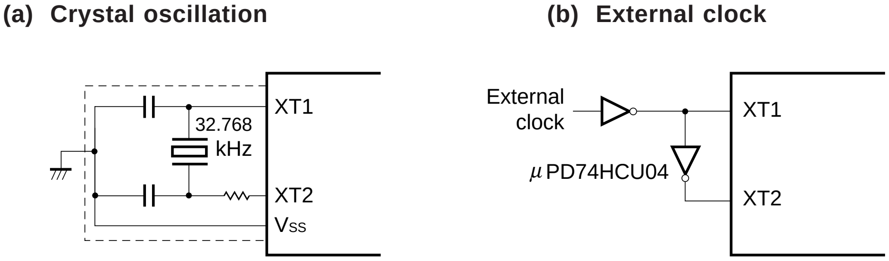
5.4.2 Subsystem clock oscillator

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

External clocks can be input to the subsystem clock oscillator. In this case, input a clock signal to the XT1 pin and an antiphase clock signal to the XT2 pin.

Figure 5-7 shows an external circuit of the subsystem clock oscillator.

Figure 5-7. External Circuit of Subsystem Clock Oscillator



(a) Crystal oscillation

(b) External clock

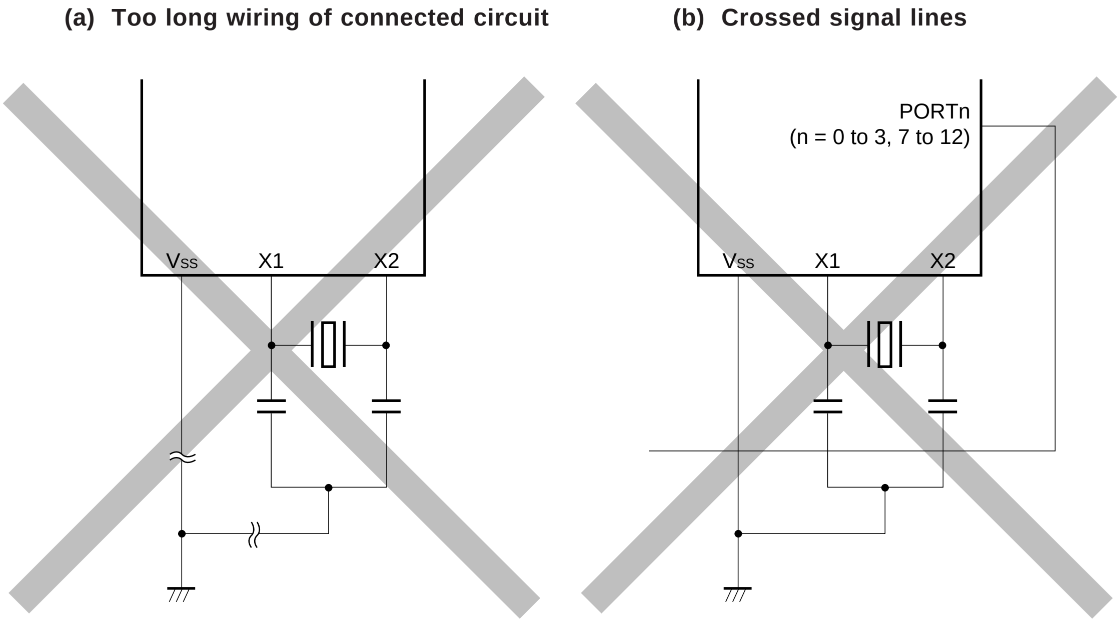
Cautions are shown to the next page.

Caution **When using a main system clock oscillator and a subsystem clock oscillator, carry out wiring in the broken-line area in Figures 5-6 and 5-7 as follows to prevent any effects from wiring capacities.**

- **Minimize the wiring length.**
- **Do not allow wiring to intersect with other signal conductors. Do not allow wiring to come near abruptly changing high current.**
- **Set the potential of the grounding position of the oscillator capacitor to that of VSS. Do not ground to any ground pattern where high current is present.**
- **Do not fetch signals from the oscillator.**

Take special note of the fact that the subsystem clock oscillator is a circuit with low-level amplification so that current consumption is maintained at low levels.
Figure 5-8 shows examples of oscillator having bad connection.

Figure 5-8. Examples of Oscillator with Bad Connection (1/2)

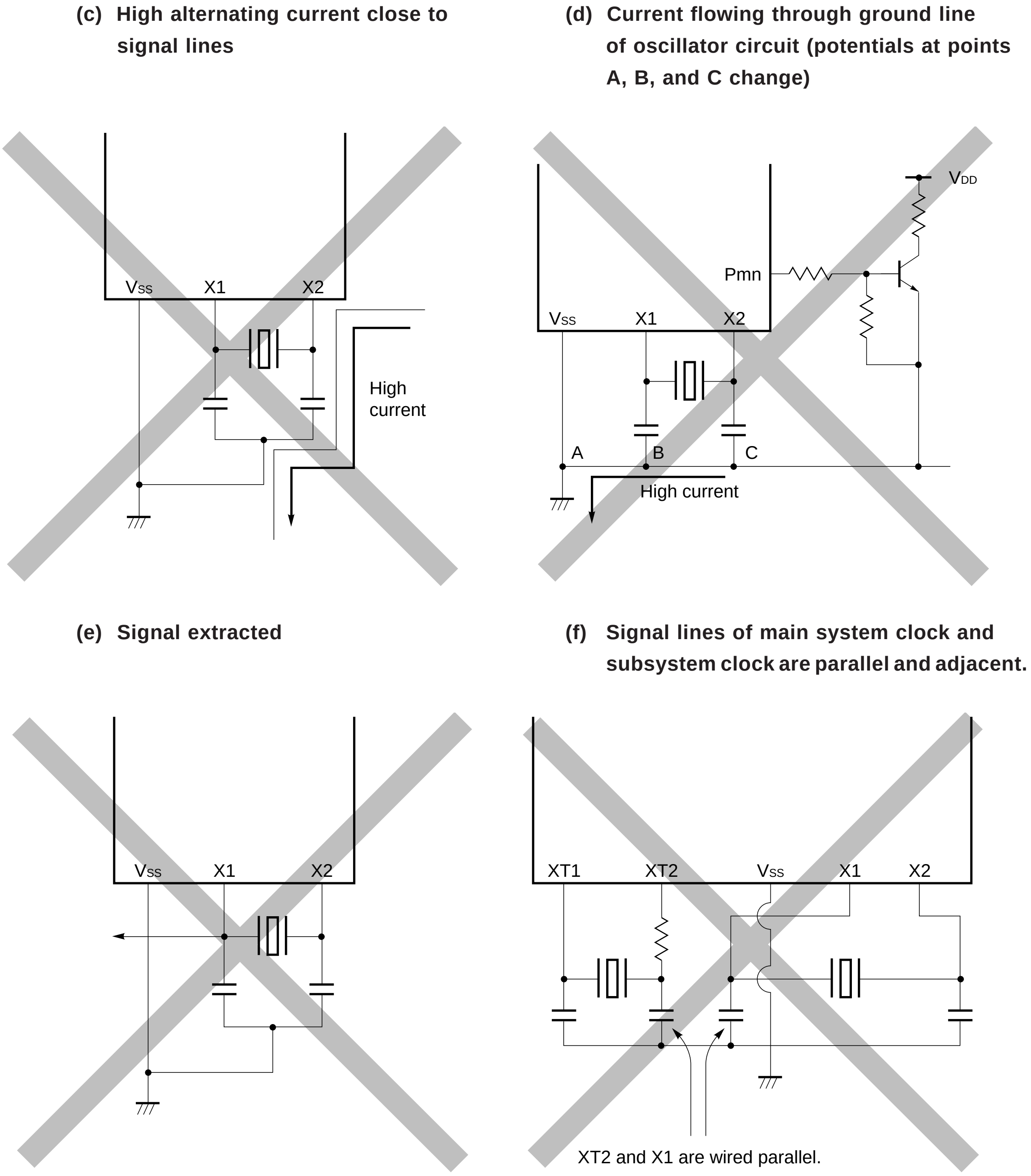


(a) Too long wiring of connected circuit

(b) Crossed signal lines

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

Figure 5-8. Examples of Oscillator with Bad Connection (2/2)



(c) High alternating current close to signal lines

(d) Current flowing through ground line of oscillator circuit (potentials at points A, B, and C change)

(e) Signal extracted

(f) Signal lines of main system clock and subsystem clock are parallel and adjacent.

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

5.4.3 Divider

The divider divides the main system clock oscillator output (f_X) and generates various clocks.

5.4.4 When no subsystem clocks are used

If it is not necessary to use subsystem clocks for low power consumption operations and clock operations, connect the XT1 and XT2 pins as follows.

XT1: Connect to V_{DD} or V_{SS}
XT2: Open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To prevent that from happening, the above internal feedback resistance can be removed with bit 6 (FRC) of the processor clock control register. In this case also, connect the XT1 and XT2 pins as described above.

5.5 Clock Generator Operations

The clock generator generates the following various types of clocks and controls the CPU operating mode including the standby mode.

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

The function and operation of the clock generator circuit are determined by the processor clock control register (PCC) as follows:

(a) Upon generation of $\overline{\text{RESET}}$ signal, the lowest speed mode of the main system clock (6.4 μs when operated at 5.0 MHz) is selected (PCC = 04H). Main system clock oscillation stops while low level is applied to the $\overline{\text{RESET}}$ pin.

(b) With the main system clock selected, one of the five (0.4 μs, 0.8 μs, 1.6 μs, 3.2 μs and 6.4 μs: when operated at 5.0 MHz) CPU clock stages can be selected by setting the PCC.

(c) With the main system clock selected, two standby modes, the STOP and HALT modes, are available. In a ★ system where the subsystem clock is not used, the current consumption in the STOP mode can be further reduced by not using the internal feedback resistor if so specified by bit 6 (FRC) of the PCC.

(d) The PCC can be used to select the subsystem clock and to operate the system with low current consumption (122 *m*s when operated at 32.768 kHz).

(e) With the subsystem clock selected, main system clock oscillation can be stopped with the PCC. The HALT mode can be used. However, the STOP mode cannot be used (subsystem clock oscillation cannot be stopped).

(f) The main system clock is divided and supplied to the peripheral hardware. The subsystem clock is supplied to the watch timer and clock output functions only. Thus, the watch function and the clock output function can also be continued in the standby state. However, since all other peripheral hardware operate with the main system clock, the peripheral hardware also stops if the main system clock is stopped (except external input clock operation).

5.5.1 Main system clock operations

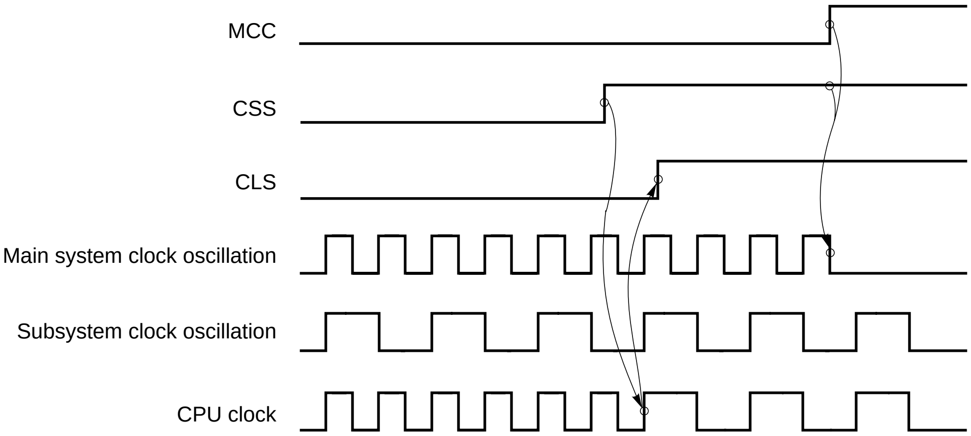
When operated with the main system clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 0), the following operations are carried out by PCC setting.

(a) Because the operation guarantee instruction execution speed depends on the power supply voltage, the instruction execution time can be changed by bits 0 to 2 (PCC0 to PCC2) of the PCC.

(b) If bit 7 (MCC) of the PCC is set to 1 when operated with the main system clock, the main system clock oscillation does not stop. When bit 4 (CSS) of the PCC is set to 1 and the operation is switched to subsystem clock operation (CLS = 1) after that, the main system clock oscillation stops (see **Figure 5-9**).

Figure 5-9. Main System Clock Stop Function (1/2)

(a) Operation when MCC is set after setting CSS with main system clock operation



(b) Operation when MCC is set with main system clock operation

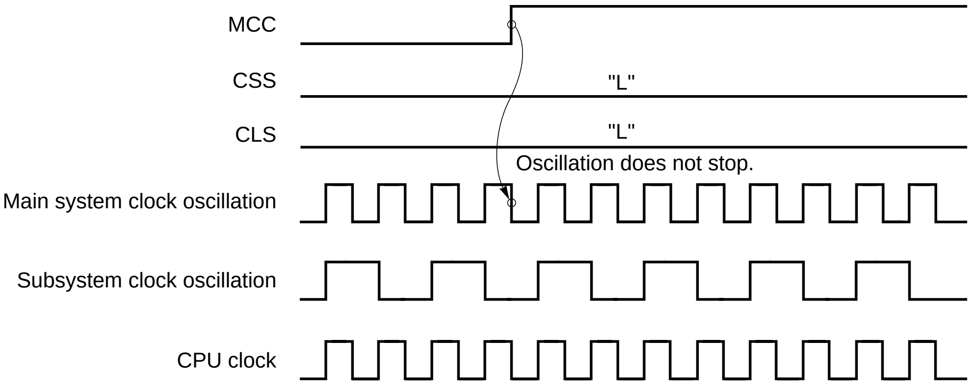
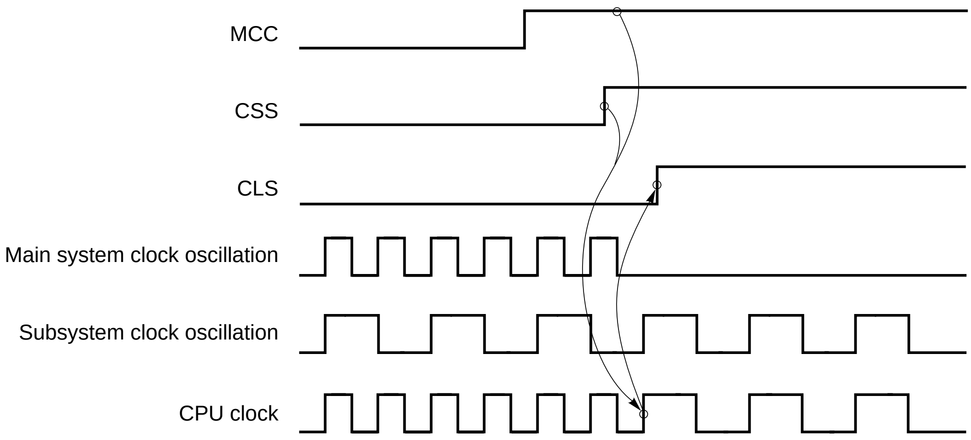

MCC
CSS
"L"
CLS
"L"
Oscillation does not stop.
Main system clock oscillation
Subsystem clock oscillation
CPU clock

Figure 5-9. Main System Clock Stop Function (2/2)

(c) Operation when CSS is set after setting MCC with main system clock operation



5.5.2 Subsystem clock operations

When operated with the subsystem clock (with bit 5 (CLS) of the processor clock control register (PCC) set to 1), the following operations are carried out.

(a) The minimum instruction execution time remains constant (122 μs when operated at 32.768 kHz) irrespective of bits 0 to 2 (PCC0 to PCC2) of the PCC.

(b) Watchdog timer counting stops.

Caution Do not execute the STOP instruction while the subsystem clock is in operation.

5.6 Changing System Clock and CPU Clock Settings

5.6.1 Time required for switchover between system clock and CPU clock

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

The actual switchover operation is not performed directly after writing to the PCC, but operation continues on the pre-switchover clock for several instructions (see **Table 5-3.**).

Determination as to whether the system is operating on the main system clock or the subsystem clock is performed by bit 5 (CLS) of the PCC register.

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

<table>
<tr><th colspan="4">Set Values before Switchover</th><th colspan="24">Set Values after Switchover</th></tr>
<tr><th rowspan="2">CSS</th><th rowspan="2">PCC2</th><th rowspan="2">PCC1</th><th rowspan="2">PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th><th>CSS</th><th>PCC2</th><th>PCC1</th><th>PCC0</th></tr>
<tr><th>0</th><th>0</th><th>0</th><th>0</th><th>0</th><th>0</th><th>0</th><th>1</th><th>0</th><th>0</th><th>1</th><th>0</th><th>0</th><th>0</th><th>1</th><th>1</th><th>0</th><th>1</th><th>0</th><th>0</th><th>1</th><th>x</th><th>x</th><th>x</th></tr>
<tr><td rowspan="5">0</td><td>0</td><td>0</td><td>0</td><td colspan="4"></td><td colspan="4">16 instructions</td><td colspan="4">16 instructions</td><td colspan="4">16 instructions</td><td colspan="4">16 instructions</td><td colspan="4">$f_X/2f_{XT}$ instructions
(64 instructions)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="4">8 instructions</td><td colspan="4"></td><td colspan="4">8 instructions</td><td colspan="4">8 instructions</td><td colspan="4">8 instructions</td><td colspan="4">$f_X/4f_{XT}$ instructions
(32 instructions)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td colspan="4">4 instructions</td><td colspan="4">4 instructions</td><td colspan="4"></td><td colspan="4">4 instructions</td><td colspan="4">4 instructions</td><td colspan="4">$f_X/8f_{XT}$ instructions
(16 instructions)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="4">2 instructions</td><td colspan="4">2 instructions</td><td colspan="4">2 instructions</td><td colspan="4"></td><td colspan="4">2 instructions</td><td colspan="4">$f_X/16f_{XT}$ instructions
(8 instructions)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4"></td><td colspan="4">$f_X/32f_{XT}$ instructions
(4 instructions)</td></tr>
<tr><td>1</td><td>×</td><td>×</td><td>×</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4">1 instruction</td><td colspan="4"></td></tr>
</table>

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

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

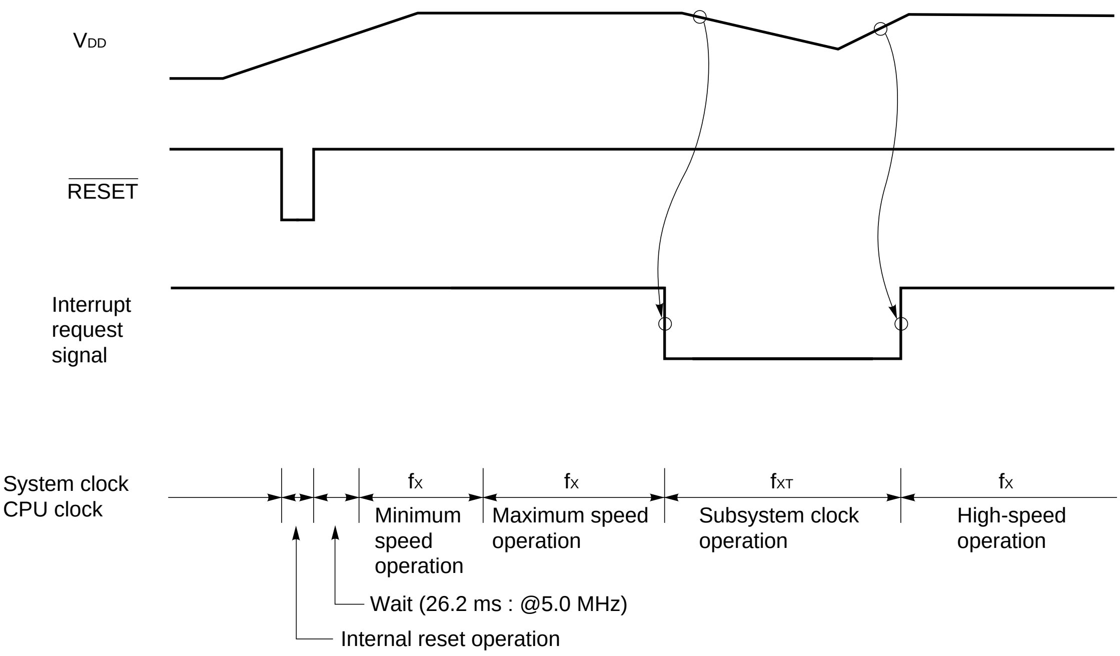
Remarks 1. One instruction is the minimum instruction execution time with the pre-switchover CPU clock.

2. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

5.6.2 System clock and CPU clock switching procedure

This section describes the switching procedure between system clock and CPU clock.

Figure 5-10. System Clock and CPU Clock Switching



[1] The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, main system clock starts oscillation. At this time, oscillation stabilization time ($2^{17}/f_X$) is secured automatically.
After that, the CPU starts executing the instruction at the minimum speed of the main system clock (6.4 μs when operated at 5.0 MHz).

[2] After the lapse of a sufficient time for the V_{DD} voltage to increase to enable operation at maximum speeds, the processor clock control register (PCC) is rewritten and the maximum-speed operation is carried out.

[3] Upon detection of a decrease of the V_{DD} voltage due to an interrupt request signal, the main system clock is switched to the subsystem clock (which must be in an oscillation stable state).

[4] Upon detection of V_{DD} voltage reset due to an interrupt request signal, 0 is set to bit 7 (MCC) of PCC and oscillation of the main system clock is started. After the lapse of time required for stabilization of oscillation, the PCC is rewritten and the maximum-speed operation is resumed.

Caution When the main system clock is stopped and the subsystem clock is performing, switching to the main system clock must be done after securing the oscillation stabilization time by a program.

[MEMO]

CHAPTER 6 16-BIT TIMER/EVENT COUNTER

6.1 Outline of Internal Timer of μPD78044F Subseries

This chapter explains the 16-bit timer/event counter. Before that, the timers incorporated into the μPD78044F Subseries, and the related functions are outlined below.

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

The TM0 can be used for an interval timer, PWM output, pulse widths measurement (infrared ray remote control receive function), external event counter or square wave output of any frequency.

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

TM1 and TM2 can be used to serve as an interval timer and an external event counter and to output square waves with any selected frequency. Two 8-bit timer/event counters can be used as one 16-bit timer/event counter (See **CHAPTER 7 8-BIT TIMER/EVENT COUNTER**).

(3) Watch timer (TM3)

This timer can set a flag every 0.5 sec. and simultaneously generates interrupt requests at the preset time intervals (See **CHAPTER 8 WATCH TIMER**).

(4) Watchdog timer (WDTM)

WDTM can perform the watchdog timer function or generate non-maskable interrupt requests, maskable interrupt requests and $\overline{\text{RESET}}$ at the preset time intervals (See **CHAPTER 9 WATCHDOG TIMER**).

(5) 6-bit up/down counter (UDC)

UDC can increment or decrement counter with valid edge input.
(See **CHAPTER 10 6-BIT UP/DOWN COUNTER**.)

(6) Clock output control circuit

This circuit supplies other devices with the divided main system clock and the subsystem clock (See **CHAPTER 11 CLOCK OUTPUT CONTROL CIRCUIT**).

(7) Buzzer output control circuit

This circuit outputs the buzzer frequency obtained by dividing the main system clock (See **CHAPTER 12 BUZZER OUTPUT CONTROL CIRCUIT**).

Table 6-1. Timer/Event Counter Types and Functions

		16-Bit Timer/ Event Counter	8-Bit Timer/ Event Counter	Watch Timer	Watchdog Timer	6-Bit Up/Down Counter
Type	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 perform both watch timer and interval timer functions at the same time.

2. Watchdog timer can perform either the watchdog timer function or the interval timer function.

6.2 16-Bit Timer/Event Counter Functions

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

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

(1) Interval timer

TM0 generates interrupt requests at the preset time interval.

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

Minimum Interval Time	Maximum Interval Time	Resolution
2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
2 × 1/f_X (400 ns)	2^{16} × 1/f_X (13.1 ms)	1/f_X (200 ns)
2^2 × 1/f_X (800 ns)	2^{17} × 1/f_X (26.2 ms)	2 × 1/f_X (400 ns)
2^3 × 1/f_X (1.6 μs)	2^{18} × 1/f_X (52.4 ms)	2^2 × 1/f_X (800 ns)
2^4 × 1/f_X (3.2 μs)	2^{19} × 1/f_X (104.9 ms)	2^3 × 1/f_X (1.6 μs)

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz

(2) PWM output

TM0 can generate 14-bit resolution PWM output.

(3) Pulse width measurement

TM0 can measure the pulse width of an externally input signal.

(4) External event counter

TM0 can measure the number of pulses of an externally input signal.

(5) Square-wave output

TM0 can output a square wave with any selected frequency.

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

Minimum Pulse Width	Maximum Pulse Width	Resolution
2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
2 × 1/fx (400 ns)	2^{16} × 1/fx (13.1 ms)	1/fx (200 ns)
2^2 × 1/fx (800 ns)	2^{17} × 1/fx (26.2 ms)	2 × 1/fx (400 ns)
2^3 × 1/fx (1.6 μs)	2^{18} × 1/fx (52.4 ms)	2^2 × 1/fx (800 ns)
2^4 × 1/fx (3.2 μs)	2^{19} × 1/fx (104.9 ms)	2^3 × 1/fx (1.6 μs)

Remarks 1. fx: Main system clock frequency

2. Values in parentheses when operated at fx = 5.0 MHz

6.3 16-Bit Timer/Event Counter Configuration

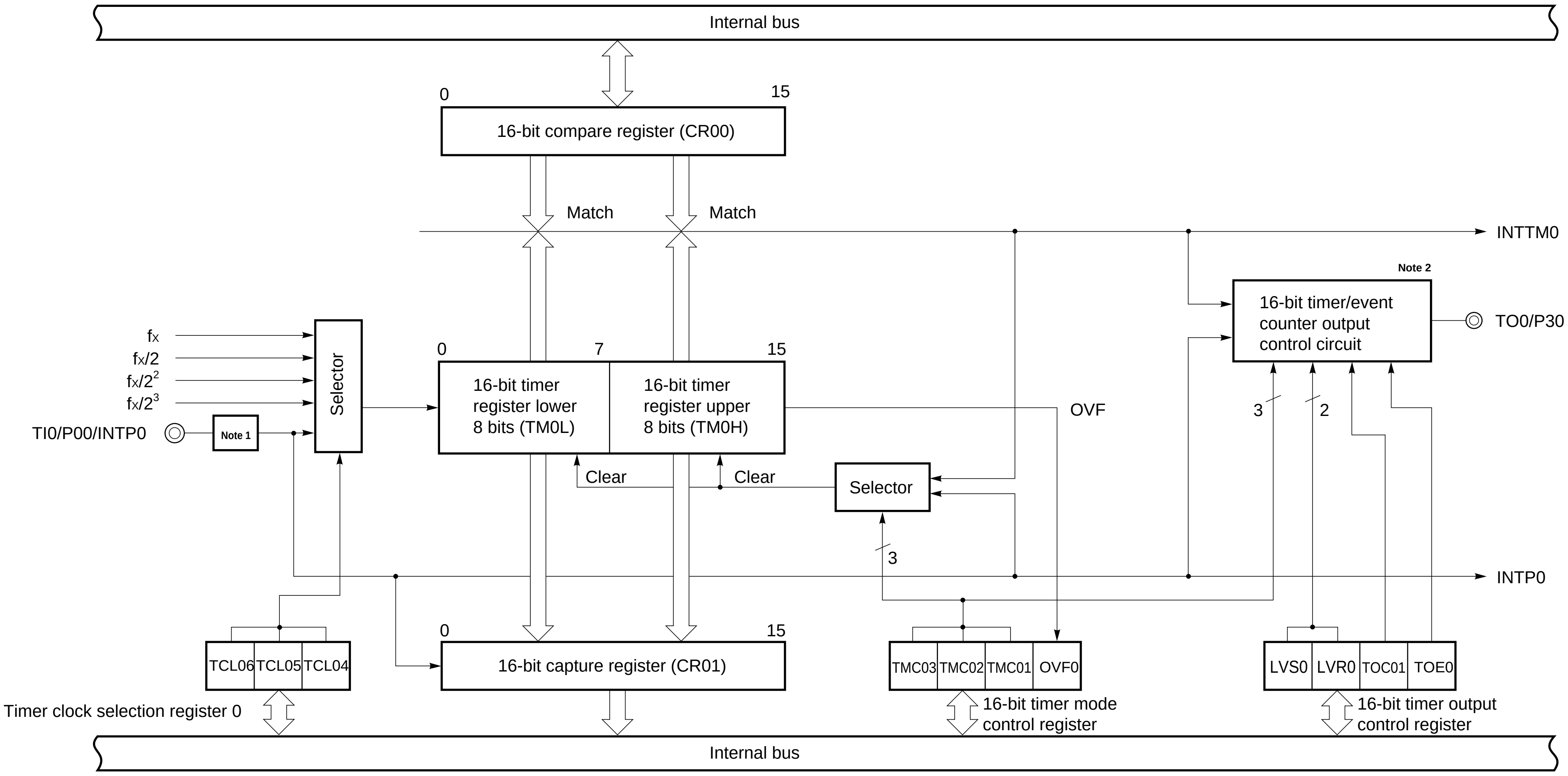
The 16-bit timer/event counter consists of the following hardware.

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

Item	Configuration
Timer register	16 bits × 1 (TM0)
Register	16-bit compare register: 1 (CR00) 16-bit capture register: 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 0 (INTM0) Sampling clock select register (SCS) Note

Note Refer to **Figure 17-1. Basic Configuration of Interrupt Function**.

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

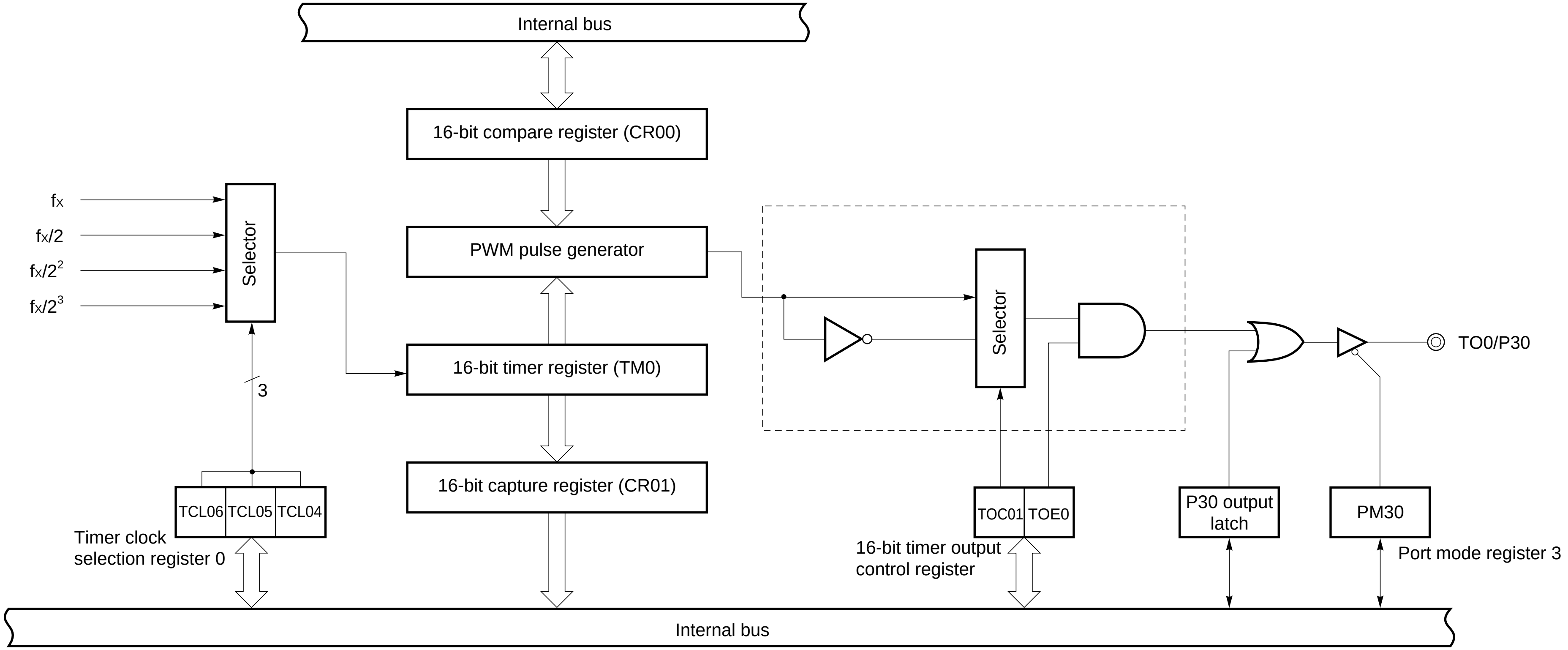


Internal bus
0
15
16-bit compare register (CR00)
Match
Match
INTTM0
Note 2
16-bit timer/event counter output control circuit
TO0/P30
fX
fX/2
fX/2²
fX/2³
TI0/P00/INTP0
Note 1
Selector
0
7
15
16-bit timer register lower 8 bits (TM0L)
16-bit timer register upper 8 bits (TM0H)
OVF
3
2
Clear
Clear
Selector
3
INTP0
0
15
TCL06
TCL05
TCL04
16-bit capture register (CR01)
TMC03
TMC02
TMC01
OVF0
LVS0
LVR0
TOC01
TOE0
Timer clock selection register 0
16-bit timer mode control register
16-bit timer output control register
Internal bus

Notes **1.** Edge detection circuit

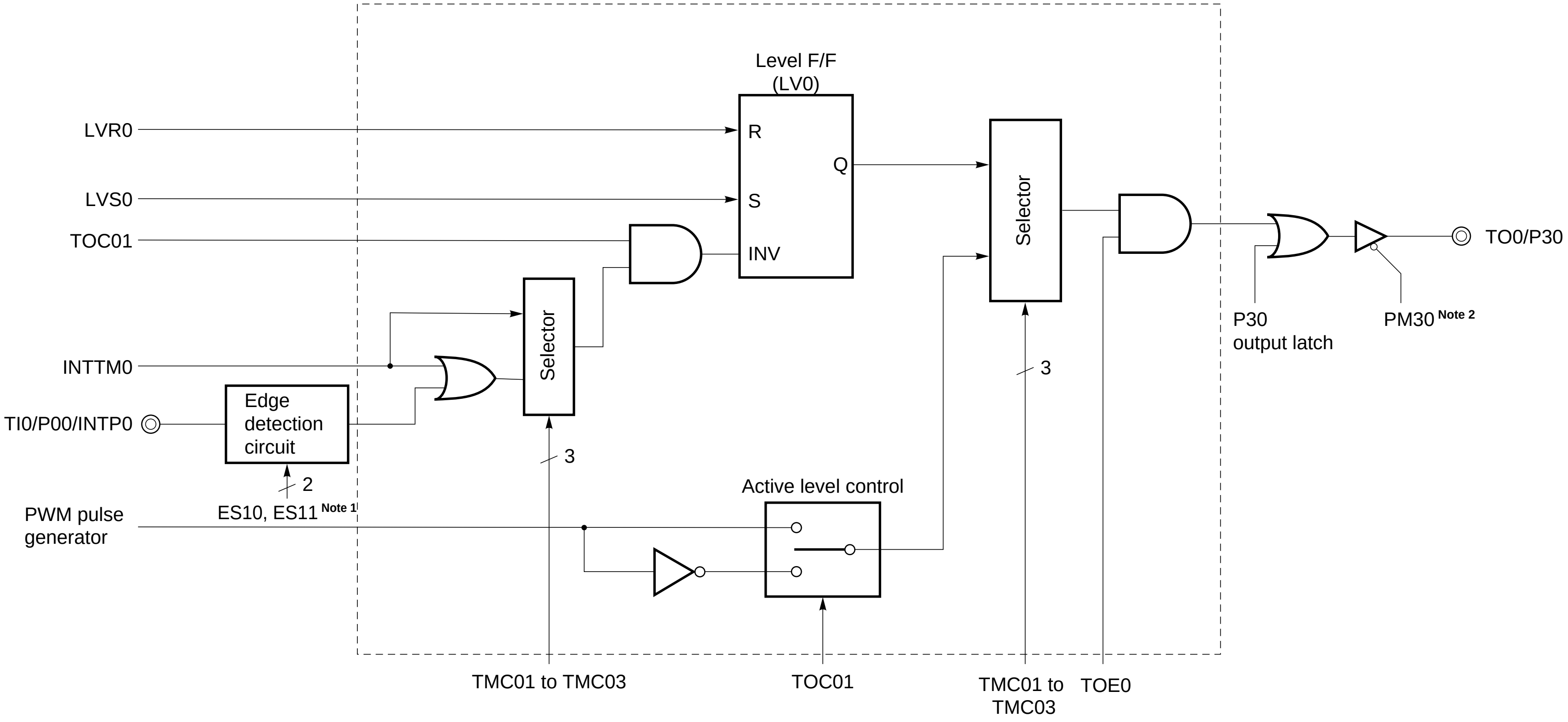
2. The configuration of the 16-bit timer/event counter output control circuit is shown in **Figure 6-3**.

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


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

Remark The circuitry enclosed by the dotted line is the output control circuit.

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



Notes **1.** Bits 2 and 3 of the external interrupt mode register (INTM0)

2. Bit 0 of the port mode register 3 (PM3)

Remark The circuitry enclosed by the dotted line is the output control circuit.

(1) 16-bit compare register (CR00)

CR00 is a 16-bit register where the value set in the CR00 is constantly compared with the 16-bit timer register (TM0) count value, and interrupt request (INTTM0) is generated if they match.

It can also be used as the register which holds the interval time when TM0 is set to interval timer operation, and as the register which sets the pulse width in the PWM operating mode.

CR00 is set with a 16-bit memory manipulation instruction. The value of 0001H to FFFFH can be set.

After $\overline{\text{RESET}}$ input, the value of CR00 is undefined.

Cautions 1. The PWM data (14 bits) must be set in the upper 14 bits of CR00. The lower two bits must be set to 00.

2. CR00 should be set to a valve other than 0000H. This means that when the timer is used as an event counter, 1-pulse count operation is not available.

3. If the value after CR00 is changed is smaller than that of the 16-bit timer register (TM0), TM0 continues counting, overflows, and then restarts counting from 0. Thus, if the value after CR00 change is smaller than that before change, it is necessary to restart the timer after changing CR00.

(2) 16-bit capture register (CR01)

CR01 is a 16-bit register capturing the content of 16-bit timer (TM0).

Capture trigger is INTP0/TI0 pin valid edge input. Setting of INTP0 valid edge is operated with the external interrupt mode register (INTM0).

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

After $\overline{\text{RESET}}$ input, the value set of CR01 is undefined.

Caution If the active edge for the TI0/P00 pin is input during a read from CR01, CR01 does not perform the capture operation and holds the previous data. In this case, however, the interrupt request flag (PIF0) is set because an active edge is detected.

(3) 16-bit timer register (TM0)

TM0 is a 16-bit register counts count pulse.

TM0 is read with a 16-bit memory manipulation instruction.

After $\overline{\text{RESET}}$ input, the value of TM0 is 0000H.

Caution As reading of the value of TM0 is performed via CR01, the previously set value of CR01 is lost.

6.4 16-Bit Timer/Event Counter Control Registers

The following six types of registers are used to 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)

This register is used to set the count clock of the 16-bit timer register.
TCL0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL0 value to 00H.

Remark TCL0 has the function of setting the PCL output clock in addition to that of setting the count clock of the 16-bit timer register.

Figure 6-4. Timer Clock Select Register 0 Format

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

TCL03	TCL02	TCL01	TCL00	PCL Output Clock Selection
0	0	0	0	f_{XT} (32.768 kHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
Other than above				Setting prohibited

TCL06	TCL05	TCL04	16-bit Timer Register Count Clock Selection
0	0	0	TI0 (Valid edge specifiable)
0	0	1	f_X (5.0 MHz)
0	1	0	$f_X/2$ (2.5 MHz)
0	1	1	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_X/2^3$ (625 kHz)
Other than above			Setting prohibited

CLOE	PCL Output Control
0	Output disabled
1	Output enabled

Cautions 1. Setting of the TI0/INTP0 pin valid edge is performed by the external interrupt mode register (INTM0), and selection of the sampling clock frequency is performed by the sampling clock select register (SCS).

2. When enabling PCL output, set TCL00 to TCL03, then set 1 in CLOE with a 1-bit memory manipulation instruction.

3. To read the count value when TI0 has been specified as the TM0 count clock, the value should be read from TM0, not from the 16-bit capture register (CR01).

4. If TCL0 is to be rewritten in data other than identical data, the timer operation must be stopped first.

Remarks 1. f_X: Main system clock frequency

2. f_{XT}: Subsystem clock oscillation frequency

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

4. TM0: 16-bit timer register

5. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

6. See **CHAPTER 11 Clock Output Control Circuit for PCL**.

(2) 16-bit timer mode control register (TMC0)

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

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

$\overline{\text{RESET}}$ input clears TMC0 value to 00H.

Caution The 16-bit timer register starts operation when TMC01 to TMC03 are set to a value other than 0, 0, 0 (operation stop mode). To stop the timer operation, set TMC01 to TCM03 to 0, 0, 0.

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

Symbol	7	6	5	4	3	2	1	⓪	Address	After Reset	R/W
TCL0	0	0	0	0	TMC03	TMC02	TMC01	OVF00	FF48H	00H	R/W

OVF0	16-Bit Timer Register Overflow Detection
0	Overflow not detected
1	Overflow detected

TMC03	TMC02	TMC01	Operating Mode & Clear Mode Selection	TO0 Output Timing Selection	Interrupt Generation
0	0	0	Operation stop (TM0 cleared to 0)	No change	Not generated
0	0	1	PWM mode (free running)	PWM pulse output	Generated on match between TM0 and CR00
0	1	0	Free running mode	Match between TM0 and CR00	
0	1	1		Match between TM0 and CR00 or TI0 valid edge	
1	0	0	Clear & start on TI0 valid edge	Match between TM0 and CR00	
1	0	1		Match between TM0 and CR00 or TI0 valid edge	
1	1	0	Clear & start on match between TM0 and CR00	Match between TM0 and CR00	
1	1	1		Match between TM0 and CR00 or TI0 valid edge	

Cautions 1. Switch the clear mode and the TO0 output timing after stopping the timer operation (by setting TMC01 to TMC03 to 0, 0, 0).

2. Set the valid edge of the TI0/INTP0 pin with an external interrupt mode register and select the sampling clock frequency with a sampling clock select register.

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

Remarks 1. TO0: 16-bit timer/event counter output pin

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

3. TM0: 16-bit timer register

4. CR00: Compare register 00

(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 R-S type flip-flop (LV0) setting/resetting, the active level in PWM mode, inversion enabling/disabling in modes other than PWM mode and data output mode.

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

$\overline{\text{RESET}}$ input clears TOC0 value to 00H.

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

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

TOE0	16-Bit Timer/Event Counter Output Control
0	Output disabled (Port mode)
1	Output enabled

TOC01	In PWM Mode	In Other Modes
	Active Level Selection	Timer Output F/F Control
0	Active high	Inversion operation disabled
1	Active low	Inversion operation enabled

LVS0	LVR0	16-Bit Timer/Event Counter Timer Output F/F Status Setting
0	0	No change
0	1	Timer output F/F reset (to 0)
1	0	Timer output F/F set (to 1)
1	1	Setting prohibited

Cautions 1. Timer operation must be stopped before setting TOC0.

2. If LVS0 and LVR0 are read after data is set, they will be 0.

(4) Port mode register 3 (PM3)

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

When using the P30/TO0 pin for timer output, set PM30 and output latch of P30 to 0.

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

$\overline{\text{RESET}}$ input sets PM3 value to FFH.

Figure 6-7. Port Mode Register 3 Format

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

PM3n	P3n Pin Input/Output Mode Selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(5) External interrupt mode register (INTM0)

This register is used to set INTP0 to INTP2 valid edges.
INTM0 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears INTM0 value to 00H.

Remarks **1.** INTP0 pin is dual used with TI0/P00.
2. INTP3 is fixed at falling edge.

Figure 6-8. External Interrupt Mode Register Format

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

ES11	ES10	INTP0 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES21	ES20	INTP1 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES31	ES30	INTP2 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

Caution **Setting the active edge of the INTP0/TI0/P00 pin should be made after the timer operation is stopped. To stop the timer, set bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register to 000.**

(6) Sampling clock select register (SCS)

This register sets clocks which undergo clock sampling of valid edges to be input to INTP0. When remote controlled reception is carried out using INTP0, digital noise is removed with sampling clock.

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

$\overline{\text{RESET}}$ input clears SCS value to 00H.

Figure 6-9. Sampling Clock Select Register Format

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

SCS1	SCS0	INTP0 Sampling Clock Selection
0	0	$f_X/2^{N+1}$
0	1	Setting prohibited
1	0	$f_X/2^6$ (78.1 kHz)
1	1	$f_X/2^7$ (39.1 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 clocks supplied to peripheral hardware. $f_X/2^{N+1}$ is stopped in HALT mode.**

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

2. f_X: Main system clock frequency

3. Figures in parentheses apply to operation with f_X = 5.0 MHz.

6.5 16-Bit Timer/Event Counter Operations

6.5.1 Interval timer operations

By setting bits 2 and 3 (TMC02 and TMC03) of the 16-bit timer mode control register (TMC0) to 1 and 1, they are operated as an interval timer. Interrupt requests are generated repeatedly using the count value set in 16-bit compare register (CR00) beforehand is used as the interval.

When the count value of the 16-bit timer register (TM0) matches the value set to CR00, counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM0) is generated. CR00 should be set to a value other than 0000H.

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

For the operation when the value of the compare register is changed while the timer/counter operates, refer to **6.6 16-Bit Timer/Event Counter Operating Precautions (3)**.

Figure 6-10. Interval Timer Configuration Diagram

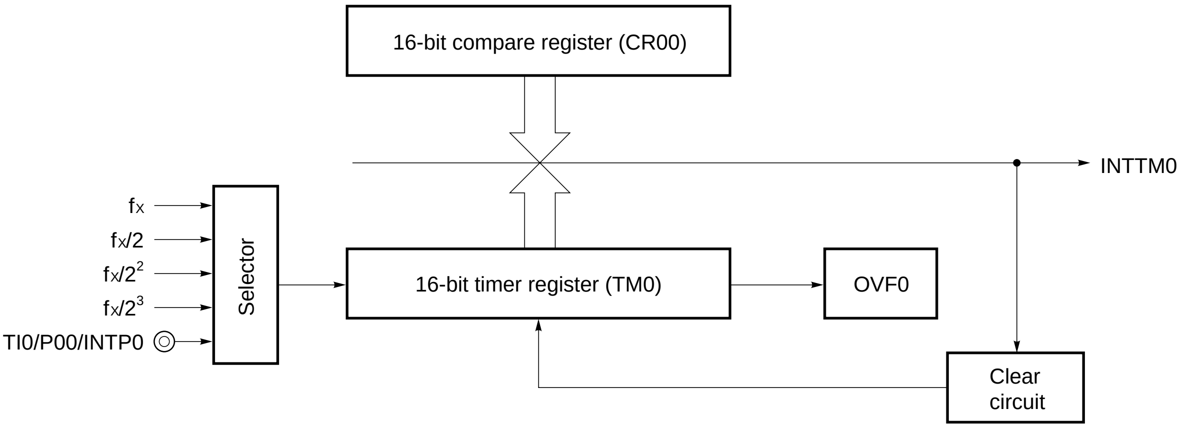
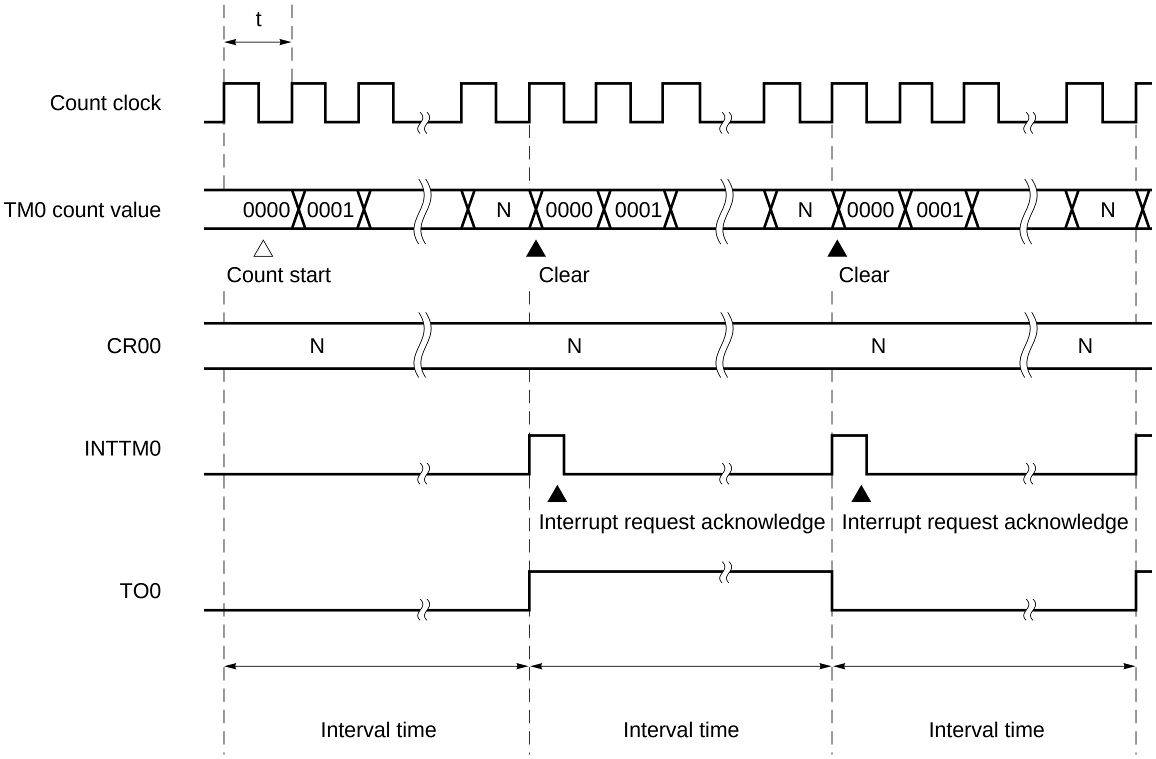


Figure 6-11. Interval Timer Operation Timings



Remark Interval time = (N + 1) × t ,where N = 0001H to FFFFH

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

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	0	1	2 × 1/fx (400 ns)	2^{16} × 1/fx (13.1 ms)	1/fx (200 ns)
0	1	0	2^{2} × 1/fx (800 ns)	2^{17} × 1/fx (26.2 ms)	2 × 1/fx (400 ns)
0	1	1	2^{3} × 1/fx (1.6 μs)	2^{18} × 1/fx (52.4 ms)	2^{2} × 1/fx (800 ns)
1	0	0	2^{4} × 1/fx (3.2 μs)	2^{19} × 1/fx (104.9 ms)	2^{3} × 1/fx (1.6 μs)
Other than above			Setting prohibited		

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

2. Figures in parentheses apply to operation with fx = 5.0 MHz

6.5.2 PWM output operations

By setting bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 1, 0, and 0, they are operated as PWM output. Pulses with the duty rate determined by the value set in 16-bit compare register (CR00) beforehand are output 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 with bit 1 (TOC01) of the 16-bit timer output control register (TOC0).

This PWM pulse has a 14-bit resolution. The pulse can be converted to an analog voltage by integrating it with an external low-pass filter (LPF). The PWM pulse has a combination of the basic cycle determined by $2^8/\phi$ and the sub-cycle determined by $2^{14}/\phi$ so that the time constant of the external LPF can be shortened. Count clock ϕ can be selected with bits 4 to 6 (TCL04 to TCL06) of the timer clock select register 0 (TCL0).

PWM output enable/disable can be selected with bit 0 (TOE0) of TOC0.

Cautions 1. PWM operation mode should be selected before setting CR00.

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

3. Do not select PWM operation mode for external clock input from the TI0/P00 pin.

By integrating 14-bit resolution PWM pulses with an external low-pass filter, they can be converted to an analog voltage and used for electronic tuning and D/A converter applications, etc.

The analog output voltage (V_{AN}) used for D/A conversion with the configuration shown in Figure 6-12 is as follows.

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

V_{REF} : External switching circuit reference voltage

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

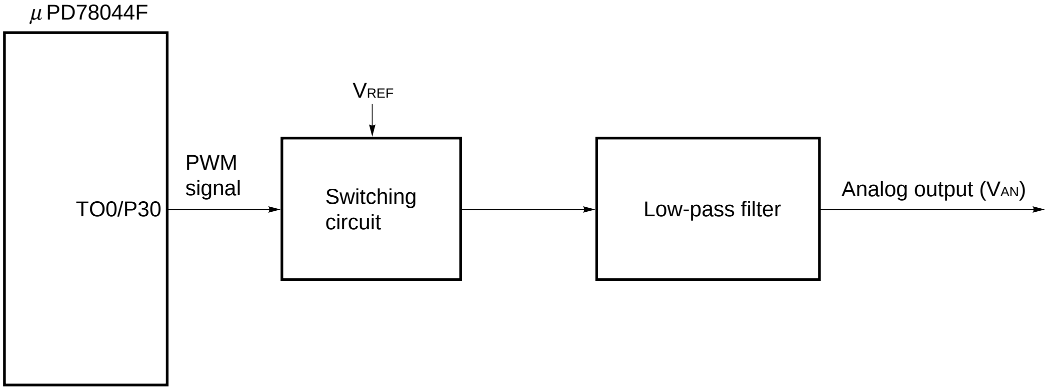
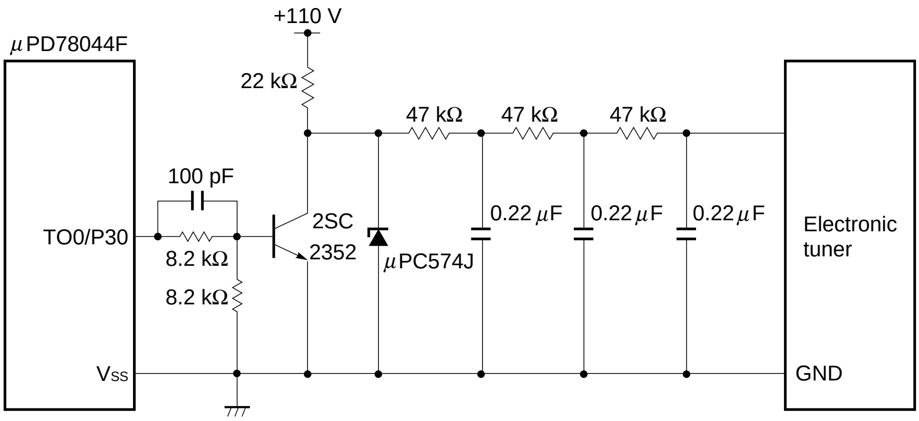


Figure 6-13 shows an example in which PWM output is converted to an analog voltage and used in a voltage synthesizer type TV tuner.

Figure 6-13. TV Tuner Application Circuit Example



6.5.3 Pulse width measurement operations

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

There are two measurement methods: measuring with the 16-bit timer register (TM0) used in free-running mode, and measuring by restarting the timer in synchronization with the valid edge of the signal input to the TI0/P00 pin.

(1) Pulse width measurement with free-running

When the TM0 is operated in the free-running mode, the edge specified by the external interrupt mode register (INTM0) is input, the value of TM0 is taken into the capture register (CR01) and an external interrupt request signal (INTP0) is set.

Any of three edge specifications can be selected–rising, falling, or both edges–by means of bits 2 and 3 (ES10 and ES11) of INTM0.

For valid edge detection, sampling is performed at the interval selected by means of the sampling clock selection register (SCS), and a capture operation is only performed when a valid level is detected twice, thus eliminating noise with a short pulse width.

Figure 6-14. Configuration Diagram for Pulse Width Measurement by Free-Running Counter

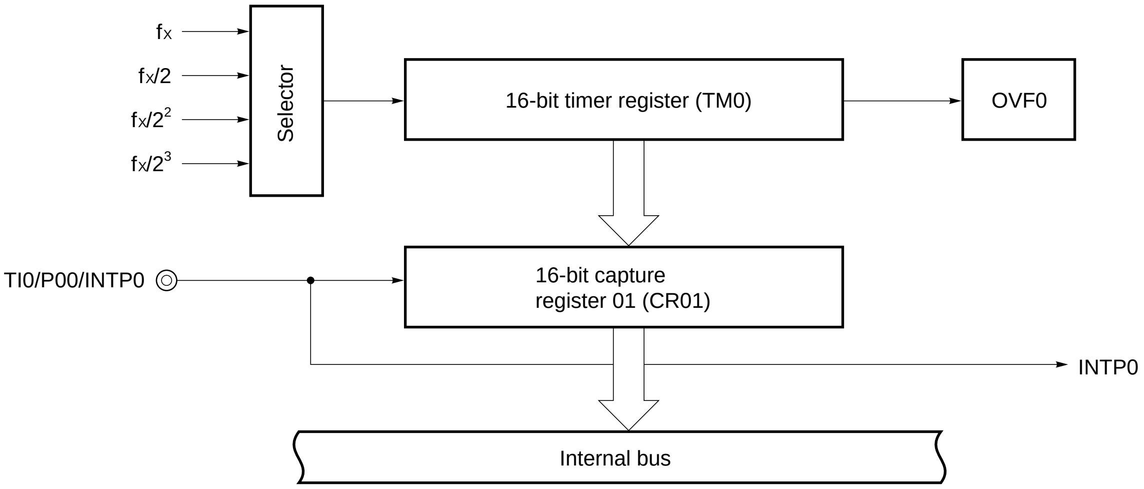
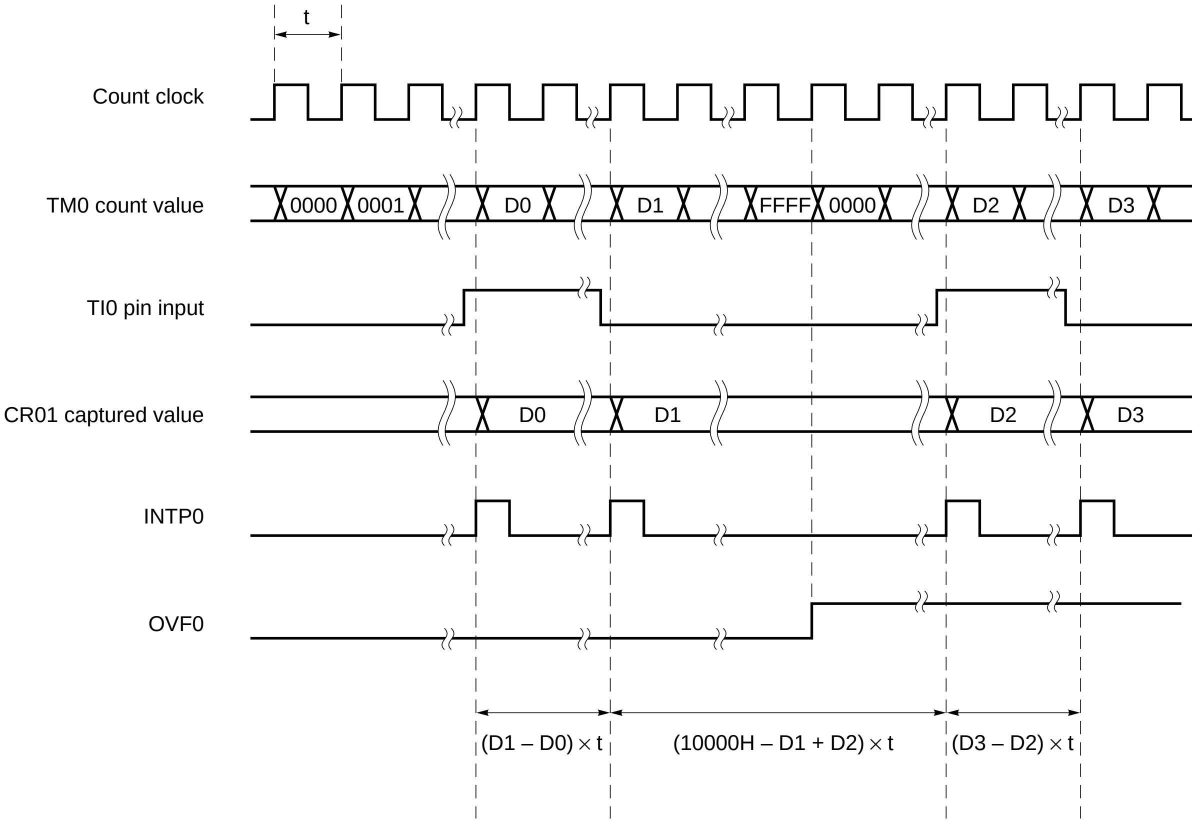

fX
fX/2
fX/2^2
fX/2^3
Selector
16-bit timer register (TM0)
OVF0
TI0/P00/INTP0
16-bit capture register 01 (CR01)
INTP0
Internal bus

Figure 6-15. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



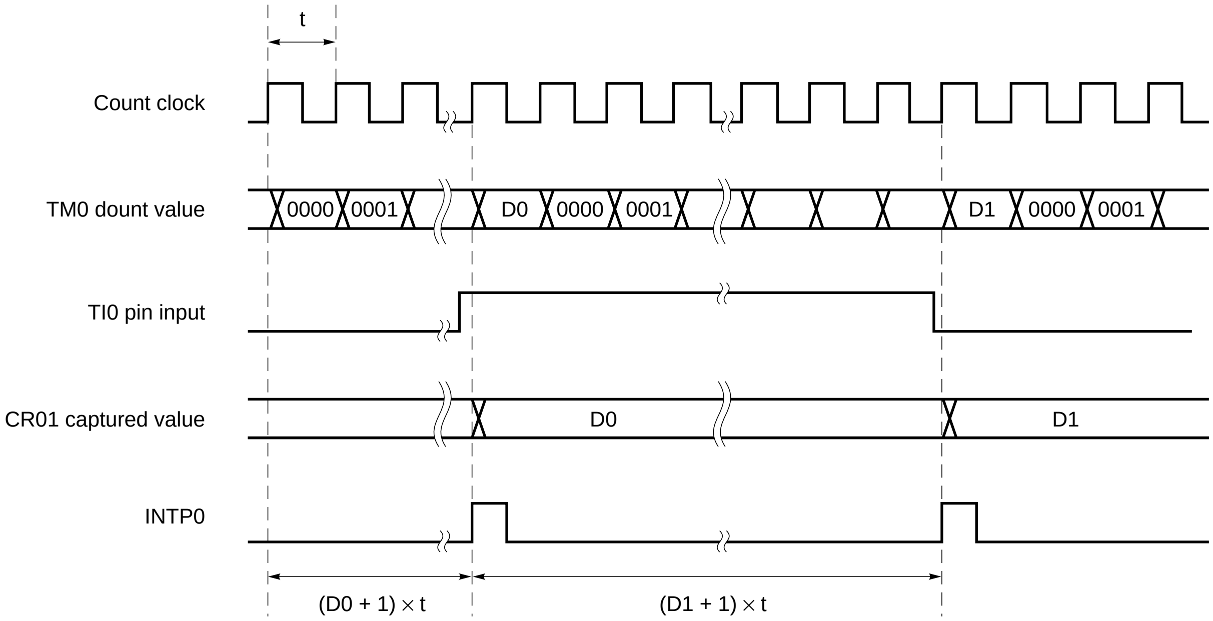
(2) Pulse width measurement by means of restart

When input of a valid edge to the TI0/P00 pin is detected, the count value of the 16-bit timer register (TM0) is taken into the 16-bit capture register (CR01), and then the pulse width of the signal input to the TI0/P00 pin is measured by clearing TM0 and restarting the count.

The edge specification can be selected from three types, rising, falling, and both edges by the external interrupt mode register (INTM0) bit 2 and bit 3 (ES10 and ES11).

In a valid edge detection, the sampling is performed by a cycle selected by the sampling clock selection register (SCS), and a capture operation is not performed before detecting valid levels twice allowing short pulse width noise to be eliminated.

Figure 6-16. Timing of Pulse Width Measurement Operation by Means of Restart (with Both Edges Specified)



6.5.4 External event counter operation

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

TM0 is incremented each time the valid edge specified with the external interrupt mode register (INTM0) is input.

When the TM0 counted value matches the 16-bit compare register (CR00) value, TM0 is cleared to 0 and the interrupt request signal (INTTM0) is generated.

Set a value other than 0000H to CR00 (1-pulse count operation cannot be performed).

The rising edge, the falling edge or both edges can be selected with bits 2 and 3 (ES10 and ES11) of INTM0.

Because operation is carried out only after the valid edge is detected twice by sampling at the cycle selected with the sampling clock select register (SCS), noise with short pulse widths can be removed.

Figure 6-17. External Event Counter Configuration Diagram

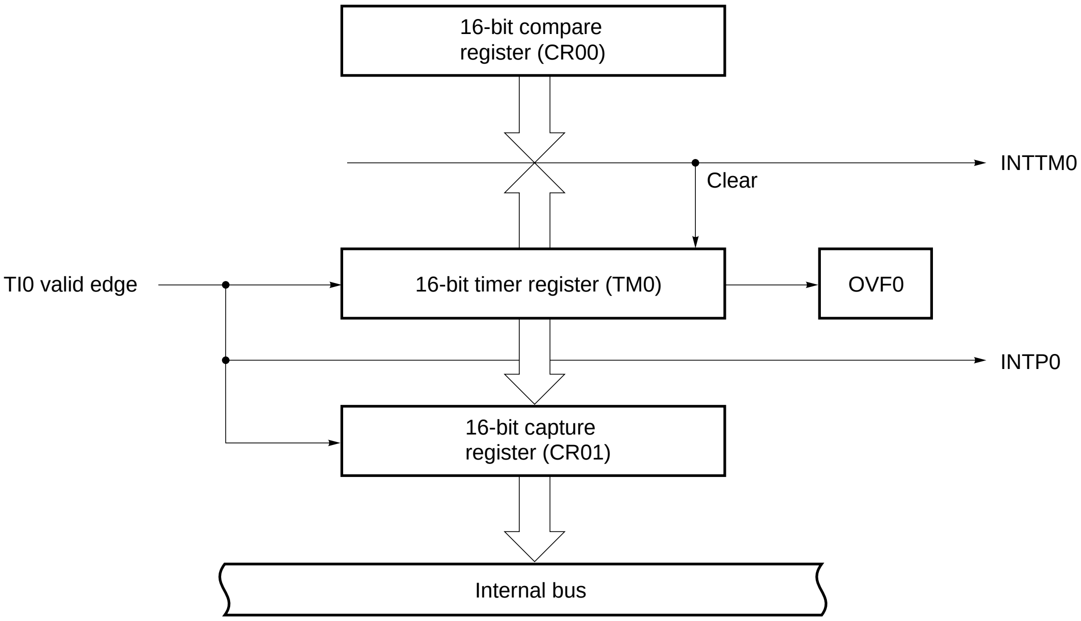
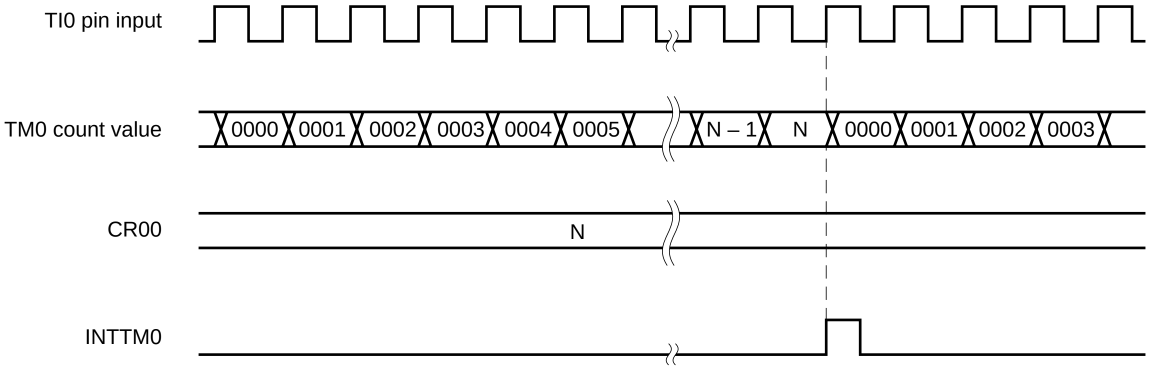


Figure 6-18 . External Event Counter Operation Timings (with Rising Edge Specified)



6.5.5 Square-wave output operation

The 16-bit timer/event counter operates as square wave output at intervals of the count value preset to the 16-bit compare register (CR00).

The TO0/P30 pin output status is reversed at intervals of the count value preset to CR00 by setting bit 0 (TOE0) and bit 1 (TOC01) of the 16-bit timer output control register (TOC0) to 1. This enables a square with any selected frequency to be output.

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

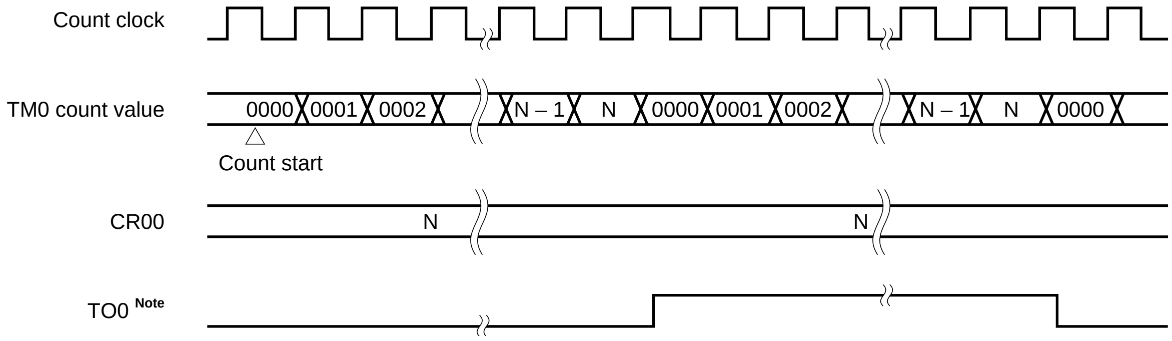
TCL06	TCL05	TCL04	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	2 × TI0 input cycle	2^{16} × TI0 input cycle	TI0 input edge cycle
0	0	1	2 × 1/f_X (400 ns)	2^{16} × 1/f_X (13.1 ms)	1/f_X (200 ns)
0	1	0	2^2 × 1/f_X (800 ns)	2^{17} × 1/f_X (26.2 ms)	2 × 1/f_X (400 ns)
0	1	1	2^3 × 1/f_X (1.6 μs)	2^{18} × 1/f_X (52.4 ms)	2^2 × 1/f_X (800 ns)
1	0	0	2^4 × 1/f_X (3.2 μs)	2^{19} × 1/f_X (104.9 ms)	2^3 × 1/f_X (1.6 μs)

Remarks 1. f_X: Main system clock frequency

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

3. Figures in parentheses apply to operation with f_X = 5.0 MHz

Figure 6-19. Square-Wave Output Operation Timings



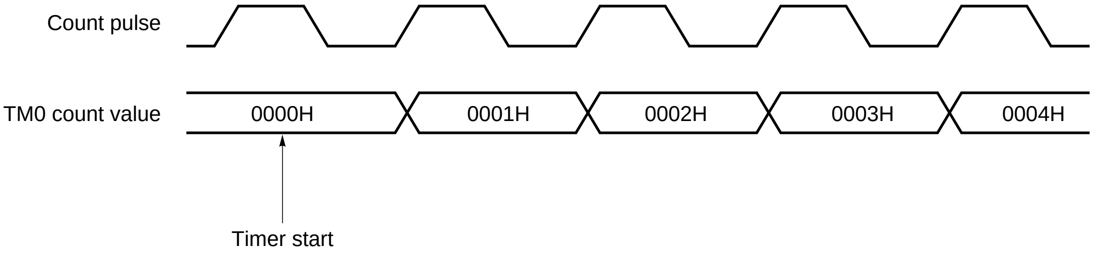
Note Initial value of TO0 output can be set with bits 2 and 3 (LVR0 and LVS0) of the 16-bit timer output control register (TOC0).

6.6 16-Bit Timer/Event Counter Operating Precautions

(1) Timer start errors

An error with a maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because the 16-bit timer register (TM0) is started asynchronously with the count pulse.

Figure 6-20. 16-Bit Timer Register Start Timings



(2) 16-bit compare register set

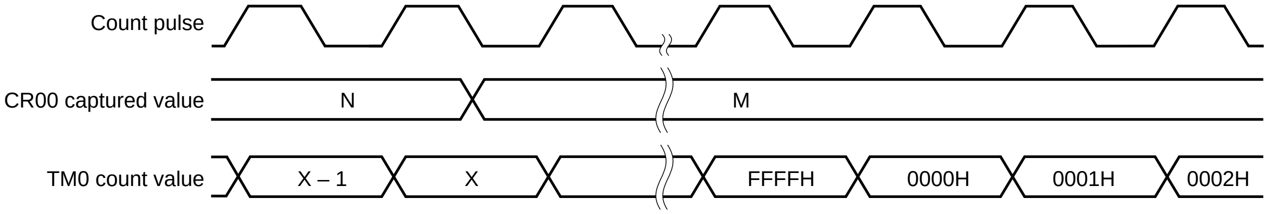
Set a value other than 0000H to the 16-bit compare register (CR00).

Thus, when using the 16-bit compare register as event counter, one-pulse count operation cannot be carried out.

(3) Operation after compare register change during timer count operation

If the value after the 16-bit compare register (CR00) is changed is smaller than that of the 16-bit timer register (TM0), TM0 continues counting, overflows and then restarts counting from 0. Thus, if the value after CR00 change (M) is smaller than that before change (N), it is necessary to restart the timer after changing CR00.

Figure 6-21. Timings after Change of Compare Register during Timer Count Operation

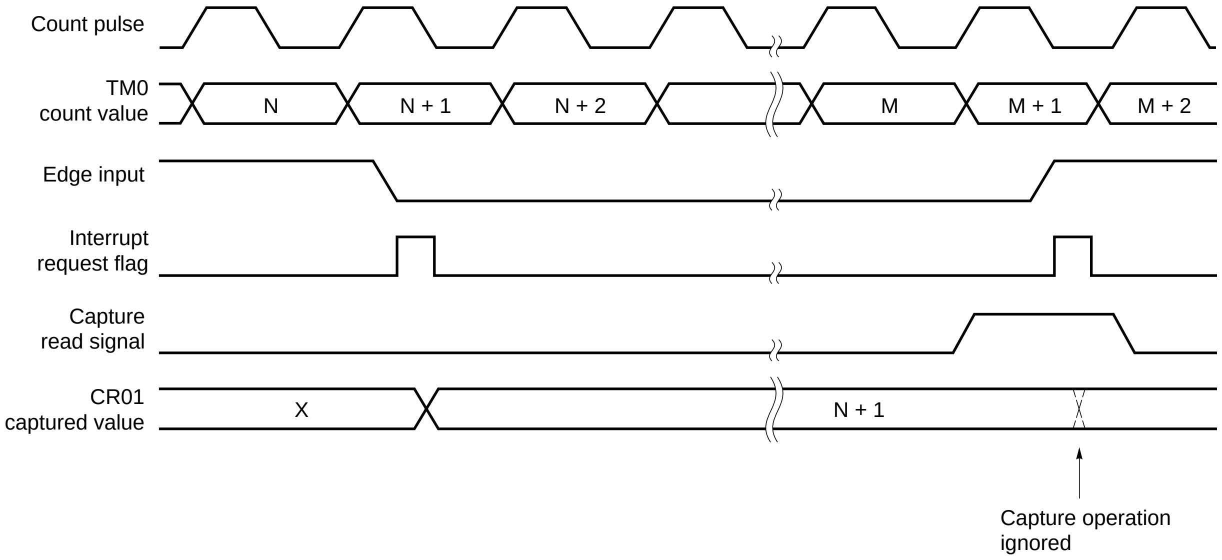


Remark N > X > M

(4) Capture register data retention timings

If the valid edge of the TI0/P00 pin is input during 16-bit capture register (CR01) read, CR01 holds data without carrying out the capture operation. However, the interrupt request flag (PIF0) is set upon detection of the valid edge.

Figure 6-22. Capture Register Data Retention Timings



(5) Valid edge set

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

★ **(6) OVF0 flag operation**

OVF0 flag is set to 1 in the following case.

The clear & start mode on match between TM0 and CR00 is selected.

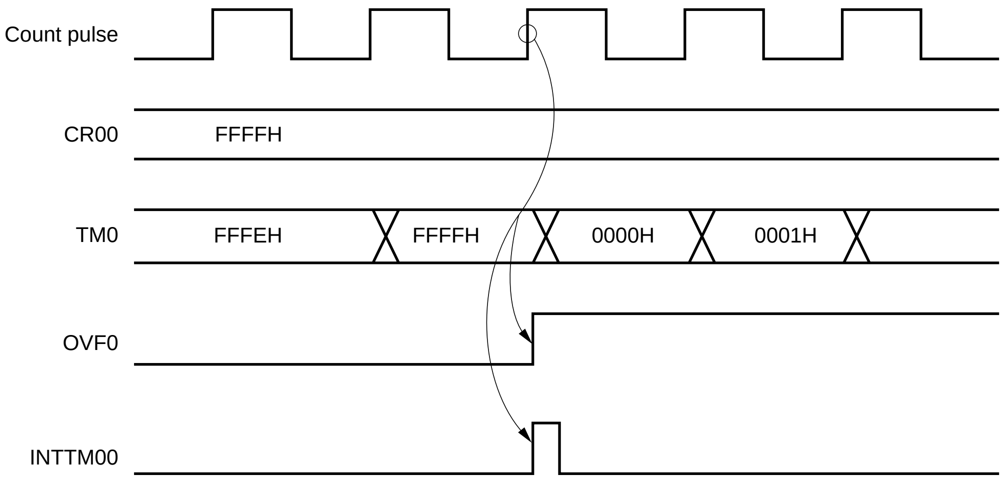
↓

CR00 is set to FFFFH.

↓

When TM0 is counted up from FFFFH to 0000H.

Figure 6-23. OVF0 Flag Operation Timings



CHAPTER 7 8-BIT TIMER/EVENT COUNTER

7.1 8-Bit Timer/Event Counter Functions

For the 8-bit timer/event counter incorporated in the μPD78044F Subseries, the following two modes are available.

- 8-bit timer/event counter mode : two-channel 8-bit timer/event counters to be used separately
- 16-bit timer/event counter mode : two-channel 8-bit timer/event counters to be used as 16-bit timer/event counter

7.1.1 8-bit timer/event counter mode

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

- Interval timer
- External event counter
- Square-wave output

(1) 8-bit interval timer

Interrupt requests are generated at the preset time intervals.

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

Minimum Interval Time	Maximum Interval Time	Resolution
2 × 1/f_X (400 ns)	2^9 × 1/f_X (102.4 μs)	2 × 1/f_X (400 ns)
2^2 × 1/f_X (800 ns)	2^{10} × 1/f_X (204.8 μs)	2^2 × 1/f_X (800 ns)
2^3 × 1/f_X (1.6 μs)	2^{11} × 1/f_X (409.6 μs)	2^3 × 1/f_X (1.6 μs)
2^4 × 1/f_X (3.2 μs)	2^{12} × 1/f_X (819.2 μs)	2^4 × 1/f_X (3.2 μs)
2^5 × 1/f_X (6.4 μs)	2^{13} × 1/f_X (1.64 ms)	2^5 × 1/f_X (6.4 μs)
2^6 × 1/f_X (12.8 μs)	2^{14} × 1/f_X (3.28 ms)	2^6 × 1/f_X (12.8 μs)
2^7 × 1/f_X (25.6 μs)	2^{15} × 1/f_X (6.55 ms)	2^7 × 1/f_X (25.6 μs)
2^8 × 1/f_X (51.2 μs)	2^{16} × 1/f_X (13.1 ms)	2^8 × 1/f_X (51.2 μs)
2^9 × 1/f_X (102.4 μs)	2^{17} × 1/f_X (26.2 ms)	2^9 × 1/f_X (102.4 μs)
2^{10} × 1/f_X (204.8 μs)	2^{18} × 1/f_X (52.4 ms)	2^{10} × 1/f_X (204.8 μs)
2^{12} × 1/f_X (819.2 μs)	2^{20} × 1/f_X (209.7 ms)	2^{12} × 1/f_X (819.2 μs)

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz.

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

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

Minimum Pulse Time	Maximum Pulse Time	Resolution
2 × 1/f_X (400 ns)	2^9 × 1/f_X (102.4 μs)	2 × 1/f_X (400 ns)
2^2 × 1/f_X (800 ns)	2^{10} × 1/f_X (204.8 μs)	2^2 × 1/f_X (800 ns)
2^3 × 1/f_X (1.6 μs)	2^{11} × 1/f_X (409.6 μs)	2^3 × 1/f_X (1.6 μs)
2^4 × 1/f_X (3.2 μs)	2^{12} × 1/f_X (819.2 μs)	2^4 × 1/f_X (3.2 μs)
2^5 × 1/f_X (6.4 μs)	2^{13} × 1/f_X (1.64 ms)	2^5 × 1/f_X (6.4 μs)
2^6 × 1/f_X (12.8 μs)	2^{14} × 1/f_X (3.28 ms)	2^6 × 1/f_X (12.8 μs)
2^7 × 1/f_X (25.6 μs)	2^{15} × 1/f_X (6.55 ms)	2^7 × 1/f_X (25.6 μs)
2^8 × 1/f_X (51.2 μs)	2^{16} × 1/f_X (13.1 ms)	2^8 × 1/f_X (51.2 μs)
2^9 × 1/f_X (102.4 μs)	2^{17} × 1/f_X (26.2 ms)	2^9 × 1/f_X (102.4 μs)
2^{10} × 1/f_X (204.8 μs)	2^{18} × 1/f_X (52.4 ms)	2^{10} × 1/f_X (204.8 μs)
2^{12} × 1/f_X (819.2 μs)	2^{20} × 1/f_X (209.7 ms)	2^{12} × 1/f_X (819.2 μs)

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz.

7.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

Interrupt requests can be generated at the preset time intervals.

Table 7-3. Interval Times when 8-Bit Timer/Event Counter is Used as 16-Bit Timer/Event Counter

Minimum Interval Time	Maximum Interval Time	Resolution
2 × 1/f_X (400 ns)	2^{17} × 1/f_X (26.2 ms)	2 × 1/f_X (400 ns)
2^2 × 1/f_X (800 ns)	2^{18} × 1/f_X (52.4 ms)	2^2 × 1/f_X (800 ns)
2^3 × 1/f_X (1.6 μs)	2^{19} × 1/f_X (104.9 ms)	2^3 × 1/f_X (1.6 μs)
2^4 × 1/f_X (3.2 μs)	2^{20} × 1/f_X (209.7 ms)	2^4 × 1/f_X (3.2 μs)
2^5 × 1/f_X (6.4 μs)	2^{21} × 1/f_X (419.4 ms)	2^5 × 1/f_X (6.4 μs)
2^6 × 1/f_X (12.8 μs)	2^{22} × 1/f_X (838.9 ms)	2^6 × 1/f_X (12.8 μs)
2^7 × 1/f_X (25.6 μs)	2^{23} × 1/f_X (1.7 s)	2^7 × 1/f_X (25.6 μs)
2^8 × 1/f_X (51.2 μs)	2^{24} × 1/f_X (3.4 s)	2^8 × 1/f_X (51.2 μs)
2^9 × 1/f_X (102.4 μs)	2^{25} × 1/f_X (6.7 s)	2^9 × 1/f_X (102.4 μs)
2^{10} × 1/f_X (204.8 μs)	2^{26} × 1/f_X (13.4 s)	2^{10} × 1/f_X (204.8 μs)
2^{12} × 1/f_X (819.2 μs)	2^{28} × 1/f_X (53.7 s)	2^{12} × 1/f_X (819.2 μs)

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz.

(2) External event counter

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

(3) Square-wave output

A square wave with any selected frequency can be output.

Table 7-4. Square-Wave Output Ranges when 8-Bit Timer/Event Counter is Used as 16-Bit Timer/Event Counter

Minimum Pulse Time	Maximum Pulse Time	Resolution
2 × 1/f_X (400 ns)	2^{17} × 1/f_X (26.2 ms)	2 × 1/f_X (400 ns)
2^2 × 1/f_X (800 ns)	2^{18} × 1/f_X (52.4 ms)	2^2 × 1/f_X (800 ns)
2^3 × 1/f_X (1.6 μs)	2^{19} × 1/f_X (104.9 ms)	2^3 × 1/f_X (1.6 μs)
2^4 × 1/f_X (3.2 μs)	2^{20} × 1/f_X (209.7 ms)	2^4 × 1/f_X (3.2 μs)
2^5 × 1/f_X (6.4 μs)	2^{21} × 1/f_X (419.4 ms)	2^5 × 1/f_X (6.4 μs)
2^6 × 1/f_X (12.8 μs)	2^{22} × 1/f_X (838.9 ms)	2^6 × 1/f_X (12.8 μs)
2^7 × 1/f_X (25.6 μs)	2^{23} × 1/f_X (1.7 s)	2^7 × 1/f_X (25.6 μs)
2^8 × 1/f_X (51.2 μs)	2^{24} × 1/f_X (3.4 s)	2^8 × 1/f_X (51.2 μs)
2^9 × 1/f_X (102.4 μs)	2^{25} × 1/f_X (6.7 s)	2^9 × 1/f_X (102.4 μs)
2^{10} × 1/f_X (204.8 μs)	2^{26} × 1/f_X (13.4 s)	2^{10} × 1/f_X (204.8 μs)
2^{12} × 1/f_X (819.2 μs)	2^{28} × 1/f_X (53.7 s)	2^{12} × 1/f_X (819.2 μs)

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz.

7.2 8-Bit Timer/Event Counter Configuration

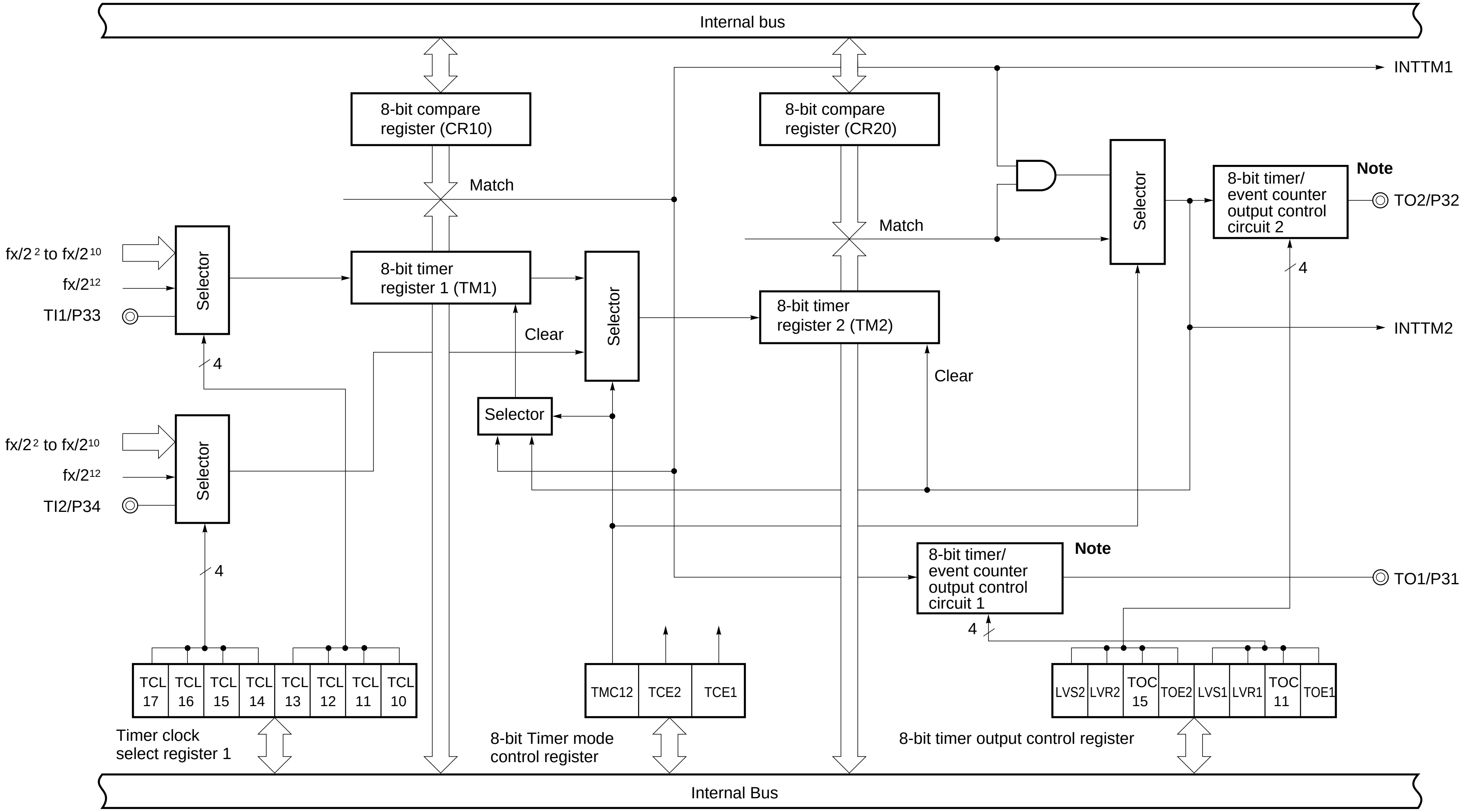
The 8-bit timer/event counter consists of the following hardware.

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

Item	Configuration
Timer register	8-bit × 2 (TM1, TM2)
Register	8-bit compare register: 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

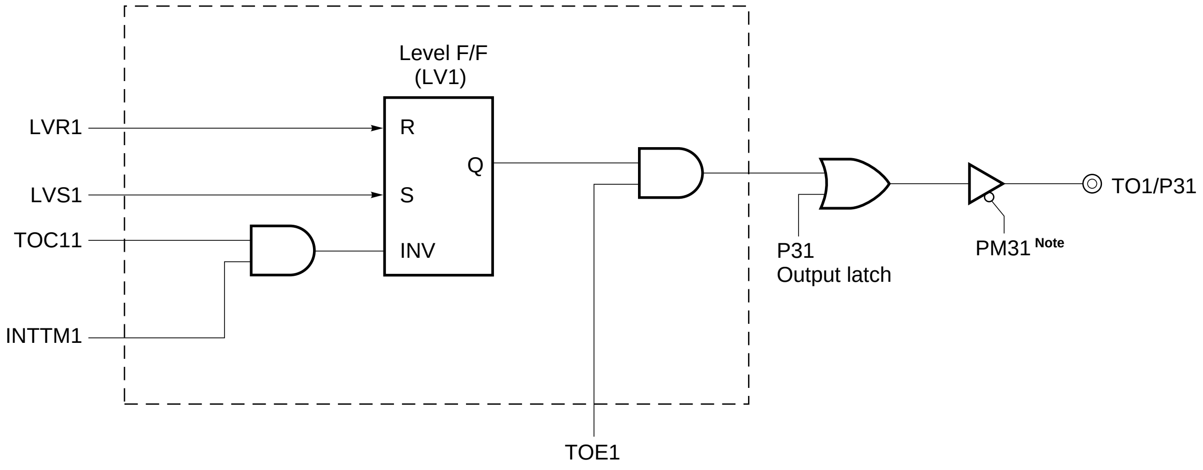
Note Refer to **Figure 4-7 P30 to P37 Block Diagram**.

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


Internal bus
INTTM1
8-bit compare register (CR10)
8-bit compare register (CR20)
Match
Match
Selector
8-bit timer/ event counter output control circuit 2
Note
TO2/P32
fx/2² to fx/2¹⁰
fx/2¹²
TI1/P33
Selector
8-bit timer register 1 (TM1)
Selector
8-bit timer register 2 (TM2)
4
INTTM2
Clear
Clear
4
Selector
fx/2² to fx/2¹⁰
fx/2¹²
TI2/P34
Selector
4
8-bit timer/ event counter output control circuit 1
Note
TO1/P31
4
TCL 17
TCL 16
TCL 15
TCL 14
TCL 13
TCL 12
TCL 11
TCL 10
TMC12
TCE2
TCE1
LVS2
LVR2
TOC 15
TOE2
LVS1
LVR1
TOC 11
TOE1
Timer clock select register 1
8-bit Timer mode control register
8-bit timer output control register
Internal Bus

Note Refer to **Figures 7-2** and **7-3** for details of 8-bit timer/event counter output control circuits 1 and 2, respectively.

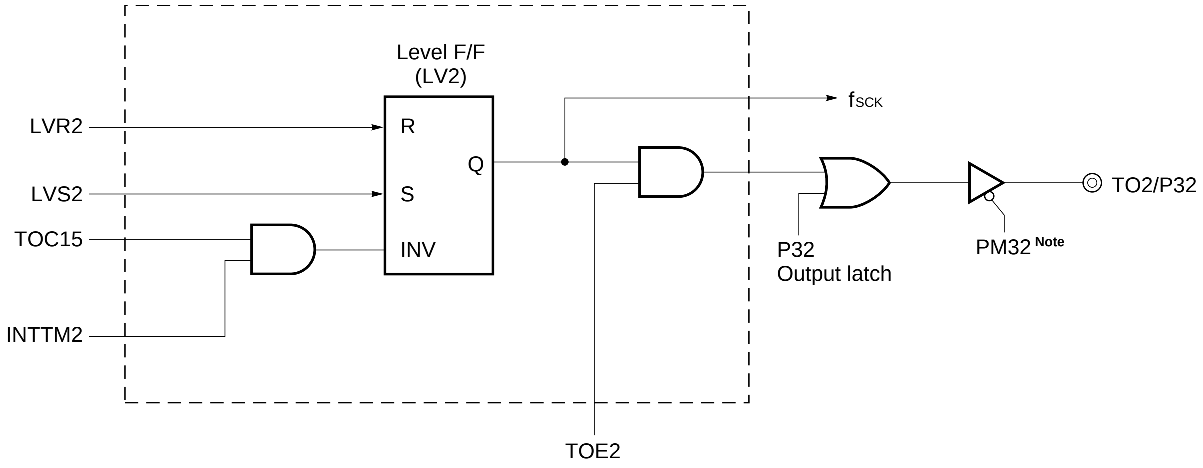
Figure 7-2. 8-Bit Timer/Event Counter Output Control Circuit 1 Block Diagram



Note Bit 1 of port mode register 3 (PM3)

Remark The section in the broken line is an output control circuit.

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



Note Bit 2 of port mode register 3 (PM3)

Remarks 1. The section in the broken line is an output control circuit.

2. f_{SCK}: Serial clock frequency

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

This is an 8-bit register to compare the value set to CR10 to the 8-bit timer register 1 (TM1) count value, and the value set to CR20 to the 8-bit timer register 2 (TM2) count value, and, if they match, generates an interrupt request (INTTM1 and INTTM2, respectively).

CR10 and CR20 are set with an 8-bit memory manipulation instruction. They cannot be set with a 16-bit memory manipulation instruction. When the compare register is used as 8-bit timer/event counter, the 00H to FFH values can be set. When the compare register is used as 16-bit timer/event counter, the 0000H to FFFFH values can be set.

$\overline{\text{RESET}}$ input makes CR10 and CR20 undefined.

Cautions 1. When using the compare register as 16-bit timer/event counter, be sure to set data after stopping timer operation.

2. If the values after CR10 and CR20 are changed are smaller than those of 8-bit timer registers (TM1 and TM2), TM1 and TM2 continue counting, overflow and then restart counting from 0. Thus, if the value after CR10 and CR20 change is smaller than that before change, it is necessary to restart the timer after changing CR10 and CR20.

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

These are 8-bit registers to count count pulses.

When TM1 and TM2 are used in the "two 8-bit timer channels" mode, they should be read with an 8-bit memory manipulation instruction. When TM1 and TM2 are used in the "one 16-bit timer channel" mode, the 16-bit timer (TMS) should be read with a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TM1 and TM2 to 00H.

7.3 8-Bit Timer/Event Counter Control Registers

The following four types of registers are used to control the 8-bit timer/event counter.

- 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 count clocks of 8-bit timer registers 1 and 2.

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

$\overline{\text{RESET}}$ input clears TCL1 to 00H.

Figure 7-4. Timer Clock Select Register 1 Format

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

TCL13	TCL12	TCL11	TCL10	8-bit Timer Register 1 Count Clock Selection
0	0	0	0	TI1 falling edge
0	0	0	1	TI1 rising edge
0	1	0	1	$f_X/2$ (2.5 MHz)
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_X/2^{12}$ (1.2 kHz)
Other than above				Setting prohibited

TCL17	TCL16	TCL15	TCL14	8-bit Timer Register 2 Count Clock Selection
0	0	0	0	TI2 falling edge
0	0	0	1	TI2 rising edge
0	1	0	1	$f_X/2$ (2.5 MHz)
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
1	1	1	0	$f_X/2^{10}$ (4.9 kHz)
1	1	1	1	$f_X/2^{12}$ (1.2 kHz)
Other than above				Setting prohibited

Caution If TCL1 is to be rewritten in data other than identical data, the timer operation must be stopped first.

Remarks 1. f_X: Main system clock frequency

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

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

4. Figures in parentheses apply to operation with f_X = 5.0 MHz.

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

This register enables/stops operation of 8-bit timer registers 1 and 2 and sets the operating mode of 8-bit timer register 2.

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

$\overline{\text{RESET}}$ input clears TMC1 to 00H.

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

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

TCE1	8-Bit Timer Register 1 Operation Control
0	Operation stop (TM1 clear to 0)
1	Operation enable

TCE2	8-Bit Timer Register 2 Operation Control
0	Operation stop (TM2 clear to 0)
1	Operation enable

TMC12	Operating Mode Selection
0	8-Bit timer register × 2-channel mode (TM1, TM2)
1	16-Bit timer register × 1-channel mode (TMS)

Cautions 1. Switch the operating mode after stopping timer operation.

2. When used as 16-bit timer register, TCE1 should be used for operation enable/stop.

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

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

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

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

$\overline{\text{RESET}}$ input clears TOC1 to 00H.

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

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

TOE1	8-Bit Timer/Event Counter 1 Output Control
0	Output disable (port mode)
1	Output enable

TOC11	8-Bit Timer/Event Counter 1 Timer Output F/F Control
0	Inverted operation disable
1	Inverted operation enable

LVS1	LVR1	8-Bit Timer/Event Counter 1 Timer Output F/F Status Set
0	0	Unchanged
0	1	Timer output F/F reset (to 0)
1	0	Timer output F/F set (to 1)
1	1	Setting prohibited

TOE2	8-Bit Timer/Event Counter 2 Output Control
0	Output disable (port mode)
1	Output enable

TOC15	8-Bit Timer/Event Counter 2 Timer Output F/F Control
0	Inverted operation disable
1	Inverted operation enable

LVS2	LVR2	8-Bit Timer/Event Counter 2 Timer Output F/F Status Set
0	0	Unchanged
0	1	Timer output F/F reset (to 0)
1	0	Timer output F/F set (to 1)
1	1	Setting prohibited

Cautions 1. Be sure to set TOC1 after stopping timer operation.

2. After data setting, 0 can be read from LVS1, LVS2, LVR1 and LVR2.

(4) Port mode register 3 (PM3)

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

When using the P31/TO1 and P32/TO2 pins for timer output, set PM31, PM32, and output latches of P31 and P32 to 0.

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

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Figure 7-7. Port Mode Register 3 Format

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

PM3n	P3n Pin Input/Output Mode Selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

7.4 8-Bit Timer/Event Counter Operations

7.4.1 8-bit timer/event counter mode

(1) Interval timer operations

The 8-bit timer/event counter operates as interval timer which generates interrupt requests repeatedly at intervals of the count value preset to 8-bit compare registers (CR10 and CR20).

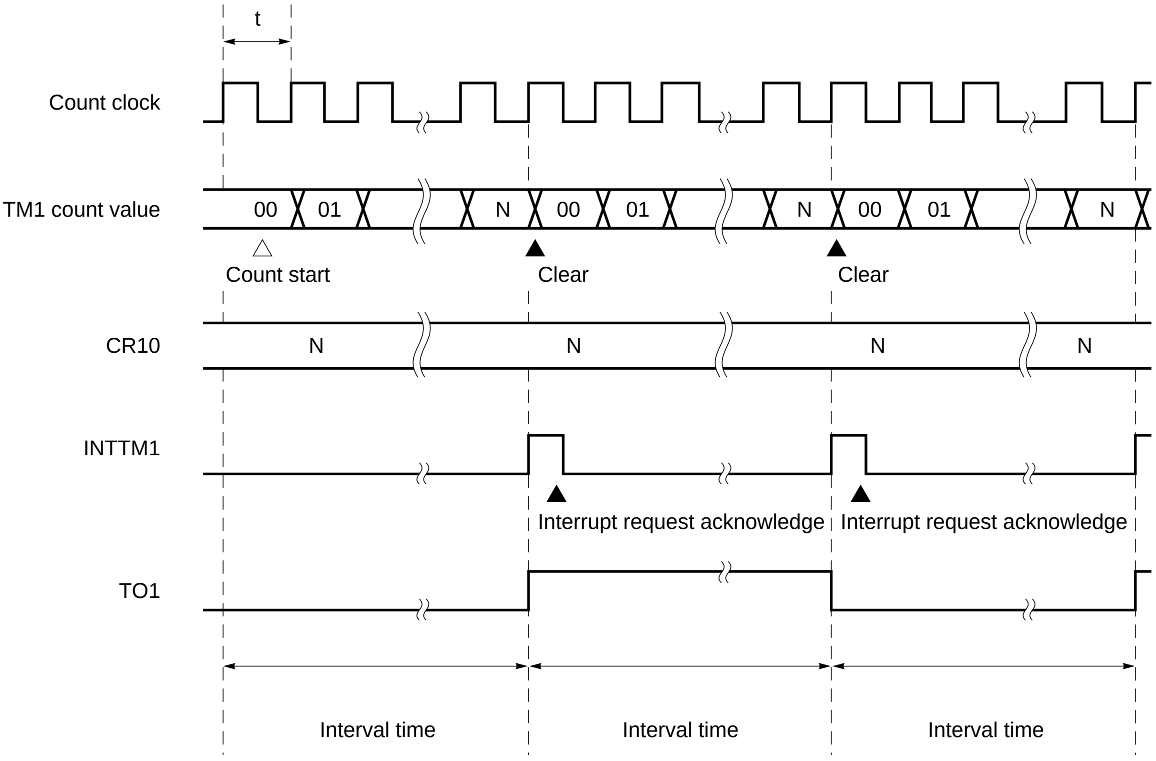
When the count values of the 8-bit timer registers 1 and 2 (TM1 and TM2) match the values set to CR10 and CR20, counting continues with the TM1 and TM2 values cleared to 0 and the interrupt request signals (INTTM1 and INTTM2) are generated.

Count clock of TM1 can be selected with bits 0 to 3 (TCL10 to TCL13) of the timer clock select register 1 (TCL1).

Count clock of TM2 can be selected with bits 4 to 7 (TCL14 to TCL17) of the timer clock select register 1 (TCL1).

For the operation when the value of the compare register is changed while the timer/counter operates, refer to **7.5 8-Bit Timer/Event Counter Operating Precautions (3)**.

Figure 7-8. Interval Timer Operation Timings



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

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

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	0	1	2 × 1/f_X (400 ns)	2^9 × 1/f_X (102.4 μs)	2 × 1/f_X (400 ns)
0	1	1	0	2^2 × 1/f_X (800 ns)	2^{10} × 1/f_X (204.8 μs)	2^2 × 1/f_X (800 ns)
0	1	1	1	2^3 × 1/f_X (1.6 μs)	2^{11} × 1/f_X (409.6 μs)	2^3 × 1/f_X (1.6 μs)
1	0	0	0	2^4 × 1/f_X (3.2 μs)	2^{12} × 1/f_X (819.2 μs)	2^4 × 1/f_X (3.2 μs)
1	0	0	1	2^5 × 1/f_X (6.4 μs)	2^{13} × 1/f_X (1.64 ms)	2^5 × 1/f_X (6.4 μs)
1	0	1	0	2^6 × 1/f_X (12.8 μs)	2^{14} × 1/f_X (3.28 ms)	2^6 × 1/f_X (12.8 μs)
1	0	1	1	2^7 × 1/f_X (25.6 μs)	2^{15} × 1/f_X (6.55 ms)	2^7 × 1/f_X (25.6 μs)
1	1	0	0	2^8 × 1/f_X (51.2 μs)	2^{16} × 1/f_X (13.1 ms)	2^8 × 1/f_X (51.2 μs)
1	1	0	1	2^9 × 1/f_X (102.4 μs)	2^{17} × 1/f_X (26.2 ms)	2^9 × 1/f_X (102.4 μs)
1	1	1	0	2^{10} × 1/f_X (204.8 μs)	2^{18} × 1/f_X (52.4 ms)	2^{10} × 1/f_X (204.8 μs)
1	1	1	1	2^{12} × 1/f_X (819.2 μs)	2^{20} × 1/f_X (209.7 ms)	2^{12} × 1/f_X (819.2 μs)
Other than above				Setting prohibited		

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz.

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

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	0	1	2 × 1/f_X (400 ns)	2^9 × 1/f_X (102.4 μs)	2 × 1/f_X (400 ns)
0	1	1	0	2^2 × 1/f_X (800 ns)	2^{10} × 1/f_X (204.8 μs)	2^2 × 1/f_X (800 ns)
0	1	1	1	2^3 × 1/f_X (1.6 μs)	2^{11} × 1/f_X (409.6 μs)	2^3 × 1/f_X (1.6 μs)
1	0	0	0	2^4 × 1/f_X (3.2 μs)	2^{12} × 1/f_X (819.2 μs)	2^4 × 1/f_X (3.2 μs)
1	0	0	1	2^5 × 1/f_X (6.4 μs)	2^{13} × 1/f_X (1.64 ms)	2^5 × 1/f_X (6.4 μs)
1	0	1	0	2^6 × 1/f_X (12.8 μs)	2^{14} × 1/f_X (3.28 ms)	2^6 × 1/f_X (12.8 μs)
1	0	1	1	2^7 × 1/f_X (25.6 μs)	2^{15} × 1/f_X (6.55 ms)	2^7 × 1/f_X (25.6 μs)
1	1	0	0	2^8 × 1/f_X (51.2 μs)	2^{16} × 1/f_X (13.1 ms)	2^8 × 1/f_X (51.2 μs)
1	1	0	1	2^9 × 1/f_X (102.4 μs)	2^{17} × 1/f_X (26.2 ms)	2^9 × 1/f_X (102.4 μs)
1	1	1	0	2^{10} × 1/f_X (204.8 μs)	2^{18} × 1/f_X (52.4 ms)	2^{10} × 1/f_X (204.8 μs)
1	1	1	1	2^{12} × 1/f_X (819.2 μs)	2^{20} × 1/f_X (209.7 ms)	2^{12} × 1/f_X (819.2 μs)
Other than above				Setting prohibited		

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz.

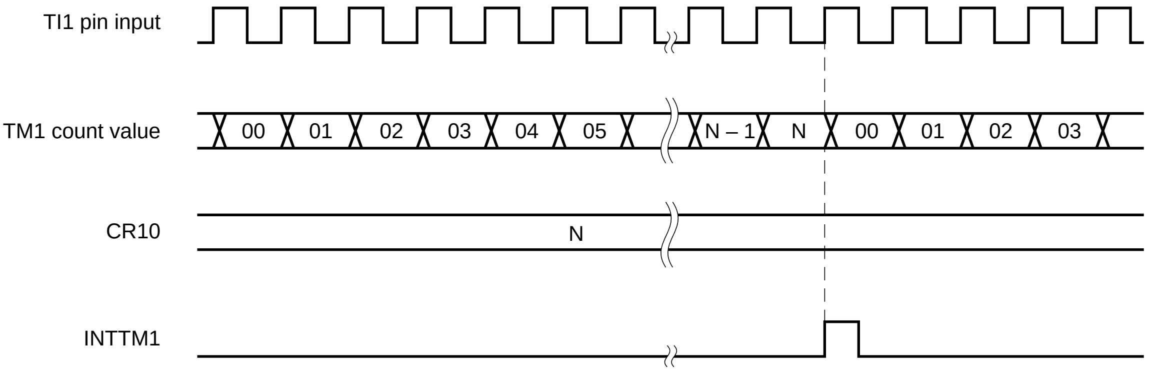
(2) External event counter operation

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

TM1 and TM2 are incremented each time the valid edge specified with the timer clock select register 1 (TCL1) is input. Either the rising or falling edge can be selected.

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

Figure 7-9. External Event Counter Operation Timings (with Rising Edge Specified)



Remark N = 00H to FFH

(3) Square-wave output operation

The 8-bit timer/event counter operates as square wave output at intervals of the value preset to 8-bit compare registers (CR10 and CR20).

The TO1/P31 or TO2/P32 pin output status is reversed at intervals of the count value preset to CR10 or CR20 by setting bit 0 (TOE1) or bit 4 (TOE2) of the 8-bit timer output control register (TOC1) to 1.

This enables a square wave with any selected frequency to be output.

Table 7-8. 8-Bit Timer/Event Counter Square-Wave Output Ranges

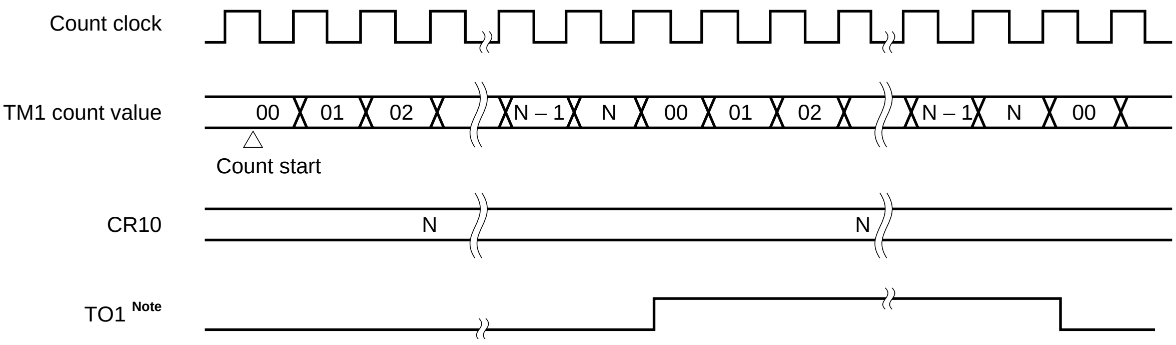
TCL13	TCL12	TCL11	TCL10	Minimum Pulse Time	Maximum Pulse Time	Resolution
0	1	0	1	$2 \times 1/f_X$ (400 ns)	$2^9 \times 1/f_X$ (102.4 μs)	$2 \times 1/f_X$ (400 ns)
0	1	1	0	$2^2 \times 1/f_X$ (800 ns)	$2^{10} \times 1/f_X$ (204.8 μs)	$2^2 \times 1/f_X$ (800 ns)
0	1	1	1	$2^3 \times 1/f_X$ (1.6 μs)	$2^{11} \times 1/f_X$ (409.6 μs)	$2^3 \times 1/f_X$ (1.6 μs)
1	0	0	0	$2^4 \times 1/f_X$ (3.2 μs)	$2^{12} \times 1/f_X$ (819.2 μs)	$2^4 \times 1/f_X$ (3.2 μs)
1	0	0	1	$2^5 \times 1/f_X$ (6.4 μs)	$2^{13} \times 1/f_X$ (1.64 ms)	$2^5 \times 1/f_X$ (6.4 μs)
1	0	1	0	$2^6 \times 1/f_X$ (12.8 μs)	$2^{14} \times 1/f_X$ (3.28 ms)	$2^6 \times 1/f_X$ (12.8 μs)
1	0	1	1	$2^7 \times 1/f_X$ (25.6 μs)	$2^{15} \times 1/f_X$ (6.55 ms)	$2^7 \times 1/f_X$ (25.6 μs)
1	1	0	0	$2^8 \times 1/f_X$ (51.2 μs)	$2^{16} \times 1/f_X$ (13.1 ms)	$2^8 \times 1/f_X$ (51.2 μs)
1	1	0	1	$2^9 \times 1/f_X$ (102.4 μs)	$2^{17} \times 1/f_X$ (26.2 ms)	$2^9 \times 1/f_X$ (102.4 μs)
1	1	1	0	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{18} \times 1/f_X$ (52.4 ms)	$2^{10} \times 1/f_X$ (204.8 μs)
1	1	1	1	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{20} \times 1/f_X$ (209.7 ms)	$2^{12} \times 1/f_X$ (819.2 μs)

Remarks 1. f_X: Main system clock frequency

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

3. Values in parentheses when operated at f_X = 5.0 MHz.

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



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

7.4.2 16-bit timer/event counter mode

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

In this mode, the count clock is selected with bits 0 to 3 (TCL10 to TCL13) of the time clock select register (TCL1). The overflow signal of the 8-bit timer/event counter 1 (TM1) is used as the count clock of the 8-bit timer/counter 2 (TM2). Count operation enable/disable in this mode is selected with bit 0 (TCE1) of TMC1.

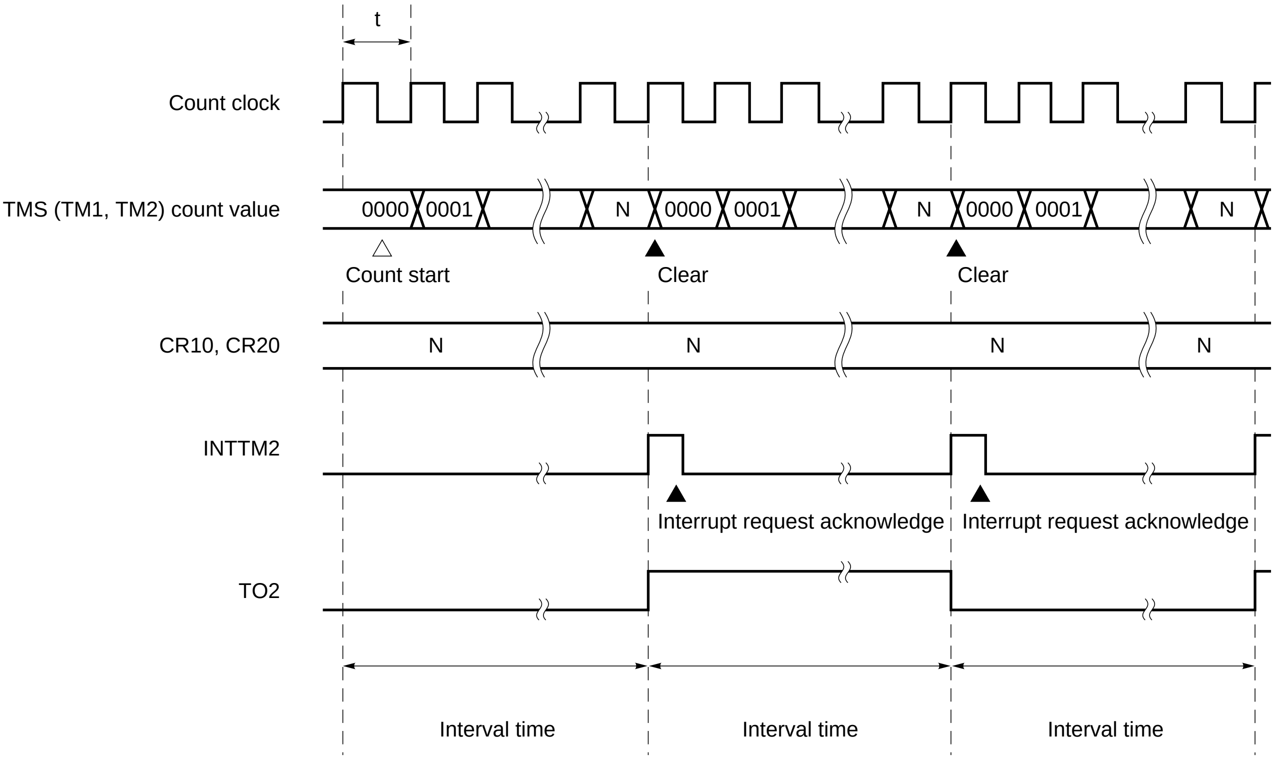
★ **(1) Interval timer operations**

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

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

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

Figure 7-11. Interval Timer Operation Timings



Remark Interval time = (N + 1) × t , where N = 0000H to FFFFH

Caution Even if the 16-bit timer/event counter mode is used, when the TM1 count value matches the CR10 value, interrupt request (INTTM1) is generated and the F/F of 8-bit timer/event counter output control circuit 1 is inverted. Thus, when using 8-bit timer/event counter as 16-bit interval timer, set the INTTM1 mask flag TMMK1 to 1 to disable INTTM1 acknowledgment.

When reading 16-bit timer (TMS) count value, use the 16-bit memory manipulation instruction.

Table 7-9. Interval Times when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 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	0	1	$2 \times 1/f_X$ (400 ns)	$2^{9} \times 1/f_X$ (26.2 ms)	$2 \times 1/f_X$ (400 ns)
0	1	1	0	$2^{2} \times 1/f_X$ (800 ns)	$2^{10} \times 1/f_X$ (52.4 ms)	$2^{2} \times 1/f_X$ (800 ns)
0	1	1	1	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{11} \times 1/f_X$ (104.9 ms)	$2^{3} \times 1/f_X$ (1.6 μs)
1	0	0	0	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{12} \times 1/f_X$ (209.7 ms)	$2^{4} \times 1/f_X$ (3.2 μs)
1	0	0	1	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{13} \times 1/f_X$ (419.4 ms)	$2^{5} \times 1/f_X$ (6.4 μs)
1	0	1	0	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{14} \times 1/f_X$ (838.9 ms)	$2^{6} \times 1/f_X$ (12.8 μs)
1	0	1	1	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{15} \times 1/f_X$ (1.7 s)	$2^{7} \times 1/f_X$ (25.6 μs)
1	1	0	0	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{16} \times 1/f_X$ (3.4 s)	$2^{8} \times 1/f_X$ (51.2 μs)
1	1	0	1	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{17} \times 1/f_X$ (6.7 s)	$2^{9} \times 1/f_X$ (102.4 μs)
1	1	1	0	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{18} \times 1/f_X$ (13.4 s)	$2^{10} \times 1/f_X$ (204.8 μs)
1	1	1	1	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{20} \times 1/f_X$ (53.7 s)	$2^{12} \times 1/f_X$ (819.2 μs)
Other than above				Setting prohibited		

Remarks 1. f_X: Main system clock frequency

2. Values in parentheses when operated at f_X = 5.0 MHz.

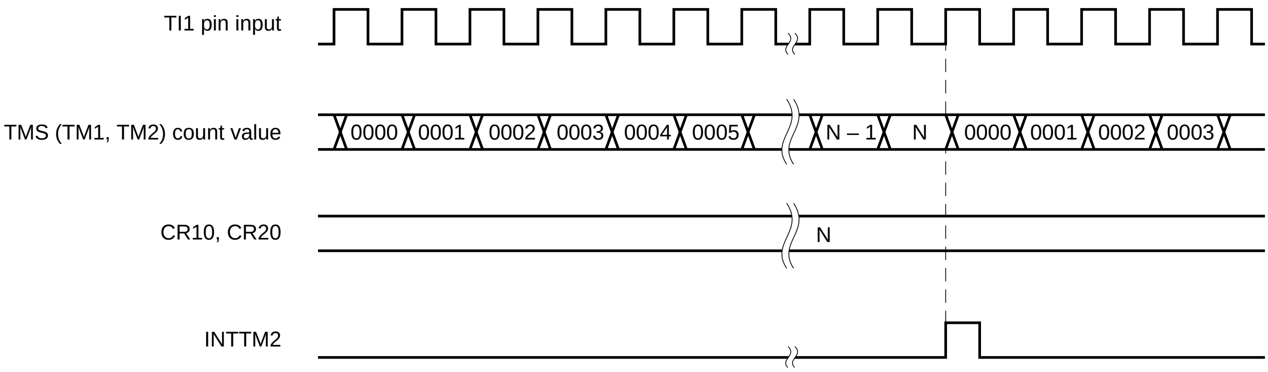
(2) External event counter operations

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

★ TM1 is incremented each time the valid edge specified with timer clock select register 1 (TCL1) is input. If TM1 overflows as a result, the overflow signal is used as a count clock, and TM2 is incremented. Either the rising or falling edge can be selected.

When the count value of TM1 and TM2 matches the value of the 8-bit compare registers (CR10 and CR20), both TM1 and TM2 are cleared to 0 and the interrupt request signal (INTTM2) is generated.

Figure 7-12. External Event Counter Operation Timings (with Rising Edge Specified)



Caution Even in the 16-bit timer/event counter mode, interrupt request (INTTM1) will be generated when the TM1 count value matches the CR10 value, inverting the flipflop of 8-bit timer/event counter output control circuit 1. Thus, when using the 8-bit timer/event counters as a 16-bit interval timer, set the INTTM1 mask flag TMMK1 to 1 to disable INTTM1 acknowledgment.

When reading the 16-bit timer (TMS) count value, use the 16-bit memory manipulation instruction.

(3) Square-wave output operation

Square wave signals can be generated at the user-specified frequency. The frequency, or pulse interval is ★ determined by the value preset in the 8-bit compare registers (CR10 and CR20). To set a count value, set the value of the high-order 8 bits to CR20, and the value of the low-order 8 bits to CR10.

The TO2/P32 pin output status is reversed at intervals of the count value preset to CR10 and CR20 by setting bit 4 (TOE2) of the 8-bit timer output control register (TOC1) to 1. This enables a square wave with any selected frequency to be output.

Table 7-10. Square-Wave Output Ranges when 2-Channel 8-Bit Timer/Event Counters (TM1 and TM2) are Used as 16-Bit Timer/Event Counter

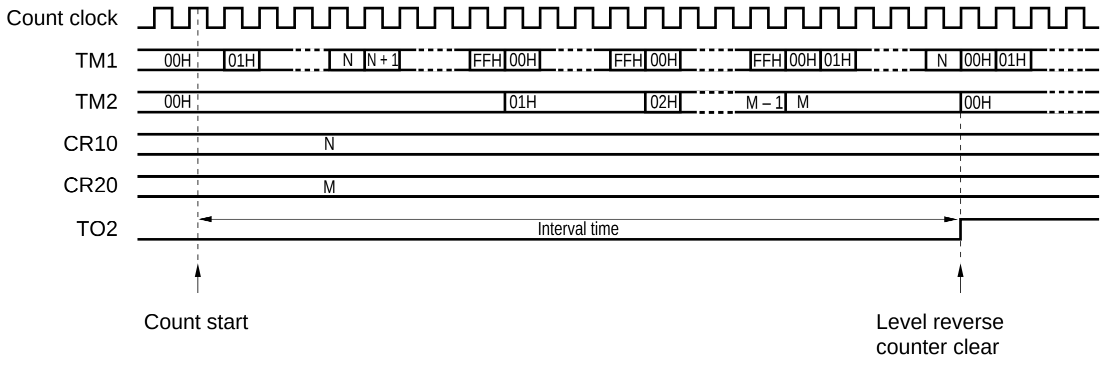
TCL13	TCL12	TCL11	TCL10	Minimum Pulse Time	Maximum Pulse Time	Resolution
0	1	0	1	$2 \times 1/f_X$ (400 ns)	$2^{9} \times 1/f_X$ (26.2 ms)	$2 \times 1/f_X$ (400 ns)
0	1	1	0	$2^{2} \times 1/f_X$ (800 ns)	$2^{10} \times 1/f_X$ (52.4 ms)	$2^{2} \times 1/f_X$ (800 ns)
0	1	1	1	$2^{3} \times 1/f_X$ (1.6 μs)	$2^{11} \times 1/f_X$ (104.9 ms)	$2^{3} \times 1/f_X$ (1.6 μs)
1	0	0	0	$2^{4} \times 1/f_X$ (3.2 μs)	$2^{12} \times 1/f_X$ (209.7 ms)	$2^{4} \times 1/f_X$ (3.2 μs)
1	0	0	1	$2^{5} \times 1/f_X$ (6.4 μs)	$2^{13} \times 1/f_X$ (419.4 ms)	$2^{5} \times 1/f_X$ (6.4 μs)
1	0	1	0	$2^{6} \times 1/f_X$ (12.8 μs)	$2^{14} \times 1/f_X$ (838.9 ms)	$2^{6} \times 1/f_X$ (12.8 μs)
1	0	1	1	$2^{7} \times 1/f_X$ (25.6 μs)	$2^{15} \times 1/f_X$ (1.7 s)	$2^{7} \times 1/f_X$ (25.6 μs)
1	1	0	0	$2^{8} \times 1/f_X$ (51.2 μs)	$2^{16} \times 1/f_X$ (3.4 s)	$2^{8} \times 1/f_X$ (51.2 μs)
1	1	0	1	$2^{9} \times 1/f_X$ (102.4 μs)	$2^{17} \times 1/f_X$ (6.7 s)	$2^{9} \times 1/f_X$ (102.4 μs)
1	1	1	0	$2^{10} \times 1/f_X$ (204.8 μs)	$2^{18} \times 1/f_X$ (13.4 s)	$2^{10} \times 1/f_X$ (204.8 μs)
1	1	1	1	$2^{12} \times 1/f_X$ (819.2 μs)	$2^{20} \times 1/f_X$ (53.7 s)	$2^{12} \times 1/f_X$ (819.2 μs)

Remarks 1. f_X: Main system clock frequency

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

3. Values in parentheses when operated at f_X = 5.0 MHz.

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

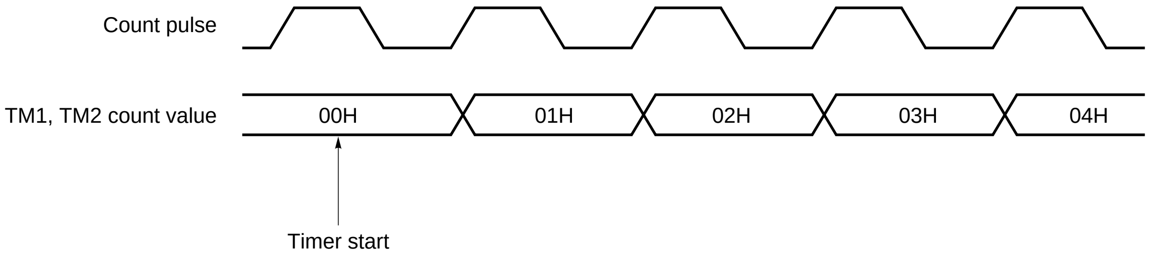


7.5 8-Bit Timer/Event Counter Operating Precautions

(1) Timer start errors

An error with a maximum of one clock may occur concerning the time required for a match signal to be generated after timer start. This is because 8-bit timer registers 1 and 2 (TM1 and TM2) are started asynchronously with the count pulse.

Figure 7-14. 8-Bit Timer Register Start Timings



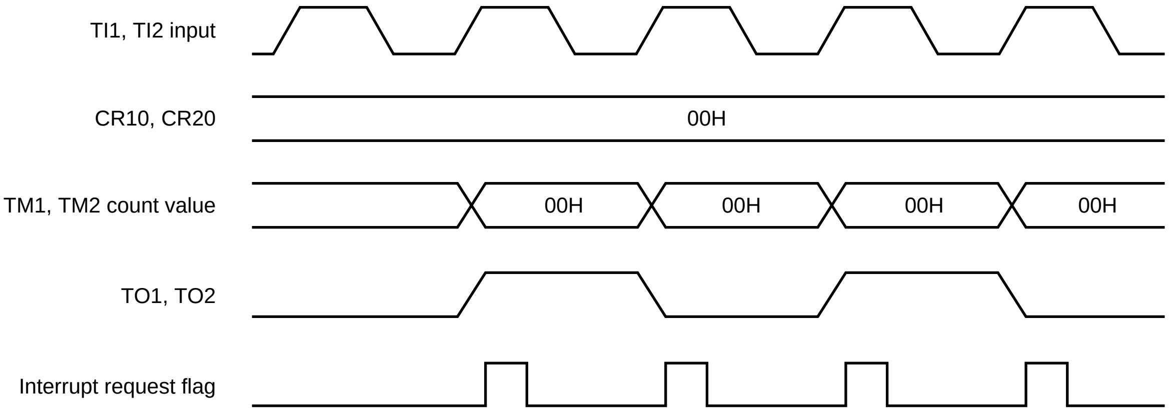
(2) 8-bit compare registers 1 and 2 sets

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

Thus, when the 8-bit compare register is used as event counter, one-pulse count operation can be carried out.

When the 8-bit compare register is used as 16-bit timer/event counter, write data to CR10 and CR20 after setting bit 0 (TCE1) of the 8-bit timer mode control register (TMC1) to 0 and stopping timer operation.

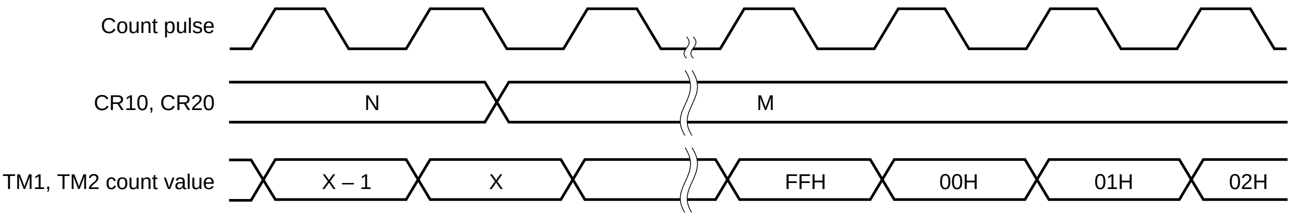
Figure 7-15. External Event Counter Operation Timings



(3) Operation after compare register change during timer count operation

If the values after the 8-bit compare registers (CR10 and CR20) are changed are smaller than those of 8-bit timer registers (TM1 and TM2), TM1 and TM2 continue counting, overflow and then restart counting from 0. Thus, if the value after CR10 and CR20 (M) change is smaller than that before change (N), it is necessary to restart the timer after changing CR10 and CR20.

Figure 7-16. Timings after Compare Register Change during Timer Count Operation



Remark N > X > M

CHAPTER 8 WATCH TIMER

8.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.

(1) Watch timer

When the 32.768-kHz subsystem clock is used, a flag (WTIF) is set at 0.5 second or 0.25 second intervals. In addition, when the 4.19-MHz (Standard: 4.19304 MHz) main system clock is used, a flag (WTIF) is set at 0.5 second or 1 second intervals.

Caution 0.5 second intervals cannot be generated with the 5.0-MHz main system clock. You should switch to the 32.768-kHz subsystem clock to generate 0.5 second intervals.

(2) Interval timer

Interrupt requests (INTTM3) are generated at the preset time interval.

Table 8-1. Interval Timer Interval Time

Interval Time	When Operated at f_X = 5.0 MHz	When Operated at f_X = 4.19 MHz	When Operated at f_{XT} = 32.768 kHz
$2^{12} \times 1/f_X$	819 μs	978 μs	488 μs
$2^{13} \times 1/f_X$	1.64 ms	1.96 ms	977 μs
$2^{14} \times 1/f_X$	3.28 ms	3.91 ms	1.95 ms
$2^{15} \times 1/f_X$	6.55 ms	7.82 ms	3.91 ms
$2^{16} \times 1/f_X$	13.1 ms	15.6 ms	7.81 ms
$2^{17} \times 1/f_X$	26.2 ms	31.3 ms	15.6 ms

Remarks 1. f_X: Main system clock frequency

2. f_{XT}: Subsystem clock oscillation frequency

8.2 Watch Timer Configuration

The watch timer consists of the following hardware.

Table 8-2. Watch Timer Configuration

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

8.3 Watch Timer Control Registers

The following two types of registers are used to control the watch timer.

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

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

This register sets the watch timer count clock.
TCL2 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL2 to 00H.

Remark Besides setting the watch timer count clock, TCL2 sets the watchdog timer count clock and buzzer output frequency.

Figure 8-1. Watch Timer Block Diagram

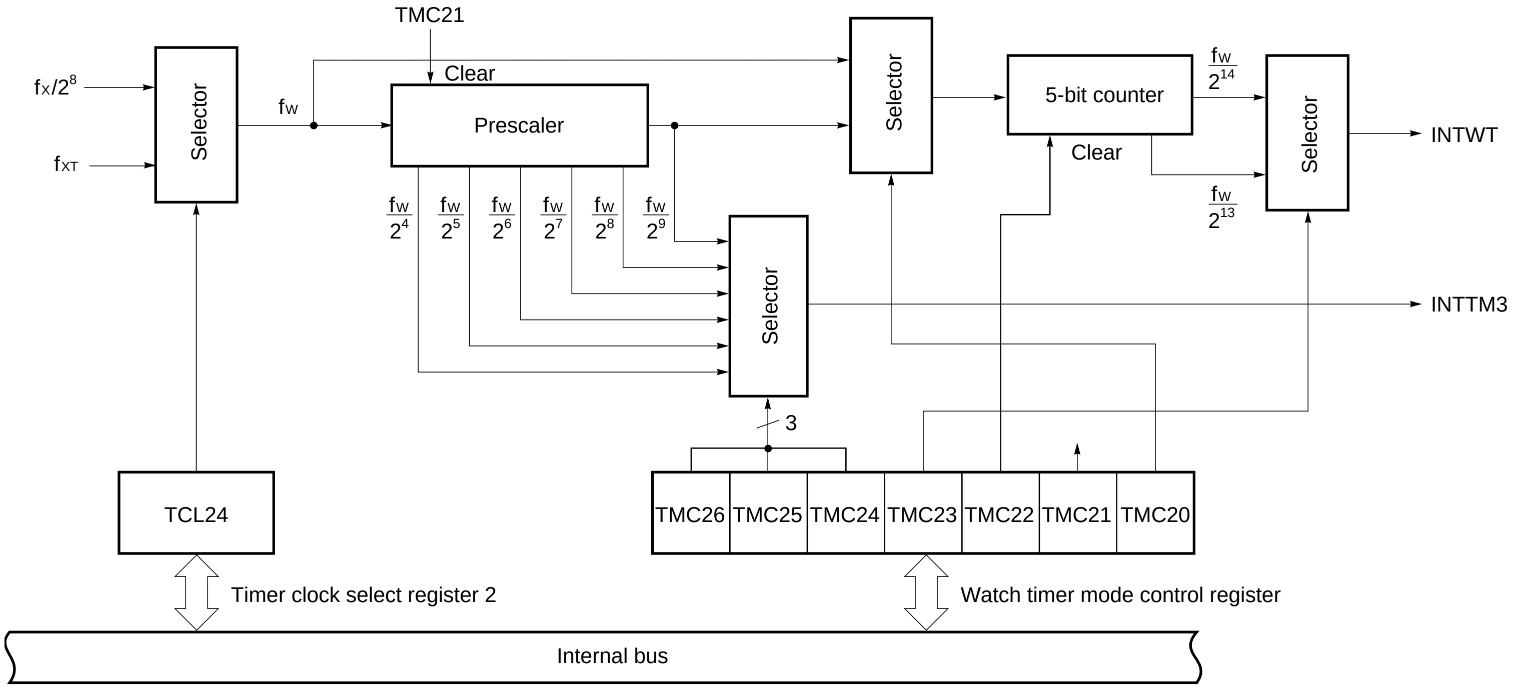


Figure 8-2. Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Count Clock Selection	
			Watchdog Timer Mode	Interval Timer Mode
0	0	0	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
0	0	1	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
0	1	0	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
0	1	1	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	0	0	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	1	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	0	$f_X/2^9$ (9.8 kHz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)

TCL24	Watch Timer Count Clock Selection Note
0	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disable
1	0	0	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

Note When using a main system clock of 1.25 MHz or less and FIP controller/driver, select $f_X/2^8$ as the counter clock for watch timer.

Caution If TCL2 is to be rewritten in data other than identical data, the timer operation must be stopped first.

Remarks 1. f_X: Main system clock frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. ×: Don't care

4. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(2) Watch timer mode control register (TMC2)

This register sets the watch timer operating mode, watch flag set time and prescaler interval time and enables/ disables prescaler and 5-bit counter operations. TMC2 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC2 to 00H.

Figure 8-3. Watch Timer Mode Control Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TMC2	0	TMC26	TMC25	TMC24	TMC23	TMC22	TMC21	TMC20	FF4AH	00H	R/W

TMC23	TMC20	Watch Flag Set Time Selection
0	0	$2^{14}/f_W$ (0.5 s)
1		$2^{13}/f_W$ (0.25 s)
0	1	$2^{5}/f_W$ (977 μs)
1		$2^{4}/f_W$ (488 μs)

TMC21	Prescaler Operation Control Note
0	Clear after operation stops
1	Operation enable

TMC22	5-Bit Counter Operation Control
0	Clear after operation stops
1	Operation enable

TMC26	TMC25	TMC24	Prescaler Interval Time Selection
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)
Other than above			Setting prohibited

Note Do not frequently clear the prescaler when using the watch timer.

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^8$ or f_{XT})

2. Figures in parentheses apply to operation with f_W = 32.768 kHz.

8.4 Watch Timer Operations

8.4.1 Watch timer operation

When the 32.768-kHz subsystem clock is used, the timer operates as a watch timer with a 0.5 second or 0.25 second interval. In addition, when the 4.19-MHz main system clock is used, the timer can operate as a watch timer with a 0.5 second or 1 second interval.

The watch timer sets the test input flag (WTIF) to 1 at the constant time interval. The standby state (STOP mode/HALT mode) can be cleared by setting WTIF to 1.

When bit 2 (TMC22) of the watch timer mode control register (TMC2) is set to 0, the 5-bit counter is cleared and the count operation stops.

For simultaneous operation of the interval timer, zero-second start can be achieved by setting TMC22 to 0 (maximum error: 26.2 ms when operated at 5.0 MHz).

8.4.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt requests repeatedly at an interval of the preset count value.

The interval time can be selected with bits 4 to 6 (TMC24 to TMC26) of the watch timer mode control register (TMC2).

Table 8-3. Interval Timer Interval Time

TMC26	TMC25	TMC24	Interval Time	When Operated at f_X = 5.0 MHz	When Operated at f_X = 4.19 MHz	When Operated at f_{XT} = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	819 μs	978 μs	488 μs
0	0	1	$2^5 \times 1/f_W$	1.64 ms	1.96 ms	977 μs
0	1	0	$2^6 \times 1/f_W$	3.28 ms	3.91 ms	1.95 ms
0	1	1	$2^7 \times 1/f_W$	6.55 ms	7.82 ms	3.91 ms
1	0	0	$2^8 \times 1/f_W$	13.1 ms	15.6 ms	7.81 ms
1	0	1	$2^9 \times 1/f_W$	26.2 ms	31.3 ms	15.6 ms
Other than above			Setting prohibited			

f_X: Main system clock frequency

f_{XT}: Subsystem clock oscillation frequency

f_W: Watch timer clock frequency ($f_X/2^8$ or f_{XT})

CHAPTER 9 WATCHDOG TIMER

9.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

Caution Select the watchdog timer mode or the interval timer mode with the watchdog timer mode register (WDTM) (The watchdog timer and interval timer cannot be used at the same time).

(1) Watchdog timer mode

An inadvertent program loop is detected. Upon detection of the inadvertent program loop, a non-maskable interrupt request or $\overline{\text{RESET}}$ can be generated.

Table 9-1. Watchdog Timer Inadvertent Detection Time

Inadvertent Detection Time	When Operated at f_X = 5.0 MHz	Inadvertent Detection Time	When Operated at f_X = 5.0 MHz
$2^{11} \times 1/f_X$	410 μs	$2^{15} \times 1/f_X$	6.55 μs
$2^{12} \times 1/f_X$	819 μs	$2^{16} \times 1/f_X$	13.1 μs
$2^{13} \times 1/f_X$	1.64 ms	$2^{17} \times 1/f_X$	26.2 ms
$2^{14} \times 1/f_X$	3.28 ms	$2^{19} \times 1/f_X$	105.0 ms

f_X: Main system clock oscillation frequency

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 9-2. Interval Time

Interval Time	When Operated at f_X = 5.0 MHz	Interval Time	When Operated at f_X = 5.0 MHz
$2^{12} \times 1/f_X$	819 μs	$2^{16} \times 1/f_X$	13.1 ms
$2^{13} \times 1/f_X$	1.64 ms	$2^{17} \times 1/f_X$	26.2 ms
$2^{14} \times 1/f_X$	3.28 ms	$2^{18} \times 1/f_X$	52.4 ms
$2^{15} \times 1/f_X$	6.55 ms	$2^{20} \times 1/f_X$	210 ms

f_X: Main system clock oscillation frequency

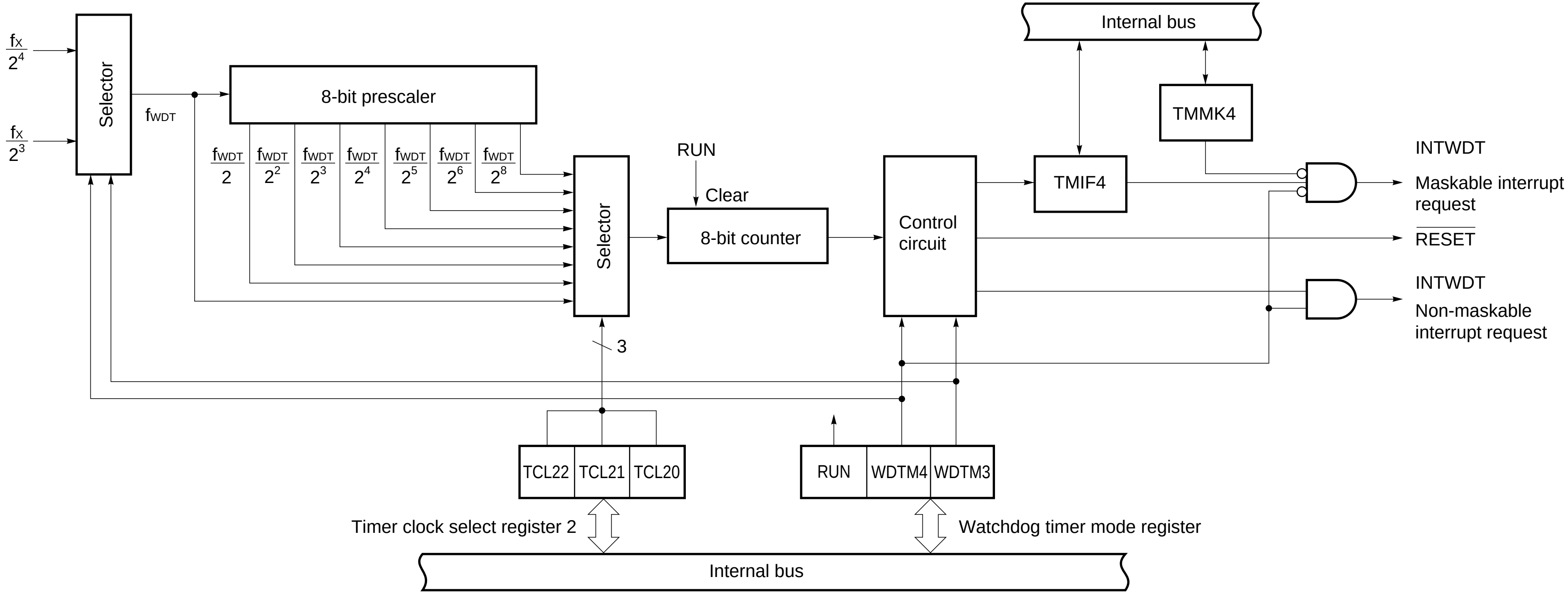
9.2 Watchdog Timer Configuration

The watchdog timer consists of the following hardware.

Table 9-3. Watchdog Timer Configuration

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

Figure 9-1. Watchdog Timer Block Diagram



9.3 Watchdog Timer Control Registers

The following two types of registers are used to control the watchdog timer.

- Timer clock select register 2 (TCL2)
- Watchdog timer mode register (WDTM)

(1) Timer clock select register 2 (TCL2)

This register sets the watchdog timer count clock.
TCL2 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL2 to 00H.

Remark Besides setting the watchdog timer count clock, TCL2 sets the watch timer count clock and buzzer output clock.

Figure 9-2. Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Count Clock Selection	
			Watchdog Timer Mode	Interval Timer Mode
0	0	0	$f_X/2^3$ (625 kHz)	$f_X/2^4$ (313 kHz)
0	0	1	$f_X/2^4$ (313 kHz)	$f_X/2^5$ (156 kHz)
0	1	0	$f_X/2^5$ (156 kHz)	$f_X/2^6$ (78.1 kHz)
0	1	1	$f_X/2^6$ (78.1 kHz)	$f_X/2^7$ (39.1 kHz)
1	0	0	$f_X/2^7$ (39.1 kHz)	$f_X/2^8$ (19.5 kHz)
1	0	1	$f_X/2^8$ (19.5 kHz)	$f_X/2^9$ (9.8 kHz)
1	1	0	$f_X/2^9$ (9.8k Hz)	$f_X/2^{10}$ (4.9 kHz)
1	1	1	$f_X/2^{11}$ (2.4 kHz)	$f_X/2^{12}$ (1.2 kHz)

TCL24	Watch Timer Count Clock Selection Note
0	$f_X/2^8$ (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disable
1	0	0	$f_X/2^{10}$ (4.9 kHz)
1	0	1	$f_X/2^{11}$ (2.4 kHz)
1	1	0	$f_X/2^{12}$ (1.2 kHz)
1	1	1	Setting prohibited

Note $f_X/2^8$ must be selected as the watch timer count clock when using main system clock at 1.25 MHz or less and FIP controller/driver.

Caution If TCL2 is to be rewritten in data other than identical data, the timer operation must be stopped first.

Remarks 1. f_X: Main system clock frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. ×: don’t care

4. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(2) Watchdog timer mode register (WDTM)

This register sets the watchdog timer operating mode and enables/disables counting.

WDTM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears WDTM to 00H.

★ **Figure 9-3. Watchdog Timer Mode Register Format**

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Watchdog Timer Operating Mode Selection[Note 1]
0	×	Interval timer Mode[Note 2] (Maskable interrupt request occurs upon generation of an overflow.)
1	0	Watchdog timer mode 1 (Non-maskable interrupt request occurs upon generation of an overflow.)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow.)

RUN	Watchdog Timer Operation Selection[Note 3]
0	Count stop
1	Counter is cleared and counting starts.

Notes 1. Once set to 1, WDTM3 and WDTM4 cannot be cleared to 0 by software.

2. The watchdog timer starts operating as an interval timer as soon as RUN has been set to 1.

3. Once set to 1, RUN cannot be cleared to 0 by software. Thus, once counting starts, it can only be stopped by $\overline{\text{RESET}}$ input.

Cautions 1. When 1 is set in RUN so that watchdog timer is cleared, the actual overflow time is up to 0.5% shorter than the time set by the timer clock select register 2 (TCL2).

★ **2. When using watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming that the interrupt request flag (TMIF4) is 0. If WDTM4 is set to 1 when TMIF4 is 1, the non-maskable interrupt request is generated regardless of the contents of WDTM3.**

Remark ×: don't care

9.4 Watchdog Timer Operations

9.4.1 Watchdog timer operation

When bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 1, the watchdog timer is operated to detect any inadvertent program loop.

The watchdog timer count clock (inadvertent program loop detection time interval) can be selected with bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2).

Watchdog timer starts by setting bit 7 (RUN) of WDTM to 1. After the watchdog timer is started, set RUN to 1 within the set overrun time detection interval. The watchdog timer can be cleared and counting is started by setting RUN to 1. If RUN is not set to 1 and the inadvertent program loop detection time is past, system reset or a non-maskable interrupt request is generated according to the WDTM bit 3 (WDTM3) value.

The watchdog timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the watchdog timer and then execute the STOP instruction.

Cautions 1. The actual overrun detection time may be shorter than the set time by a maximum of 0.5%.

2. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 9-4. Watchdog Timer Overrun Detection Time

TCL22	TCL21	TCL20	Runaway Detection	f_X = 5.0 MHz
0	0	0	$2^{11} \times 1/f_X$	410 μs
0	0	1	$2^{12} \times 1/f_X$	819 μs
0	1	0	$2^{13} \times 1/f_X$	1.64 ms
0	1	1	$2^{14} \times 1/f_X$	3.28 ms
1	0	0	$2^{15} \times 1/f_X$	6.55 ms
1	0	1	$2^{16} \times 1/f_X$	13.1 ms
1	1	0	$2^{17} \times 1/f_X$	26.2 ms
1	1	1	$2^{19} \times 1/f_X$	105.0 ms

f_X: Main system clock frequency

9.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at intervals ★ of a preset count value when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0.

★ Count clock (interval time) can be selected with bits 0 to 2 (TCL20 to TCL22) of the timer clock select register 2 (TCL2). The watchdog timer starts operating as an interval timer when bit 7 (RUN) of WDTM has been set to 1.

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specify flag (TMPR4) are validated and the maskable interrupt request (INTWDT) can be generated. Among maskable interrupt requests, the INTWDT default has the highest priority.

The interval timer continues operating in the HALT mode but it stops in the STOP mode. Thus, set RUN to 1 before the STOP mode is set, clear the interval timer and then execute the STOP instruction.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (with the watchdog timer mode selected), the interval timer mode is not set unless $\overline{\text{RESET}}$ input is applied.

2. The interval time just after setting with WDTM may be shorter than the set time by a maximum of 0.5%.

3. When the subsystem clock is selected for CPU clock, watchdog timer count operation is stopped.

Table 9-5. Interval Timer Interval Time

TCL22	TCL21	TCL20	Interval Detection	f_X = 5.0 MHz
0	0	0	$2^{12} \times 1/f_X$	819 μs
0	0	1	$2^{13} \times 1/f_X$	1.64 ms
0	1	0	$2^{14} \times 1/f_X$	3.28 ms
0	1	1	$2^{15} \times 1/f_X$	6.55 ms
1	0	0	$2^{16} \times 1/f_X$	13.1 ms
1	0	1	$2^{17} \times 1/f_X$	26.2 ms
1	1	0	$2^{18} \times 1/f_X$	52.4 ms
1	1	1	$2^{20} \times 1/f_X$	210.0 ms

f_X: Main system clock frequency

CHAPTER 10 6-BIT UP/DOWN COUNTER

10.1 6-Bit Up/Down Counter Function

The 6-bit up/down counter increases and decreases its count when effective edges are input from the C10/P03/INTP3 pin.

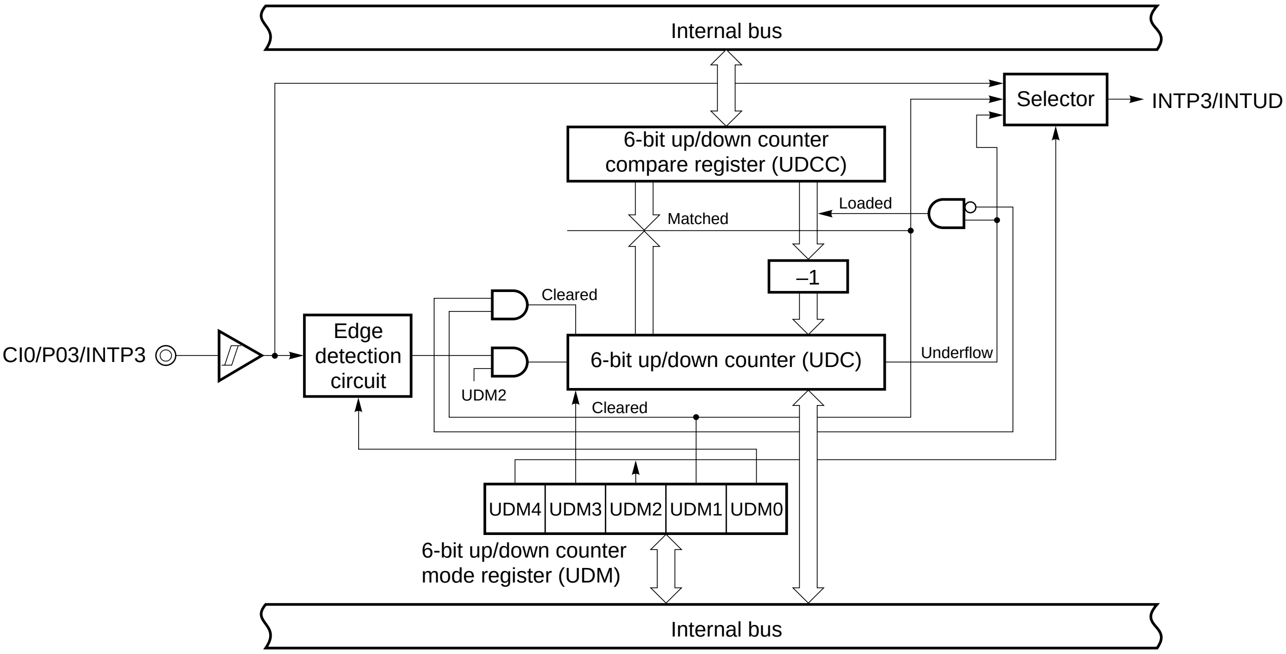
10.2 6-Bit Up/Down Counter Configuration

The 6-bit up/down counter consists of the following hardware.

Table 10-1. 6-Bit Up/Down Counter Configuration

Item	Configuration
Counter register	6-bit up/down counter (UDC)
Register	6-bit up/down counter compare register (UDCC)
Control register	6-bit up/down counter mode register (UDM)

Figure 10-1. 6-Bit Up/Down Counter Block Diagram



Caution When using the 6-bit up/down counter, set the CI0/P03/INTP3 pin to the input mode (set 1 to bit 3 (PM03) of the port mode register 0).

(1) 6-bit up/down counter (UDC)

This 6-bit register counts the count pulses input to the CI0/P03/INTP3 pin.
The UDC is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears UDC to 00H.

Cautions 1. Be sure to stop count operations first before setting the UDC.

2. Read and write the UDC after inputting the $\overline{\text{RESET}}$ signal and setting data to the UDM.

(2) 6-bit up/down counter compare register (UDCC)

This register constantly compares the values set for the UDCC and the count values in the 6-bit up/down counter (UDC), and generates the interruption request (INTUD) when the values match.
When UDCC and UDC values match during up count operations, the UDCC sets the interruption request flag (INTUD) and clears the UDC to 0.
When UDCC and UDC values match during down count operations, the UDCC sets the interruption request flag (INTUD) and loads a value less than the UDCC by one to the UDC.
The UDCC is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears UDCC to 00H.

Caution Be sure to stop count operations first before setting the UDCC.

10.3 6-Bit Up/Down Counter Control Register

The 6-bit up/down counter is controlled by the 6-bit up/down counter mode register (UDM). The register clears the counter, enables/disables count operations, selects up count or down count, and sets the valid edges of input signals and the INTP3 flag set signal.

The UDM is set with a 1-bit memory manipulation instruction or 8-bit memory manipulation instruction. $\overline{\text{RESET}}$ input clears UDM to 00H.

Figure 10-2. 6-Bit Up/Down Counter Mode Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
UDM	0	0	0	UDM4	UDM3	UDM2	UDM1	UDM0	FFA8H	00H	R/W

UDM0	Valid Edge Selection
0	CI0 falling edge
1	CI0 rising edge

UDM1	Operation Mode Selection
0	Down counter operation
1	Up counter operation

UDM2	Count Operation Control
0	The count operation is stopped (The count value is retained)
1	The count operation is enabled

UDM3	Counter Clear
0	Count operation
1	The counter is cleared

UDM4	INTP3 Flag Set Signal Selection
0	Set using the INTP3 interruption signal
1	Set using the INTUD interruption signal

Cautions 1. Do not set the UDM0, UDM1, and UDM3 using the same timings as the effective edges input to the CI0/P03/INTP3 pin.

2. The UDC is cleared to 0 by writing 1 in the UDM3.
When the UDC is cleared, the UDM3 is automatically cleared to 0.

3. The UDC cannot be read/written until the $\overline{\text{RESET}}$ signal is input and data is set to the UDM.

10.4 6-Bit Up/Down Counter Precautions

(1) Precautions for the 6-bit up/down counter (UDC)

- Be sure to stop counter operations first before setting the UDC.
- When using the UDC, set the CI0/P03/INTP3 pin to the input mode (set 1 to bit 3 (PM03) of the port mode register 0).
- When writing the data in the UDC after releasing the reset and before starting UDC operations, set the UDM to 0 first.

(2) Precautions for the 6-bit up/down counter compare register (UDCC)

- Be sure to stop counter operations first before setting the UDCC.

(3) Precautions for the 6-bit up/down counter mode register (UDM)

- Do not set the UDM0, UDM1, and UDM3 using the same timings as the effective edges input to the CI0/P03/INTP3 pin.
- The UDM4 can be rewritten from 0 to 1 immediately after it is reset. Do not rewrite it during UDC operations nor rewrite it from 1 to 0.

CHAPTER 11 CLOCK OUTPUT CONTROL CIRCUIT

11.1 Clock Output Control Circuit Functions

The clock output control circuit is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSI. Clocks selected with the timer clock select register 0 (TCL0) are output from the PCL/P35 pin.

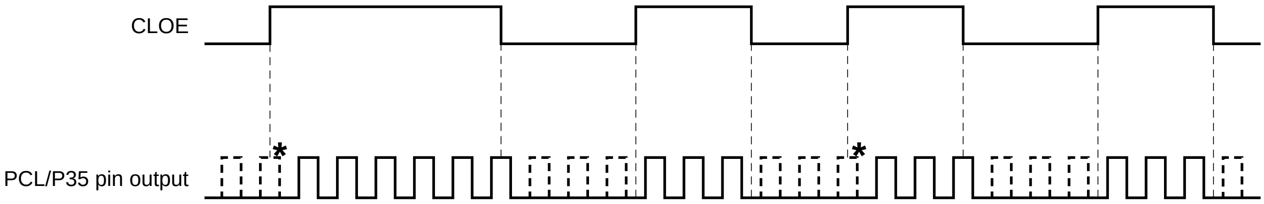
Follow the procedure below to output clock pulses.

[1] Select the clock pulse output frequency (with clock pulse output disabled) with bits 0 to 3 (TCL00 to TCL03) of TCL0.
[2] Set the P35 output latch to 0.
[3] Set bit 5 (PM35) of port mode register 3 (PM3) to 0 (set to output mode).
[4] Set bit 7 (CLOE) of TCL0 to 1.

Caution   Clock output cannot be used when setting P35 output latch to 1.

Remark When clock output enable/disable is switched, the clock output control circuit does not output pulses with small widths (See the mark * in **Figure 11-1**).

Figure 11-1. Remote Controlled Output Application Example



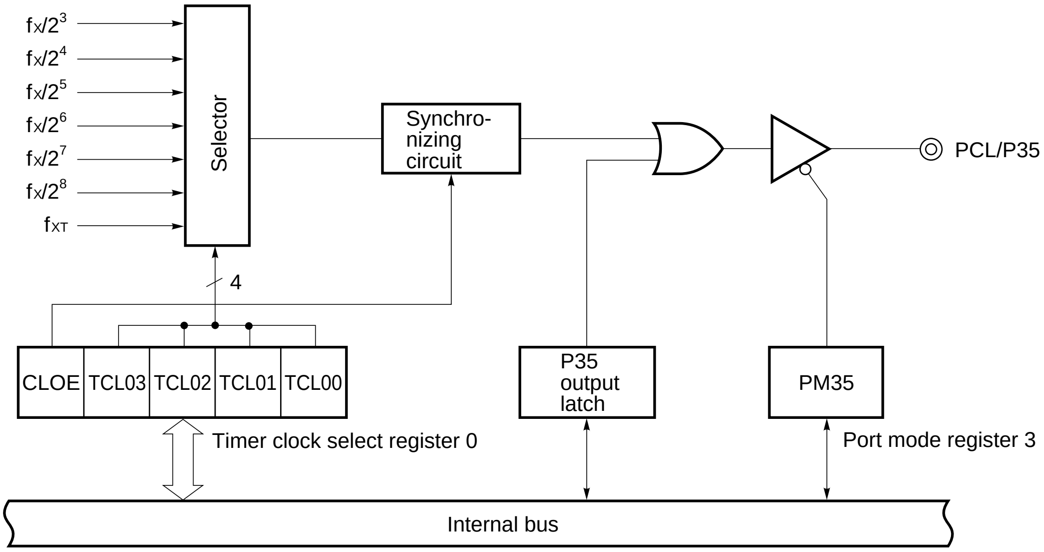
11.2 Clock Output Control Circuit Configuration

The clock output control circuit consists of the following hardware.

Table 11-1. Clock Output Control Circuit Configuration

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

Figure 11-2. Clock Output Control Circuit Block Diagram



11.3 Clock Output Function Control Registers

The following two types of registers are used to 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 PCL output clock.

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

$\overline{\text{RESET}}$ input clears TCL0 to 00H.

Remark Besides setting PCL output clock, TCL0 sets the 16-bit timer register count clock.

Figure 11-3. Timer Clock Select Register 0 Format

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

TCL03	TCL02	TCL01	TCL00	PCL Output Clock Selection
0	0	0	0	f_{XT} (32.768 kHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
Other than above				Setting prohibited

TCL06	TCL05	TCL04	16-bit Timer Register Count Clock Selection
0	0	0	TI0 (Valid edge specifiable)
0	0	1	f_X (5.0 MHz)
0	1	0	$f_X/2$ (2.5 MHz)
0	1	1	$f_X/2^2$ (1.25 MHz)
1	0	0	$f_X/2^3$ (625 kHz)
Other than above			Setting prohibited

CLOE	PCL Output Control
0	Output disable
1	Output enable

Cautions 1. Setting of the TI0/P00/INTP0 pin valid edge is performed by external interrupt mode register (INTM0), and selection of the sampling clock frequency is performed by the sampling clock select register (SCS).

2. When enabling PCL output, set TCL00 to TCL03, then set 1 in CLOE with a 1-bit memory manipulation instruction.

3. To read the count value when TI0 has been specified as the TM0 count clock, the value should be read from TM0, not from the 16-bit capture register (CR01).

4. If TCL0 is to be rewritten in data other than identical data, the timer operation must be stopped first.

Remarks 1. f_X : Main system clock frequency

2. f_{XT} : Subsystem clock oscillation frequency

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

4. TM0 : 16-bit timer register

5. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register 3 (PM3)

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

When using the P35/PCL pin for clock output function, set PM35 and output latch of P35 to 0.

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

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Figure 11-4. Port Mode Register 3 Format

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

PM3n	P3n Pin Input/Output Mode Selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

CHAPTER 12 BUZZER OUTPUT CONTROL CIRCUIT

12.1 Buzzer Output Control Circuit Functions

The buzzer output control circuit outputs 1.2 kHz, 2.4 kHz, or 4.9 kHz frequency square-wave. The buzzer frequency selected with timer clock select register 2 (TCL2) is output from the BUZ/P36 pin.

Follow the procedure below to output the buzzer frequency.

[1] Select the buzzer output frequency with bits 5 to 7 (TCL25 to TCL27) of TCL2.
[2] Set the P36 output latch to 0.
[3] Set bit 6 (PM36) of port mode register 3 (PM3) to 0 (Set to output mode).

Caution   Buzzer output cannot be used when setting P36 output latch to 1.

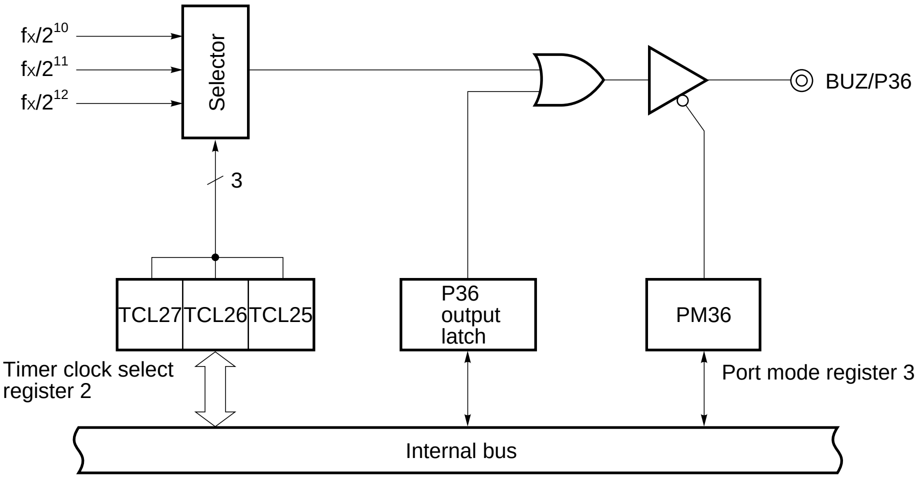
12.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 12-1. Buzzer Output Control Circuit Configuration

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

Figure 12-1. Buzzer Output Control Circuit Block Diagram



12.3 Buzzer Output Function Control Registers

The following two types of registers are used to 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 buzzer output frequency.
TCL2 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL2 to 00H.

Remark Besides setting the buzzer output frequency, TCL2 sets the watch timer count clock and the watchdog timer count clock.

Figure 12-2. Timer Clock Select Register 2 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL2	TCL27	TCL26	TCL25	TCL24	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Count Clock Selection	
			Watchdog Timer Mode	Interval Timer Mode
0	0	0	fx/2^3 (625 kHz)	fx/2^4 (313 kHz)
0	0	1	fx/2^4 (313 kHz)	fx/2^5 (156 kHz)
0	1	0	fx/2^5 (156 kHz)	fx/2^6 (78.1 kHz)
0	1	1	fx/2^6 (78.1 kHz)	fx/2^7 (39.1 kHz)
1	0	0	fx/2^7 (39.1 kHz)	fx/2^8 (19.5 kHz)
1	0	1	fx/2^8 (19.5 kHz)	fx/2^9 (9.8 kHz)
1	1	0	fx/2^9 (9.8 kHz)	fx/2^{10} (4.9 kHz)
1	1	1	fx/2^{11} (2.4 kHz)	fx/2^{12} (1.2 kHz)

TCL24	Watch Timer Count Clock Selection
0	fx/2^8 (19.5 kHz)
1	f_{XT} (32.768 kHz)

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection
0	×	×	Buzzer output disable
1	0	0	fx/2^{10} (4.9 kHz)
1	0	1	fx/2^{11} (2.4 kHz)
1	1	0	fx/2^{12} (1.2 kHz)
1	1	1	Setting prohibited

Caution If TCL2 is to be rewritten in data other than identical data, the timer operation must be stopped first.

Remarks 1. f_X : Main system clock frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. × : don't care

4. Figures in parentheses apply to operation with f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register 3 (PM3)

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

When using the P36/BUZ pin for buzzer output function, set PM36 and output latch of P36 to 0.

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

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Figure 12-3. Port Mode Register 3 Format

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

PM3n	P3n Pin Input/Output Mode Selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

CHAPTER 13 A/D CONVERTER

13.1 A/D Converter Functions

The A/D converter converts an analog input into a digital value. It consists of 8 channels (ANI0 to ANI7) with an 8-bit resolution.

The conversion method is based on successive approximation and the conversion result is held in the 8-bit A/D conversion result register (ADCR).

The following two ways are available to start A/D conversion.

(1) Hardware start

Conversion is started by trigger input (INTP3).

(2) Software start

Conversion is started by setting the A/D converter mode register (ADM).

One channel of analog input is selected from ANI0 to ANI7 and A/D conversion is carried out. In the case of hardware start, A/D conversion operation stops when it terminates, and an interrupt request (INTAD) is generated. In the case of software start, the A/D conversion operation is repeated. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

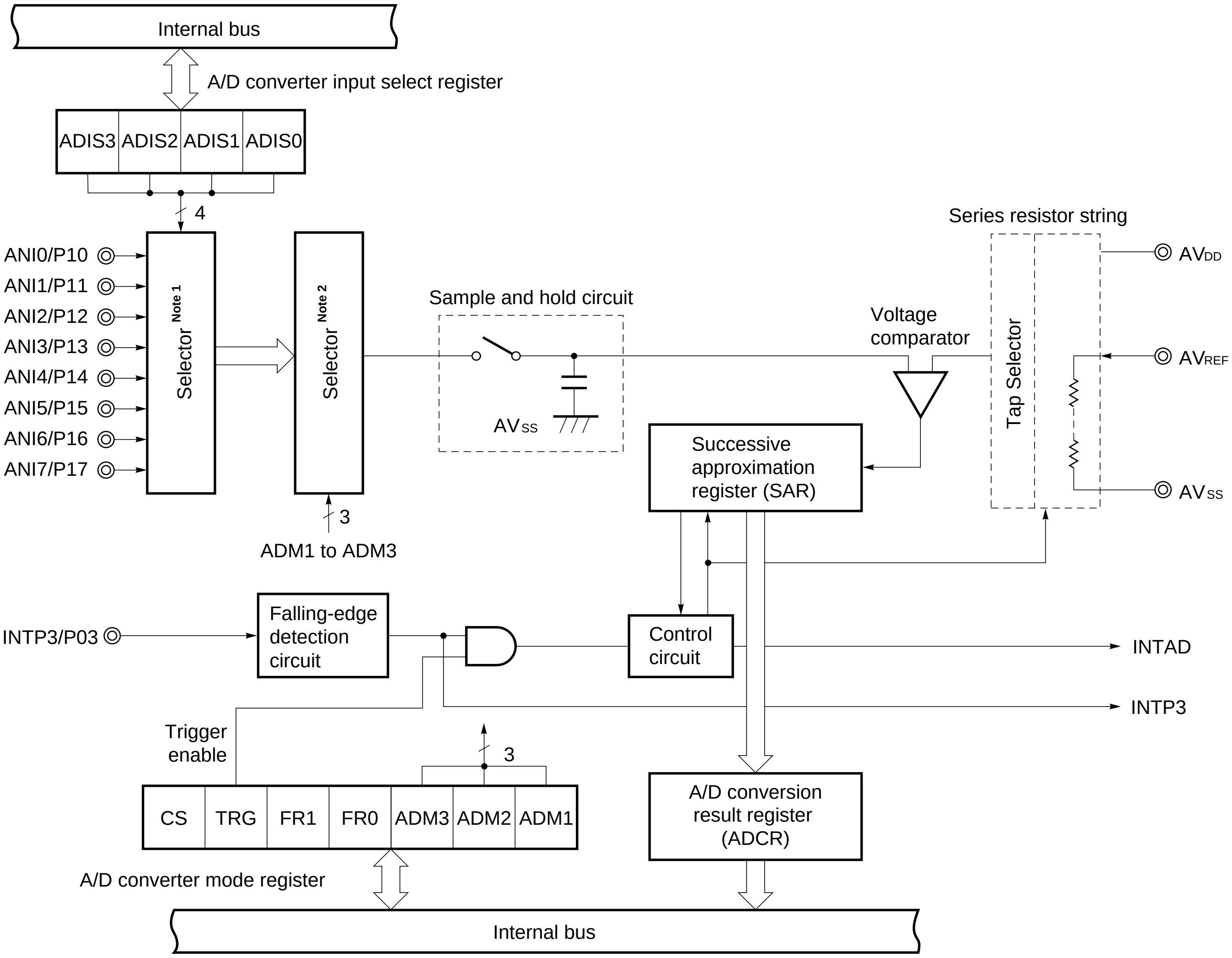
13.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 13-1. A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Control register	A/D converter mode register (ADM) A/D converter input select register (ADIS)
Register	Successive approximation register (SAR) A/D conversion result register (ADCR)

Figure 13-1.   A/D Converter Block Diagram



Notes 1. Selector to select the number of channels to be used for analog input

2. Selector to select the channel for A/D conversion

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string and holds the result from the most significant bit (MSB).

When up to the least significant bit (LSB) is held (termination of A/D conversion), the SAR contents are transferred to the A/D conversion result register (ADCR).

(2) A/D conversion result register (ADCR)

This register holds the A/D conversion result. Each time A/D conversion terminates, the conversion result is loaded from the successive approximation register (SAR).

ADCR is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR undefined.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input signal sequentially applied from the input circuit and sends it 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 to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS} and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are 8-channel analog input pins to input analog signals to undergo A/D conversion to the A/D converter.

These pins except analog input pins selected with the A/D converter input select register (ADIS) can be used as the input/output port.

Caution Use ANI0 to ANI7 input voltages within the specified range. If a voltage higher than AV_{REF} or lower than AV_{SS} is applied (even if within the absolute maximum ratings), the converted value of the corresponding channel becomes indeterminate and may adversely affect the converted values of other channels.

(7) AV_{REF} pin

This pin inputs the A/D converter reference voltage.

It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.

(8) AV_{SS} pin

Ground potential pin of the A/D converter. It must be at the same level as the V_{SS} pin even if the A/D converter is not used.

(9) AV_{DD} pin

Analog power supply pin of the A/D converter. It must be at the same level as the V_{DD} pin even if the A/D converter is not used.

13.3 A/D Converter Control Registers

The following two types of registers are used to 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 analog input channel for A/D conversion, conversion time, conversion start/stop, and external trigger.

ADM is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADM to 01H.

Figure 13-2. A/D Converter Mode Register Format

Symbol	⑦	⑥	5	4	3	2	1	0	Address	After Reset	R/W
ADM	CS	TRG	FR1	FR0	ADM3	ADM2	ADM1	HSC	FF80H	01H	R/W

ADM3	ADM2	ADM1	Analog Input Channel Selection
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	A/D Conversion Time Selection Note 1	
		When Operated at f_X = 5.0 MHz	When Operated at f_X = 4.19 MHz
0	0	160/f_X (32.0 μs)	160/f_X (38.1 μs)
0	1	80/fx (setting prohibited Note 2)	80/f_X (19.1 μs)
1	0	200/f_X (40.0 μs)	200/f_X (47.7 μs)
1	1	Setting prohibited	

TRG	External Trigger Selection
0	No external trigger (software starts mode)
1	Conversion started by external trigger (hardware starts mode)

CS	A/D Conversion Operation Control
0	Operation stop
1	Operation start

Notes 1. Set so that the A/D conversion time is 19.1 μs or more.

2. Setting prohibited because A/D conversion time is less than 19.1 μs.

Cautions 1. "1" should be set in bit 0.

2. In consideration of power consumption reduction at the A/D converter when the standby function is working, clear bit 7 (CS) of this register to 0 to stop the A/D conversion operation before executing a HALT or STOP instruction.

3. When restarting the stopped A/D conversion operation, start the A/D conversion operation after clearing the interrupt request flag (ADIF) to 0.

Remark f_X: Main system clock frequency

(2) A/D converter input select register (ADIS)

This register determines whether the ANI0/P10 to ANI7/P17 pins should be used for analog input channels or ports. The pins which are not selected for analog input pins can be used as the input/output port.

ADIS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADIS to 00H.

Cautions 1. Set the analog input channel in the following order.

[1] Set the number of analog input channels with ADIS.

[2] Using ADM, select one channel to undergo A/D conversion among the channels which are set for analog input with ADIS.

2. No internal pull-up resistor can be connected to the channels set for analog input with ADIS, irrespective of the value of bit 1 (PUO1) of the pull-up resistor option register.

Figure 13-3. A/D Converter Input Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADIS	0	0	0	0	ADIS3	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADIS3	ADIS2	ADIS1	ADIS0	Number of Analog Input Channel Selection
0	0	0	0	No analog input channel (P10 to P17)
0	0	0	1	1 channel (ANI0, P11 to P17)
0	0	1	0	2 channels (ANI0, ANI1, P12 to P17)
0	0	1	1	3 channels (ANI0 to ANI2, P13 to P17)
0	1	0	0	4 channels (ANI0 to ANI3, P14 to P17)
0	1	0	1	5 channels (ANI0 to ANI4, P15 to P17)
0	1	1	0	6 channels (ANI0 to ANI5, P16, P17)
0	1	1	1	7 channels (ANI0 to ANI6, P17)
1	0	0	0	8 channels (ANI0 to ANI7)
Other than above				Setting prohibited

13.4 A/D Converter Operations

13.4.1 Basic operations of A/D converter

[1] Set the number of analog input channels with the A/D converter input select register (ADIS).

[2] From among the analog input channels set with ADIS, select one channel for A/D conversion with the A/D converter mode register (ADM).

[3] Sample the voltage input to the selected analog input channel with the sample & hold circuit.

[4] Sampling for the specified period of time sets the sample & hold circuit to the hold state so that the circuit holds the input analog voltage until termination of A/D conversion.

[5] Bit 7 of successive approximation register (SAR) is set and the tap selector sets the series resistor string voltage tap to (1/2) AV_{REF}.

[6] The voltage difference between the series resistor string voltage tap and analog input is compared with a voltage comparator. If the analog input is larger than (1/2) AV_{REF}, the MSB of SAR remains set. If the input is smaller than (1/2) AV_{REF}, the MSB is reset.

[7] Next, bit 6 of SAR is automatically set and the operation proceeds to the next comparison. In this case, the series resistor string voltage tap is selected according to the preset value of bit 7 as described below.

- Bit 7 = 1: (3/4) AV_{REF}
- Bit 7 = 0: (1/4) AV_{REF}

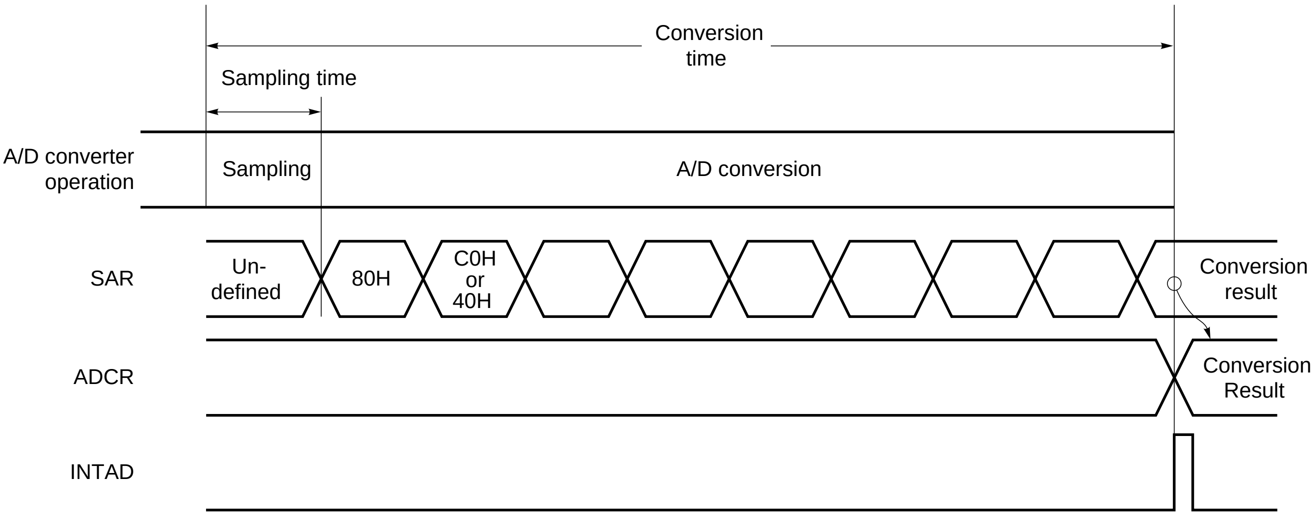
The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated with the result as follows.

- Analog input voltage ≥ Voltage tap: Bit 6 = 1
- Analog input voltage ≤ Voltage tap: Bit 6 = 0

[8] Comparison of this sort continues up to bit 0 of SAR.

[9] Upon completion of the comparison of 8 bits, any effective digital resultant value remains in SAR and the resultant value is transferred to and latched in the A/D conversion result register (ADCR).
At the same time, the A/D conversion termination interrupt request (INTAD) can also be generated.

Figure 13-4. A/D Converter Basic Operation



A/D conversion operations are performed continuously until the CS bit is reset (to 0) by software.

If a write to the ADM register is performed during an A/D conversion operation, the conversion operation is initialized, and if the CS bit is set (to 1), conversion starts again from the beginning.

After $\overline{\text{RESET}}$ input, the value of ADCR is undefined.

13.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (the value stored in the AD conversion result register (ADCR)) is shown by the following expression.

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

or

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

Remarks 1. INT ( ) : Function which returns integer parts of value in parentheses.

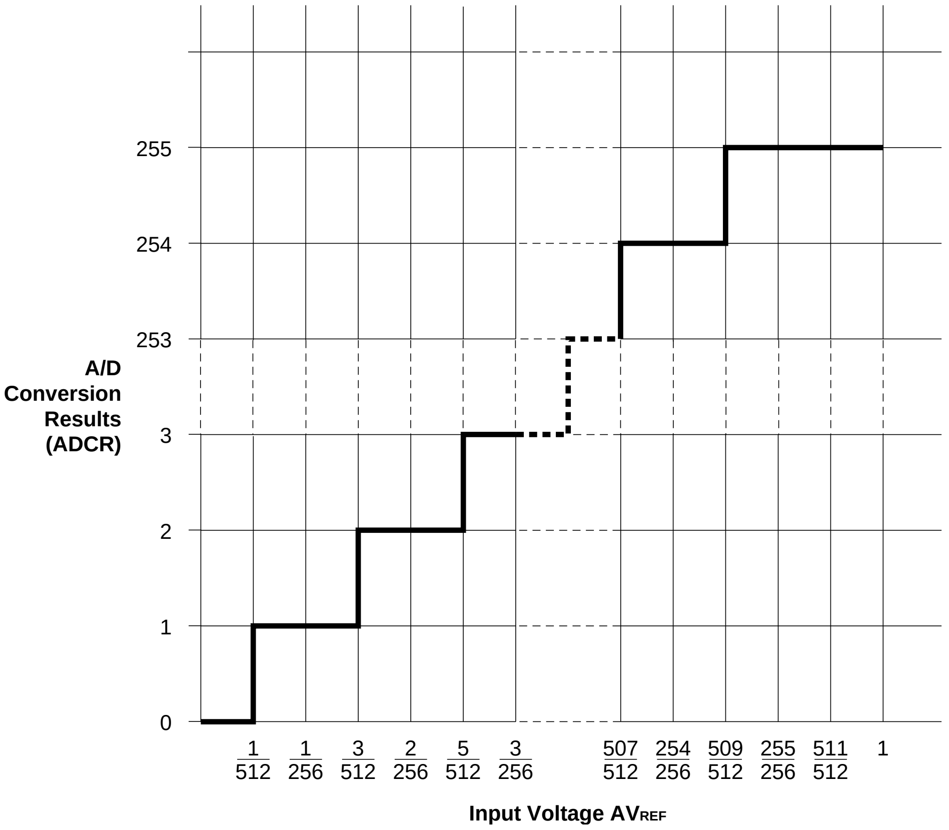
2. V_{IN} : Analog input voltage

3. AV_{REF} : AV_{REF} pin voltage

4. ADCR : A/D conversion result register (ADCR) value

Figure 13-5 shows the relation between the analog input voltage and the A/D conversion result.

Figure 13-5. Relation between Analog Input Voltage and A/D Conversion Result



13.4.3 A/D converter operating mode

One analog input channel is selected from among ANI0 to ANI7 with the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and A/D conversion is started.

The following two ways are available to start A/D conversion.

- Hardware start: Conversion is started by trigger input (INTP3).
- Software start: Conversion is started by setting ADM.

The A/D conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is simultaneously generated.

(1) A/D conversion by hardware start

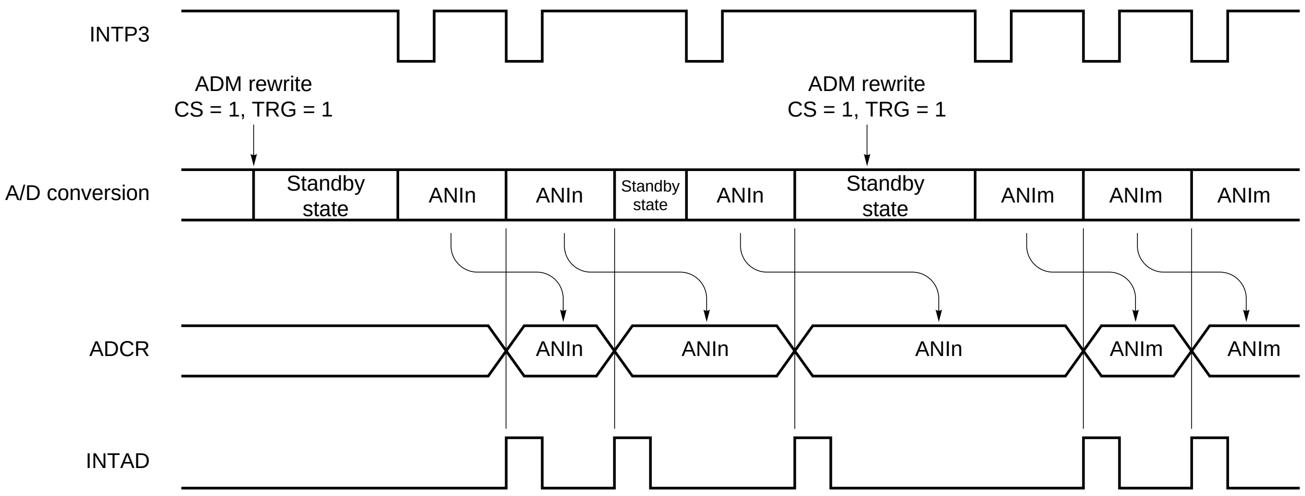
When bit 6 (TRG) and bit 7 (CS) of ADM are set to 1, the A/D conversion standby state is set. When the external trigger signal (INTP3) is input, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of the A/D converter mode register (ADM).

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, another operation is not started until a new external trigger signal is input.

If data with CS set to 1 is written to ADM again during A/D conversion, the converter suspends its A/D conversion operation and waits for a new external trigger signal to be input. When the external trigger input signal is reinput, A/D conversion is carried out from the beginning.

If data with CS set to 0 is written to ADM during A/D conversion, the A/D conversion operation stops immediately.

Figure 13-6. A/D Conversion by Hardware Start



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

(2) A/D conversion by software start

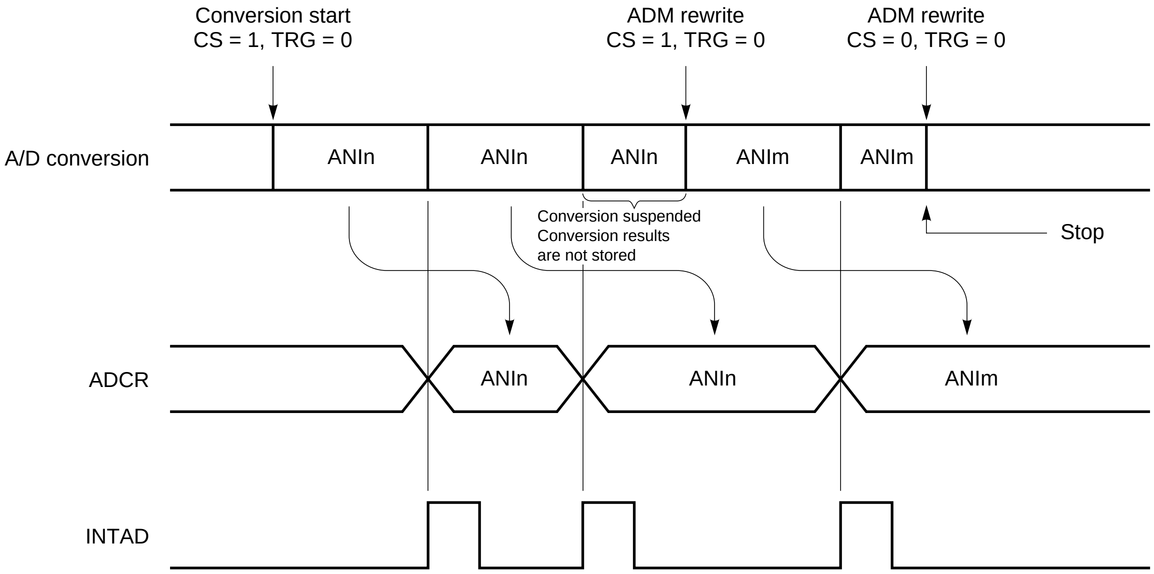
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

Upon termination of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR) and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and terminated, the next A/D conversion operation starts immediately. The A/D conversion operation continues repeatedly until new data is written to ADM.

If data with CS set to 1 is written to ADM again during A/D conversion, the converter suspends its A/D conversion operation and starts A/D conversion on the newly written data.

If data with CS set to 0 is written to ADM during A/D conversion, the A/D conversion operation stops immediately.

Figure 13-7. A/D Conversion by Software Start



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

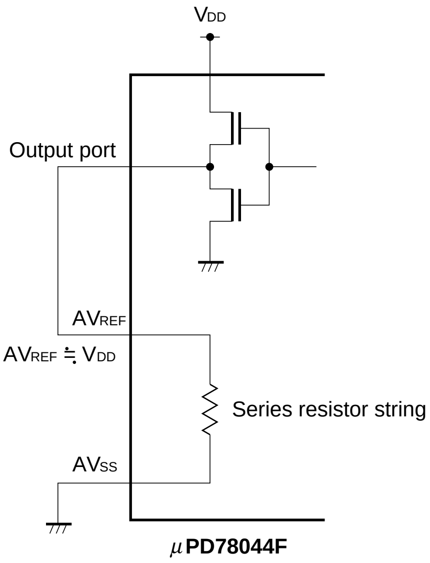
13.5 A/D Converter Precautions

(1) Current consumption in standby mode

The A/D converter operates on the main system clock. Therefore, its operation stops in STOP mode or in HALT mode with the subsystem clock. As a current still flows in the AV_{REF} pin at this time, this current must be cut in order to minimize the overall system power dissipation.

In this example, the power dissipation can be reduced if a low level is output to the output port in the standby mode. However, the actual AV_{REF} voltage is not so accurate and, accordingly, the converted value is not accurate and should be used for relative comparison only.

Figure 13-8. Example of Method of Reducing Current Consumption in Standby Mode



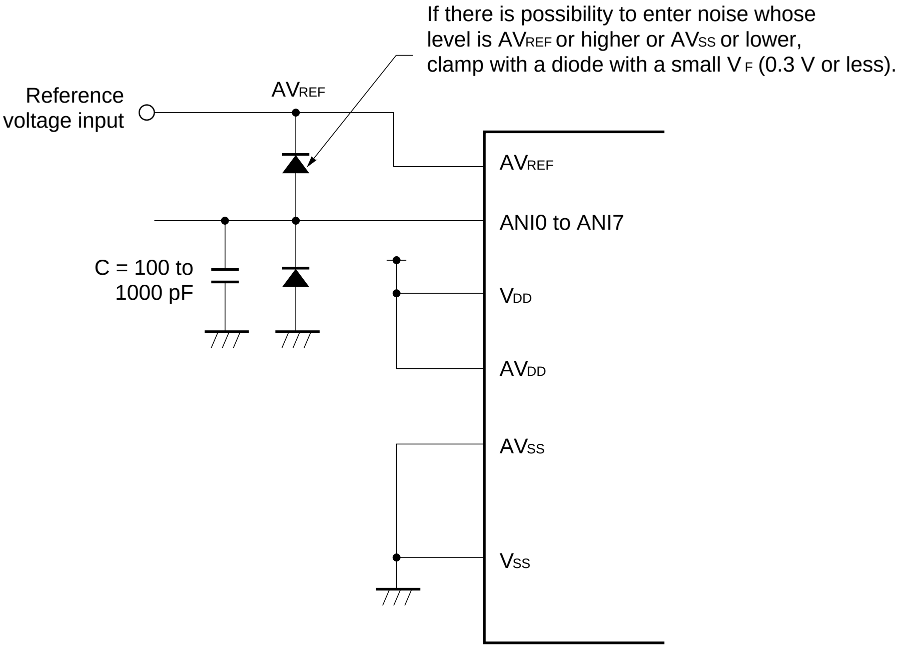
(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage above AV_{REF} or below AV_{SS} is input (even if within the absolute maximum rating range), the conversion value for that channel will be indeterminate. The conversion values of the other channels may also be affected.

(3) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on pins AV_{REF} and ANI0 to ANI7. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor is connected externally as shown in Figure 13-9 in order to reduce noise.

Figure 13-9. Analog Input Pin Disposition


If there is possibility to enter noise whose level is AV_{REF} or higher or AV_{SS} or lower, clamp with a diode with a small V_F (0.3 V or less).
Reference voltage input
AV_{REF}
C = 100 to 1000 pF
AV_{REF}
ANI0 to ANI7
V_{DD}
AV_{DD}
AV_{SS}
V_{SS}

(4) Pins ANI0/P10 to ANI7/P17

The analog input pins ANI0 to ANI7 also function as input/output port (PORT1) pins. Pins used as the analog input should be specified to the input mode. When A/D conversion is performed with any of pins To execute A/D conversion by selecting any of ANI0 to ANI7, do not execute an instruction that inputs data to PORT 1 during conversion, as this may reduce the conversion resolution.

Also, if digital pulses are applied to a pin adjacent to the pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

(5) AV_{REF} pin input impedance

A series resistor string of approximately 10 kΩ is connected between the AV_{REF} pin and the AV_{SS} pin.

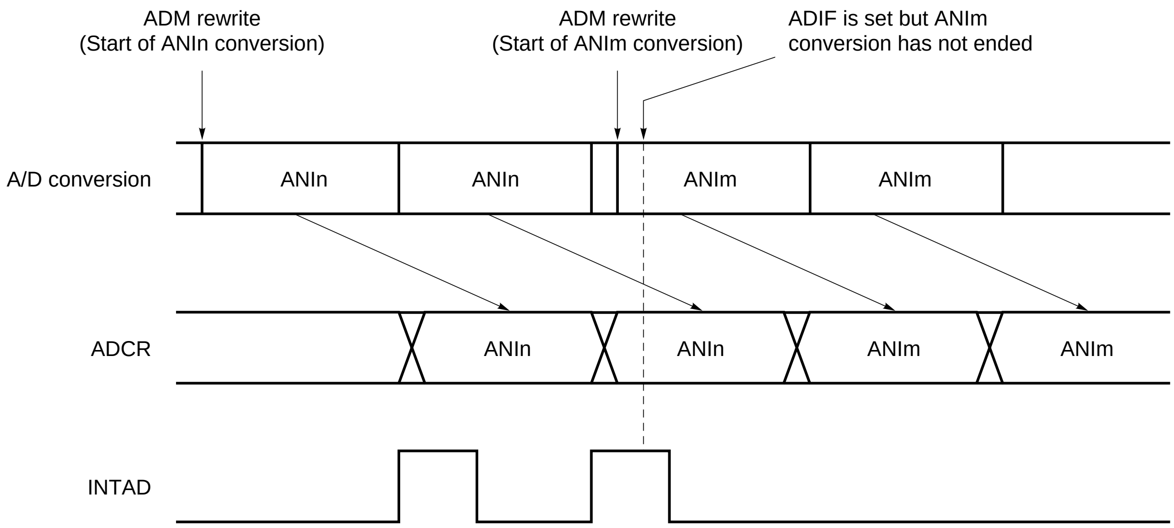
Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{REF} pin and the AV_{SS} pin, and there will be a large reference voltage error.

(6) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the A/D converter mode register (ADM) is changed. Caution is therefore required since, if a change of analog input pin is performed during A/D conversion, the ADIF and the A/D conversion result for the pre-change analog input may be set just before the ADM rewrite, and when ADIF is read immediately after the ADM rewrite, ADIF may be set despite the fact that the A/D conversion for the post-change analog input has not ended. (Refer to Figure 13-10.)

When the A/D conversion is stopped, the ADIF must be cleared before restarting.

Figure 13-10. A/D Conversion End Interrupt Request Generation Timing



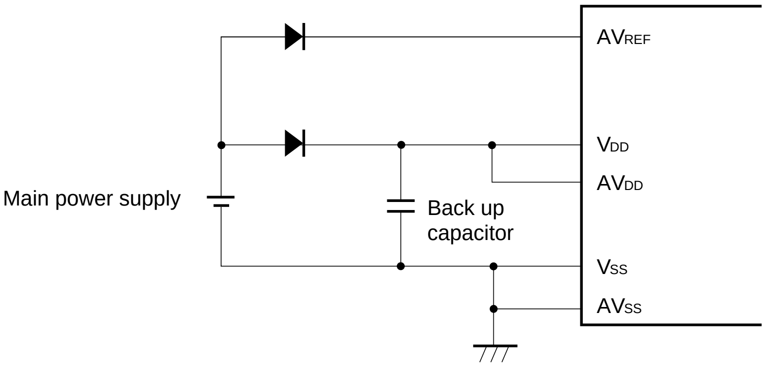
Remark n = 0, 1, ... , 7
m = 0, 1, ... , 7

(7) AV$_{DD}$ pin

The AV$_{DD}$ pin is the analog circuit power supply pin, and supplies power to the input circuits of ANI0/P10 to ANI7/P17.

Therefore, be sure to apply the voltage at the same level as V$_{DD}$ as shown in Figure 13-11 even in an application where the power supply is switched to the back-up power supply.

Figure 13-11. AV$_{DD}$ Pin Connection



CHAPTER 14 SERIAL INTERFACE CHANNEL 0

The μPD78044F Subseries incorporates two channels of clock synchronous serial interfaces.

Differences between channels 0 and 1 are as follows (Refer to **CHAPTER 15 SERIAL INTERFACE CHANNEL 1** for details of the serial interface channel 1).

Table 14-1. Differences between Channels 0 and 1

Serial Transfer Mode		Channel 0	Channel 1
3-wire serial I/O	Clock selection	$f_X/2^2$, $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, $f_X/2^8$, $f_X/2^9$ external clock, TO2 output	$f_X/2^2$, $f_X/2^3$, $f_X/2^4$, $f_X/2^5$, $f_X/2^6$, $f_X/2^7$, $f_X/2^8$, $f_X/2^9$ external clock, TO2 output
	Transfer method	MSB/LSB switchable as the start bit	MSB/LSB switchable as the start bit Automatic transmit/ receive function
	Transfer end flag	Serial transfer end interrupt request flag (CSIIF0)	Serial transfer end interrupt request flag (CSIIF1)
SBI (serial bus interface)		Use possible	None
2-wire serial I/O			

14.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following four modes.

Table 14-2. Difference of Serial Interface Channel 0 Mode

Operating Mode	Used Pin	Features	Usage
Operation stop mode	–	• Used when serial transfer is not carried. • Power consumption can be reduced.	–
3-wire serial I/O mode	$\overline{\text{SCK0}}$ (serial clock), SO0 (serial output), SI0 (serial input)	• Input and output lines are independent and they can transfer/receive at the same time, so the data transfer processing time is fast. • Since the start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB.	These modes are valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K and 17K series.
SBI mode	$\overline{\text{SCK0}}$ (serial clock), SB0 or SB1 (serial data bus)	• Enables configuration of serial bus with two signal lines, thus, even when connected to some microcontrollers, the number of ports can be cut and the wiring and drawing around on a board reduced. • High-speed serial interface complying with the NEC standard bus format. • Address, command and data information onto the serial bus • The wake-up function for handshake and the output function of acknowledge signal and busy signal can also be used.	
2-wire serial I/O mode	$\overline{\text{SCK0}}$ (serial clock), SB0 or SB1 (serial data bus)	• Enables to cope with any data transfer format by program, thus, the handshake lines previously necessary for connection of two or more devices can be removed.	

★ **Caution Do not change the operating 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 operating mode, stop the serial operation once.**

14.2 Serial Interface Channel 0 Configuration

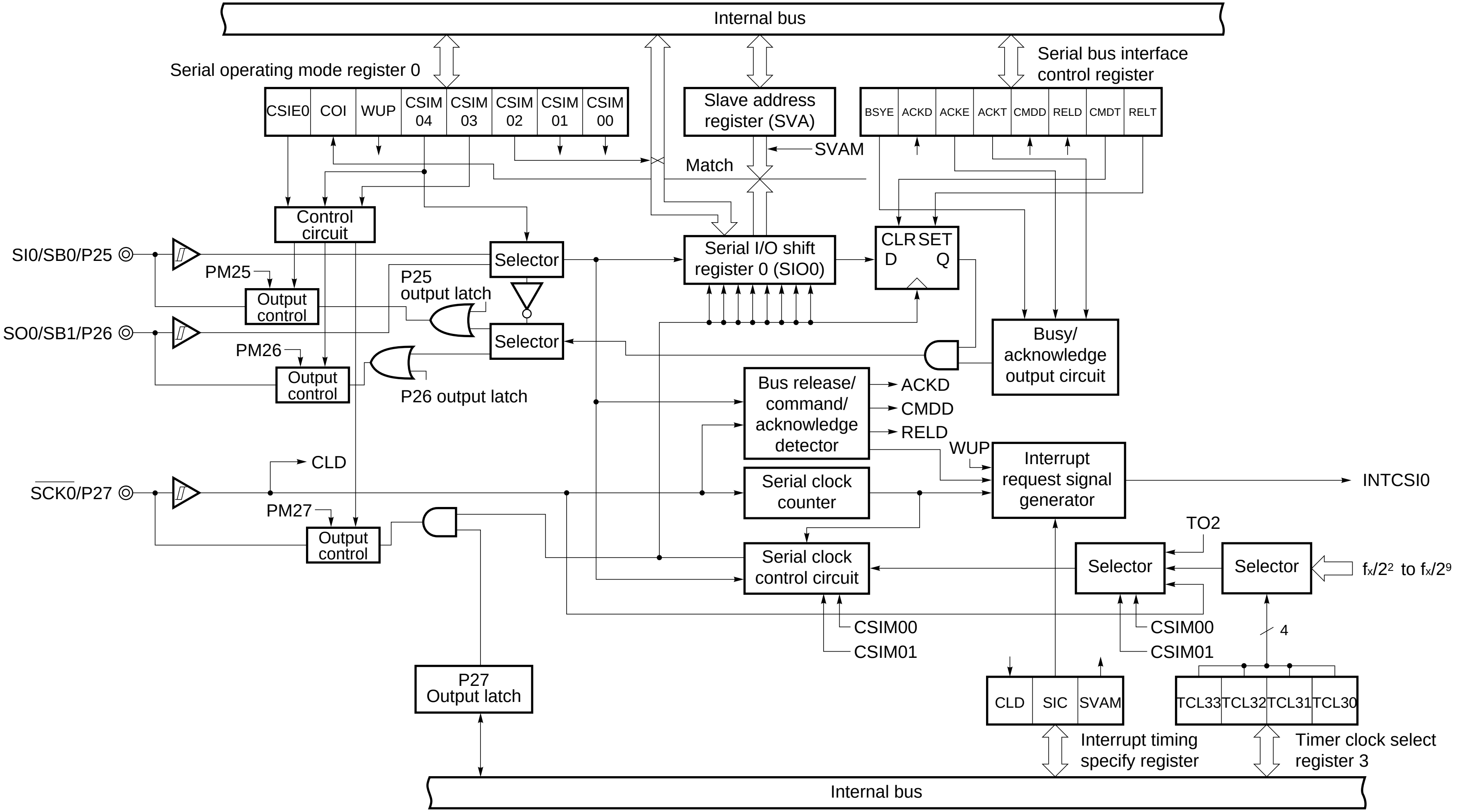
Serial interface channel 0 consists of the following hardware.

Table 14-3. Serial Interface Channel 0 Configuration

Item	Configuration
Register	Serial I/O shift register 0 (SIO0) Slave address register (SVA)
Control register	Timer clock select register 3 (TCL3) Serial operating mode register 0 (CSIM0) Serial bus interface control register (SBIC) Interrupt timing specify register (SINT) Port mode register 2 (PM2) Note

Note Refer to **Figure 4-5 P20, P21, P23 to P26 Block Diagram** and **Figure 4-6 P22 and P27 Block Diagram**.

Figure 14-1. Serial Interface Channel 0 Block Diagram



Remark Output Control performs selection between CMOS output and N-ch open-drain output.

(1) Serial I/O shift register 0 (SIO0)

This is an 8-bit register to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO0 is set with an 8-bit memory manipulation instruction.

When bit 7 (CSIE0) of serial operating mode register 0 (CSIM0) is 1, writing data to SIO0 starts serial operation.

In transmission, data written to SIO0 is output to the serial output (SO0) or serial data bus (SB0/SB1).

In reception, data is read from the serial input (SI0) or SB0/SB1 to SIO0.

The SBI mode and 2-wire serial I/O mode bus configurations enables the pin to serve for both input and output. Thus, in the case of a device for reception, write FFH to SIO0 in advance (except when address reception is carried out by setting bit 5 (WUP) of CSIM0 to 1).

In the SBI mode, the busy state can be cleared by writing data to SIO0. In this case, bit 7 (BSYE) of the serial bus interface control register (SBIC) is not cleared to 0.

$\overline{\text{RESET}}$ input makes SIO0 undefined.

(2) Slave address register (SVA)

This is an 8-bit register to set the slave address value for connection of a slave device to the serial bus. SVA is set with an 8-bit memory manipulation instruction. This register is not used in the 3-wire serial I/O mode.

The master device outputs a slave address for selection of a particular slave device to the connected slave device. These two data (the slave address output from the master device and the SVA value) are compared with an address comparator. If they match, the slave device has been selected. In that case, bit 6 (COI) of serial operating mode register 0 (CSIM0) becomes 1.

Address comparison can also be executed on the data of LSB-masked high-order 7 bits by setting bit 4 (SVAM) of the interrupt timing specify register (SINT) to 1.

★ If no matching is detected in address reception, 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 this case, the interrupt request signal (INTCSI0) is generated only when the slave address output by the master coincides with the value of SVA, and it can be learned by this interrupt request that the master requests for communication. If bit 5 (SIC) of the interrupt timing specify register (SINT) is set to 1, the wake-up function cannot be used even if WUP is set to 1 (an interrupt request signal is generated when bus release is detected). To use the wake-up function, clear SIC to 0.

Further, an error can be detected by using SVA when the device transmits data as master or slave device in the SBI or 2-wire serial I/O mode.

$\overline{\text{RESET}}$ input makes SVA undefined.

(3) SO0 latch

This latch holds SI0/SB0/P25 and SO0/SB1/P26 pin levels. It can be directly controlled by software.

In the SBI mode, this latch is set upon termination of the 8th serial clock.

(4) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

(5) Serial clock control circuit

This circuit controls serial clock supply to the serial I/O shift register 0 (SIO0). When the internal system clock is used, the circuit also controls clock output to the $\overline{\text{SCK0}}$/P27 pin.

(6) Interrupt request signal generator

This circuit controls interrupt request signal generation. It generates the interrupt request signal in the following cases.

- **In the 3-wire serial I/O mode and 2-wire serial I/O mode**
  This circuit generates an interrupt request signal every eight serial clocks.

- **In the SBI mode**
  When WUP[Note] is 0 .... Generates an interrupt request signal every eight serial clocks.
  When WUP[Note] is 1 .... Generates an interrupt request signal when the serial I/O shift register 0 (SIO0) value matches the slave address register (SVA) value after address reception.

★ **Note** WUP is a wake-up function specify bit. It is bit 5 of serial operating mode register 0 (CSIM0). To use the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

(7) Busy/acknowledge output circuit and bus release/command/acknowledge detector

These two circuits output and detect various control signals in the SBI mode.
These do not operate in the 3-wire serial I/O mode and 2-wire serial I/O mode.

(8) P27 output latch

This latch generates a serial clock by software after termination of eight serial clocks.
When using serial interface channel 0, set the P27 output latch to 1.
$\overline{\text{RESET}}$ input clears the latch to 0.

14.3 Serial Interface Channel 0 Control Registers

The following four types of registers are used to control serial interface channel 0.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 0 (CSIM0)
- Serial bus interface control register (SBIC)
- Interrupt timing specify register (SINT)

(1) Timer clock select register 3 (TCL3) (See **Figure 14-2**.)
This register sets the serial clock of serial interface channel 0.
TCL3 is set with an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Remark TCL3 has functions to set the serial clock of serial interface channel 1 except to set the serial clock of serial interface channel 0.

(2) Serial operating mode register 0 (CSIM0) (See **Figure 14-3**.)
This register sets serial interface channel 0 serial clock, operating mode, operation enable/stop wake-up function and displays the address comparator match signal.
CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

★ **Caution Do not change the operating 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 operating mode, stop the serial operation once.**

Figure 14-2. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

Caution If a value which is not the same as the previous value must be written to TCL3, stop the serial transfer before writing the value.

Remarks 1. f_X: Main system clock frequency

2. Figures in parentheses apply to operation with f_X = 5.0 MHz.

Figure 14-3. Serial Operating Mode Register 0 Format (1/2)

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After 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

CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/P25 Pin Function	SO0/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 input/output)
		1								LSB			
1	0	0	Note 3 ×	Note 3 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (CMOS input/output)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/ output)	
1	1	0	Note 3 ×	Note 3 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (N-ch open-drain input/ output)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/output)	

R/W

WUP	Wake-up Function Control Note 4
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register in SBI mode

Notes **1.** Bit 6 (COI) is a Read-Only bit.

2. Can be used as P25 (CMOS input) when used only for transmission.

3. Can be used freely as port function.

★ **4.** To use the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

Remark ×: don't care
PMxx: Port mode register
Pxx: Port output latch

Figure 14-3. Serial Operating Mode Register 0 Format (2/2)

R	COI	Slave Address Comparison Result Flag [Note]
	0	Slave address register not equal to serial I/O shift register 0 data
	1	Slave address register equal to serial I/O shift register 0 data

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enable

Note COI becomes 0 when CSIE0 = 0.

(3) Serial bus interface control register (SBIC)

This register sets serial bus interface operation and displays statuses.

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SBIC to 00H.

Figure 14-4. Serial Bus Interface Control Register Format (1/2)

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W [Note]

R/W	RELT	Use for bus release signal output. When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	Use for command signal output. When CMDT = 1, SO latch is cleared to (0). After SO latch clearance, automatically cleared to (0). Also cleared to (0) when CSIE0 = 0.

R	RELD	Bus Release Detection
	Clear Conditions (RELD = 0)	Set Conditions (RELD = 1)
	• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When bus release signal (REL) is detected

R	CMDD	Command Detection
	Clear Conditions (CMDD = 0)	Set Conditions (CMDD = 1)
	• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When command signal (CMD) is detected

R/W	ACKT	Acknowledge signal is output in synchronization with the falling edge clock of $\overline{\text{SCK0}}$ just after execution of the instruction to be set to 1, and after acknowledge signal output, automatically cleared to 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

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

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

Figure 14-4. Serial Bus Interface Control Register Format (2/2)

R/W

ACKE	Acknowledge Signal Output Control	
0	Acknowledge signal automatic output disable (output with ACKT enable)	
1	Before completion of transfer	Acknowledge signal is output in synchronization with the 9th clock falling edge of $\overline{\text{SCK0}}$ (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

R

ACKD	Acknowledge Detection
Clear Conditions (ACKD = 0)	Set Conditions (ACKD = 1)
• At the falling edge of $\overline{\text{SCK0}}$ immediately after the busy mode has been released when a transfer start instruction is executed • When CSIE0 = 0 • When $\overline{\text{RESET}}$ input is applied	• When acknowledge signal ($\overline{\text{ACK}}$) is detected at the rising edge of $\overline{\text{SCK0}}$ clock after completion of transfer

R/W

BSYE[Note]	Synchronizing Busy Signal Output Control
0	Disables busy signal which is output in synchronization with the falling edge of $\overline{\text{SCK0}}$ clock just after execution of the instruction to be cleared to 0.
1	Outputs busy signal at the falling edge of $\overline{\text{SCK0}}$ clock following the acknowledge signal.

Note Busy mode can be cleared by start of serial interface transfer or reception of address signal. However, BSYE flag is not cleared to 0.

Remarks 1. Bits 0, 1, and 4 (RELD, CMDT, and ACKT) are 0 when read after data setting.

2. CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

(4) Interrupt timing specify register (SINT)

This register sets the bus release interrupt and address mask functions and displays the $\overline{\text{SCK0}}$ pin level status.

SINT is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SINT to 00H.

Figure 14-5. Interrupt Timing Specify Register Format

Symbol	7	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W Note 1

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Factor Selection Note 2
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ Pin Level Note 3
0	Low Level
1	High Level

Notes 1. Bit 6 (CLD) is a Read-Only bit.

2. When using wake-up function, set SIC to 0.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remarks 1. SVA: Slave address register

2. CSIIF0: Interrupt request flag corresponding to INTCSI0

3. CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

14.4 Serial Interface Channel 0 Operations

The following four operating modes are available to the serial interface channel 0.

- Operation stop mode
- 3-wire serial I/O mode
- SBI mode
- 2-wire serial I/O mode

14.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 0 (SIO0) does not carry out shift operation either and thus it can be used as an ordinary 8-bit register.

In the operation stop mode, the P25/SI0/SB0, P26/SO0/SB1 and P27/$\overline{\text{SCK0}}$ pins can be used as ordinary input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 0 (CSIM0).
CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	CSIM00	FF60H	00H	R/W

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enable

14.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K series.

Communication is carried out with three lines of serial clock ($\overline{\text{SCK0}}$), serial output (SO0), and serial input (SI0).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0) and serial bus interface control register (SBIC).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After 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

CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W

CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/P25 Pin Function	SO0/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 input/output)
		1								LSB			
1	0	SBI mode (Refer to **14.4.3 SBI mode operation**)											
1	0	2-wire serial I/O mode (Refer to **14.4.4 2-wire serial I/O mode operation**)											

R/W

WUP	Wake-up Function Control Note 3
0	Interrupt request signal generation with each serial transfer in any mode
1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register in SBI mode

R

COI	Slave Address Comparison Result Flag Note 4
0	Slave address register not equal to serial I/O shift register 0 data
1	Slave address register equal to serial I/O shift register 0 data

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Operation stopped
1	Operation enable

Notes **1.** Bit 6 (COI) is a Read-Only bit.

2. Can be used as P25 (CMOS input) when used only for transmission.

★ **3.** Be sure to set WUP to 0 when the 3-wire serial I/O mode is selected.

4. When CSIE0 = 0, COI becomes 0.

Remarks **1.** ×: don't care

2. PMxx: Port mode register

3. Pxx: Port output latch

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SBIC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W	RELT	When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	When CMDT = 1, SO latch is cleared to 0. After SO latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

Bit 7 of the serial operating mode register 0 (CSIM0)

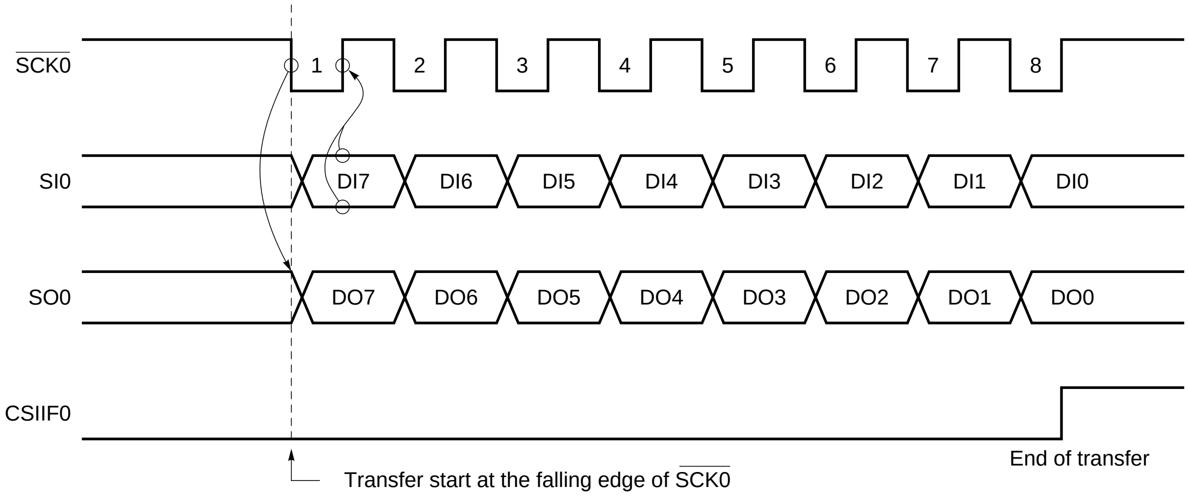
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization of the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of the serial clock ($\overline{\text{SCK0}}$). The transmitted data is held in the SO0 latch and is output from the SO0 pin. The received data input to the SIO pin is latched in SIO0 at the rising edge of $\overline{\text{SCK0}}$.

Upon termination of 8-bit transfer, SIO0 operation stops automatically and the interrupt request flag (CSIIF0) is set.

Figure 14-6. 3-Wire Serial I/O Mode Timings



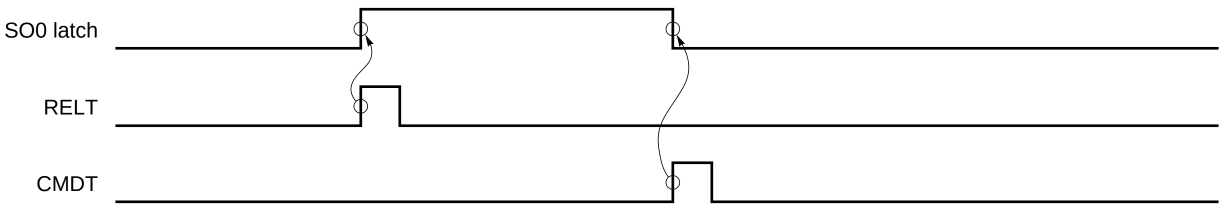
The SO0 pin serves for CMOS output and generates the SO0 latch status. Thus, the SO0 pin output status can be manipulated by setting bits 0 (RELT) and 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **14.4.5 $\overline{\text{SCK0}}$ pin output manipulation**).

(3) Various signals

Figure 14-7 shows RELT and CMDT operations.

Figure 14-7. RELT and CMDT Operations



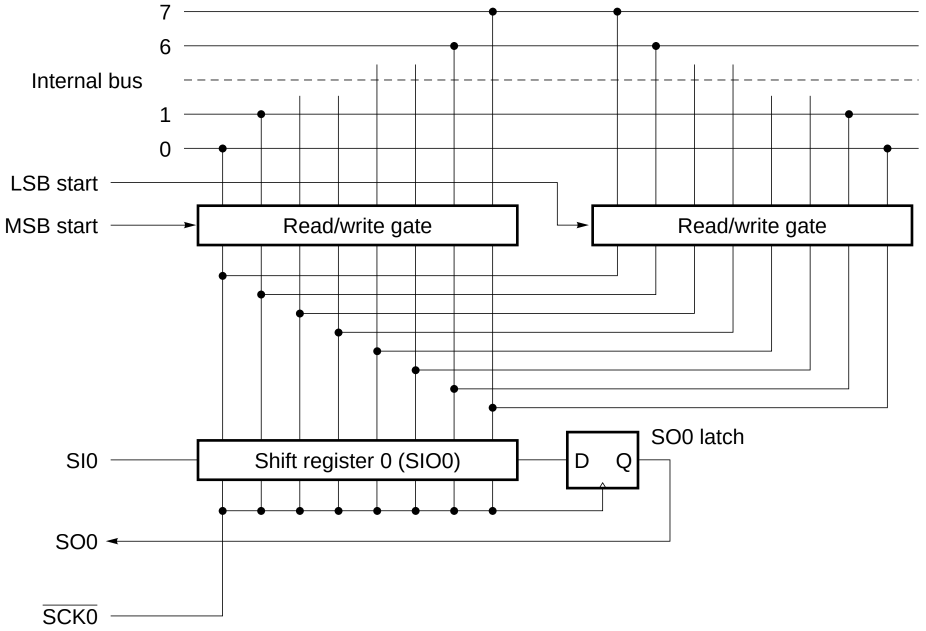
(4) MSB/LSB switching as the start bit

The 3-wire serial I/O mode enables to select transfer to start at MSB or LSB.

Figure 14-8 shows the configuration of the serial I/O shift register 0 (SIO0) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 2 (CSIM02) of the serial operating mode register 0 (CSIM0).

Figure 14-8. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO0. The SIO0 shift order remains unchanged.

Thus, switch the MSB/LSB start bit before writing data to the shift register.

(5) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0) = 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is a high level after 8-bit serial transfer.

Caution If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

14.4.3 SBI mode operation

SBI (Serial Bus Interface) is a high-speed serial interface that complies with the NEC serial bus format.

SBI has a format with the bus configuration function added to the clocked serial I/O method so that it can carry out communication with two or more devices with two signal conductors on the single-master high-speed serial bus. Thus, when making up a serial bus with two or more microcomputers and peripheral ICs, the number of ports to be used and the number of wires on the board can be decreased.

The master device can output to the serial data bus of the slave device “addresses” for selection of the serial communication target device, “commands” to instruct the target device and actual “data”. The slave device can identify the received data into “address”, “command” or “data”, by hardware. This function helps simplify the application program that controls serial interface channel 0.

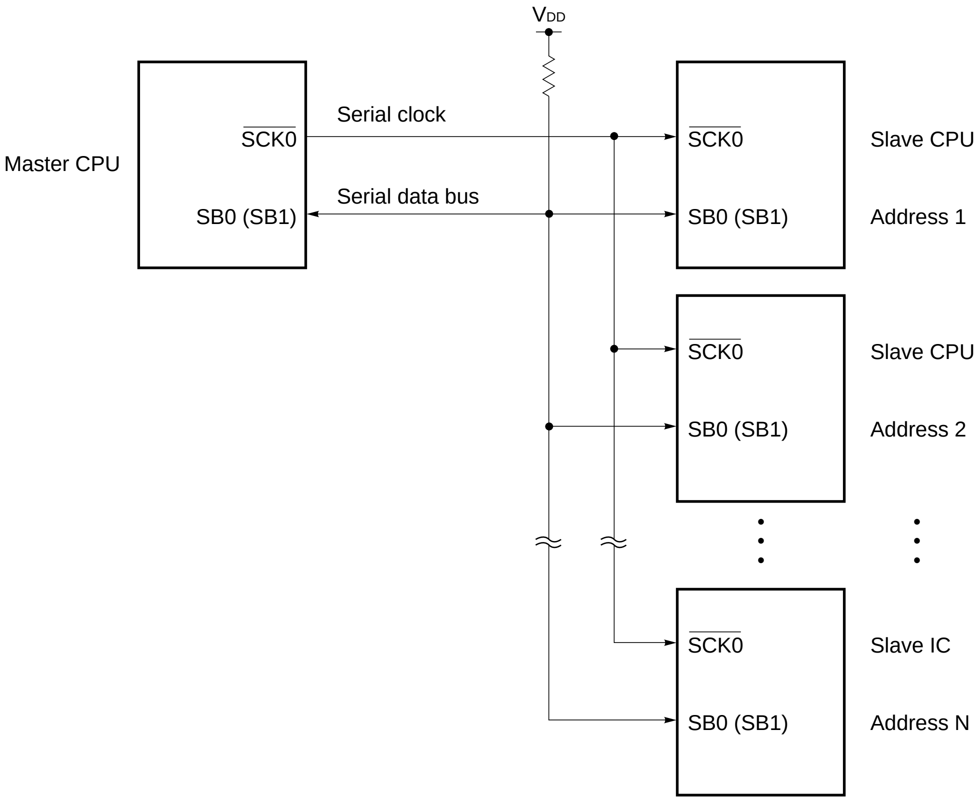
The SBI function is incorporated into various devices including 75X/XL-Series 78K-Series devices.

Figure 14-9 shows a serial bus configuration example when a CPU having a serial interface compliant with SBI and peripheral ICs are used.

In SBI, the SB0 (SB1) serial data bus pin serves for open-drain output and so the serial data bus line is in wired-OR state. A pull-up resistor is necessary for the serial data bus line.

Refer to **(11) SBI mode precautions (d)** described later when the SBI mode is used.

Figure 14-9. Example of Serial Bus Configuration with SBI



Caution   When replacing the master CPU/slave CPU, a pull-up resistor is necessary for the serial clock line ($\overline{SCK0}$) as well because serial clock line ($\overline{SCK0}$) input/output switching is carried out asynchronously between the master and slave CPUs.

(1) SBI functions

In the conventional serial I/O method, when a serial bus is constructed by connecting two or more devices, many ports and wiring are necessary to distinguish chip select signals and command/data and to judge the busy state because only the data transfer function is available. If these operations are to be controlled by software, the software must be heavily loaded.

In SBI, a serial bus can be constructed with two signal conductors of serial clock $\overline{\text{SCK0}}$ and serial data bus SB0 (SB1). Thus, SBI is effective to decrease the number of microcomputer ports and that of wirings and routings on the board.

The SBI functions are described below.

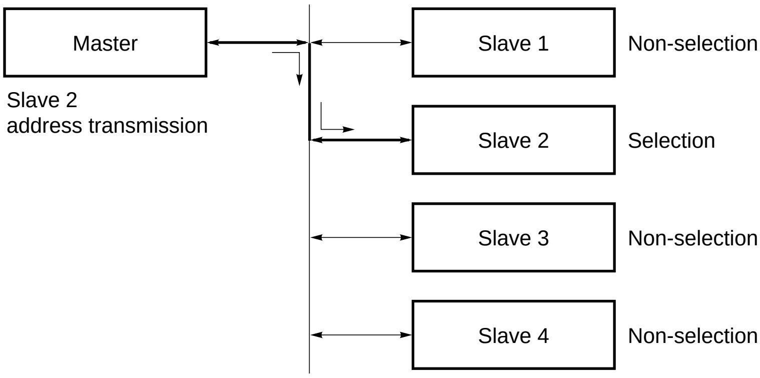
(a) Address/command/data identify function

Serial data is distinguished into addresses, commands and data.

(b) Chip select function by address transmission

The master executes slave chip selection by address transmission.

Figure 14-10. Slave Selection with Address



(c) Wake-up function

The slave can easily judge address reception (chip select judgment) with the wake-up function (which can be set/reset by software).

When the wake-up function is set, the interrupt request signal (CSIIF0) is generated upon reception of a match address. Thus, when communication is executed with two or more devices, the CPU except the selected slave devices can operate regardless of serial communication.

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

The acknowledge signal to check serial data reception is controlled.

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

The busy signal to report the slave busy state is controlled.

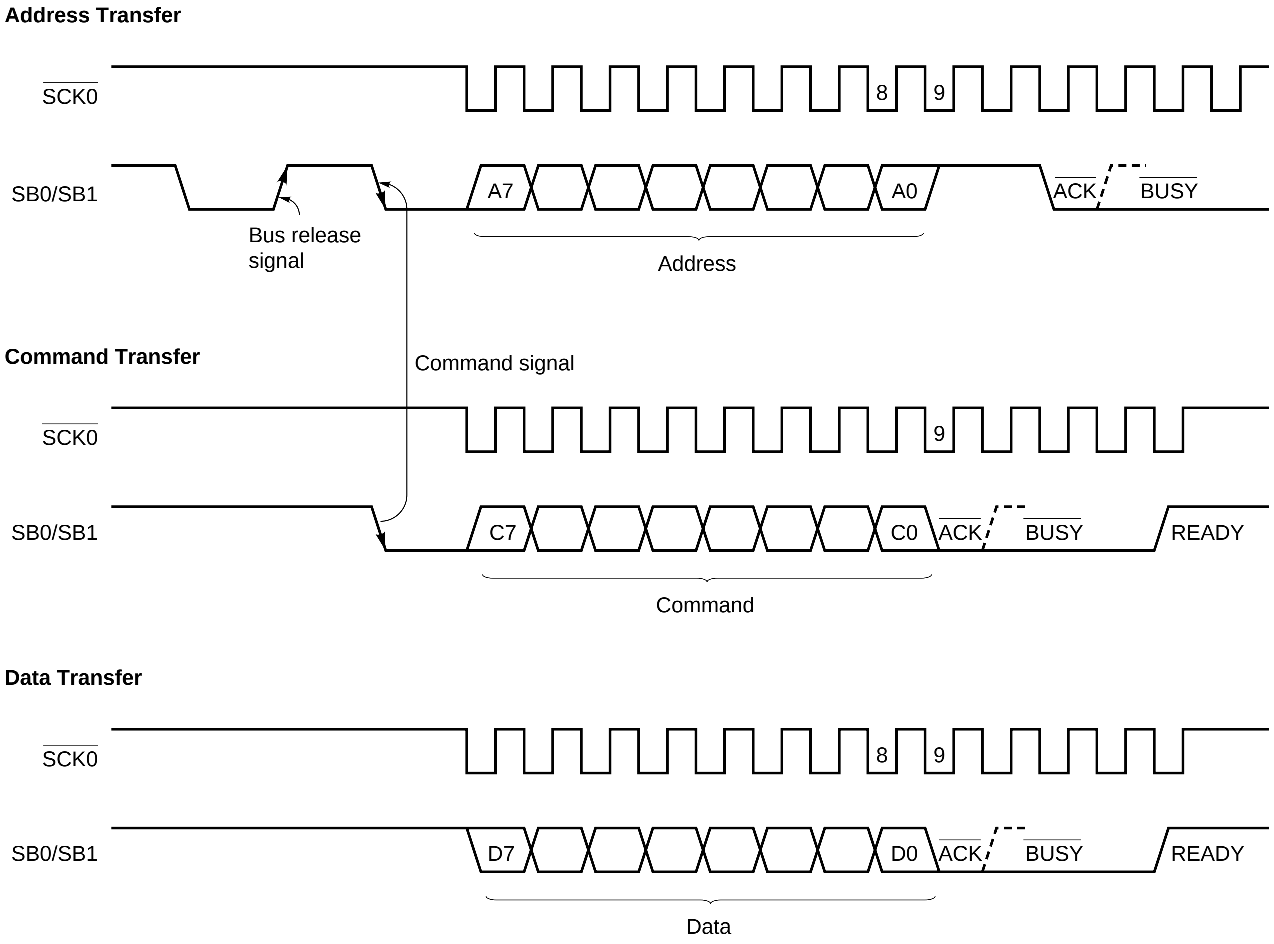
(2) SBI definition

The SBI serial data format is defined as follows.

Serial data to be transferred with SBI is distinguished into three types, “address”, “command” and “data”.

Figure 14-11 shows the address, command and data transfer timings.

Figure 14-11. SBI Transfer Timings



Remark The dotted line indicates READY status.

The bus release signal and the command signal are output by the master device. $\overline{\text{BUSY}}$ is output by the slave signal. $\overline{\text{ACK}}$ can be output by either the master or slave device (normally, the 8-bit data receiver outputs).

Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{\text{BUSY}}$ reset.

(3) Register setting

The SBI mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC) and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After 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	CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
	0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/P25 Pin Function	SO0/P26 Pin Function	$\overline{\text{SCK0}}$/P27 Pin Function
	0	×	3-wire serial I/O mode (Refer to **14.4.2 3-wire serial I/O mode operation**)											
	1	0	0	Note 2 ×	Note 2 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS input/output)	SB1 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (CMOS input/output)
			1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/ output)	
	1	1	2-wire serial I/O mode (Refer to **14.4.4 2-wire serial I/O mode operation**)											

R/W	WUP	Wake-up Function Control Note 3
	0	Interrupt request signal generation with each serial transfer in any mode
	1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register in SBI mode

R	COI	Slave Address Comparison Result Flag Note 4
	0	Slave address register not equal to serial I/O shift register 0 data
	1	Slave address register equal to serial I/O shift register 0 data

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enable

Notes 1. Bit 6 (COI) is a Read-Only bit.

2. Can be used freely as port function.

★ **3.** To use the wake-up function (WUP = 1), clear bit 5 (SIC) of the interrupt timing specify register (SINT) to 0.

4. When CSIE0 = 0, COI becomes 0.

Remark ×: don’t care

PMxx: Port mode register

Pxx: Port output latch

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.
RESET input clears SBIC to 00H.

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

R/W	RELT	Use for bus release signal output. When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	Use for command signal output. When CMDT = 1, SO latch is cleared to (0). After SO latch clearance, automatically cleared to (0). Also cleared to (0) when CSIE0 = 0.

R

RELD	Bus Release Detection
Clear Conditions (RELD = 0)	Set Conditions (RELD = 1)
• When transfer start instruction is executed • If SIO0 and SVA values do not match in address reception • When CSIE0 = 0 • When RESET input is applied	• When bus release signal (REL) is detected

R

CMDD	Command Detection
Clear Conditions (CMDD = 0)	Set Conditions (CMDD = 1)
• When transfer start instruction is executed • When bus release signal (REL) is detected • When CSIE0 = 0 • When RESET input is applied	• When command signal (CMD) is detected

R/W	ACKT	Acknowledge signal is output in synchronization with the falling edge clock of SCK0 just after execution of the instruction to be set to 1, and after acknowledge signal output, automatically cleared to 0. Used as ACKE = 0. Also cleared to 0 upon start of serial interface transfer or when CSIE0 = 0.

(continued)

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

Remarks 1. Bits 0, 1, and 4 (RELT, CMDT, and ACKT) are 0 when read after data setting.
2. CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

(continued)

R/W	ACKE	Acknowledge Signal Output Control	
	0	Acknowledge signal automatic output disable (output with ACKT enable)	
	1	Before completion of transfer	Acknowledge signal is output in synchronization with the 9th clock falling edge of $\overline{\mathrm{SCK0}}$ (automatically output when ACKE = 1).
		After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{\mathrm{SCK0}}$ just after execution of the instruction to be set to 1 (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output.

R	ACKD	Acknowledge Detection	
	Clear Conditions (ACKD = 0)		Set Conditions (ACKD = 1)
	• At the falling edge of $\overline{\mathrm{SCK0}}$ immediately after the busy mode has been released when a transfer start instruction is executed • When CSIE0 = 0 • When $\overline{\mathrm{RESET}}$ input is applied		• When acknowledge signal ($\overline{\mathrm{ACK}}$) is detected at the rising edge of $\overline{\mathrm{SCK0}}$ clock after completion of transfer

R/W	BSYE[Note]	Synchronizing Busy Signal Output Control
	0	Disables busy signal which is output in synchronization with the falling edge of $\overline{\mathrm{SCK0}}$ clock just after execution of the instruction to be cleared to 0.
	1	Outputs busy signal at the falling edge of $\overline{\mathrm{SCK0}}$ clock following the acknowledge signal.

Note Busy mode can be cleared by start of serial interface transfer. However, the BSYE flag is not cleared to 0.

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

(c) Interrupt timing specify register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears SINT to 00H.

Symbol	7	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W Note 1

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Factor Selection Note 2
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ Pin Level Note 3
0	Low Level
1	High Level

Notes **1.** Bit 6 (CLD) is a Read-Only bit.

2. When using wake-up function in the SBI mode, set SIC to 0.

3. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register

CSIIF0: Interrupt request flag corresponding to INTCSI0

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

(4) Various signals

Figures 14-12 to 14-17 show various signals and flag operations in SBI. Table 14-4 lists various signals in SBI.

Figure 14-12. RELT, CMDT, RELD and CMDD Operations (Master)

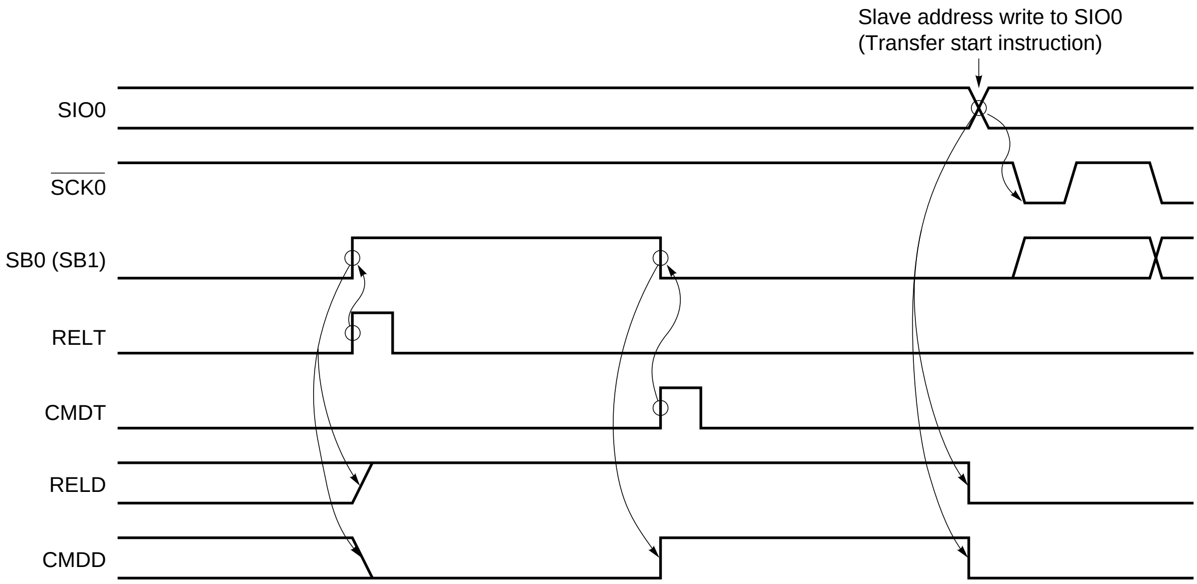


Figure 14-13. RELD and CMDD Operations (Slave)

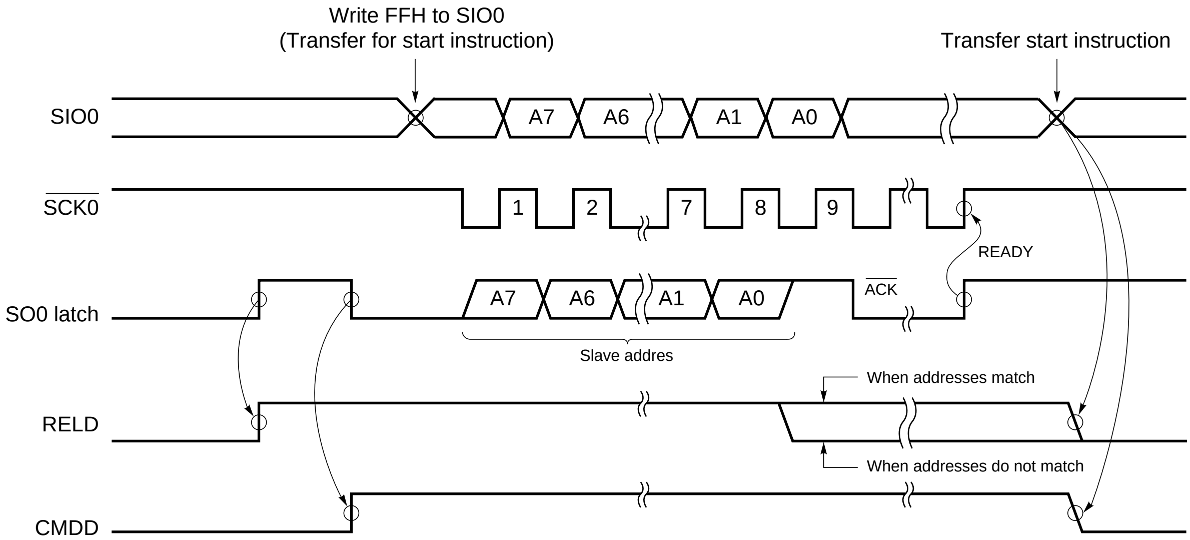
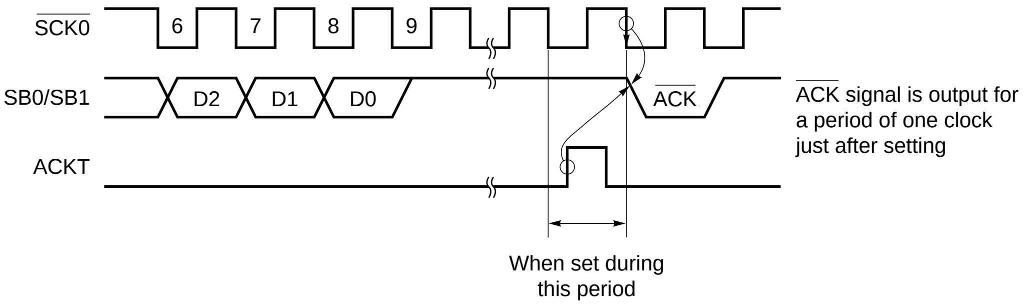
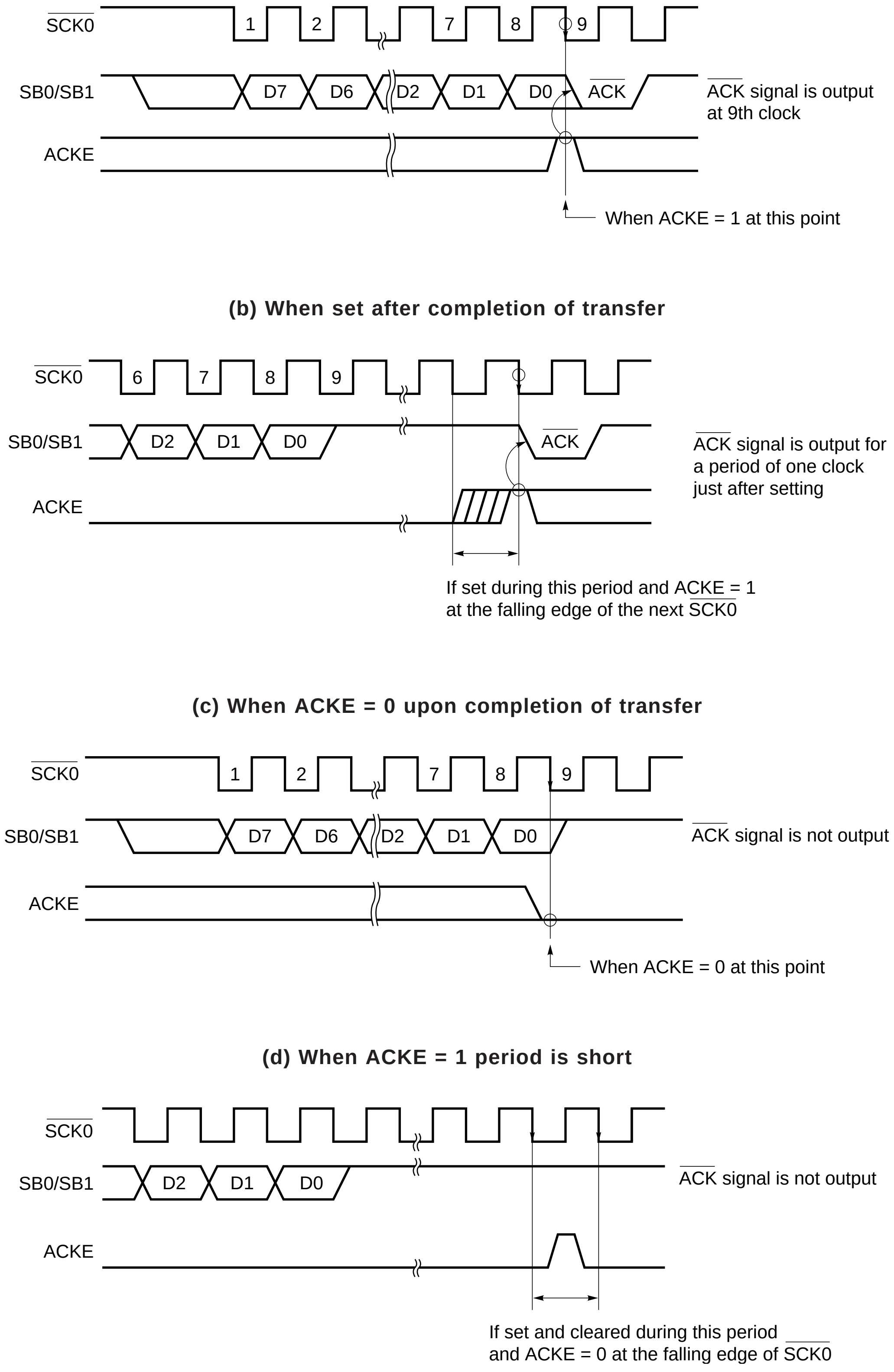


Figure 14-14. ACKT Operation


SCK0
6
7
8
9
SB0/SB1
D2
D1
D0
ACK
ACKT
ACK signal is output for a period of one clock just after setting
When set during this period

Caution Do not set ACKT before termination of transfer.

Figure 14-15. ACKE Operations



(a) When ACKE = 1 upon completion of transfer

(b) When set after completion of transfer

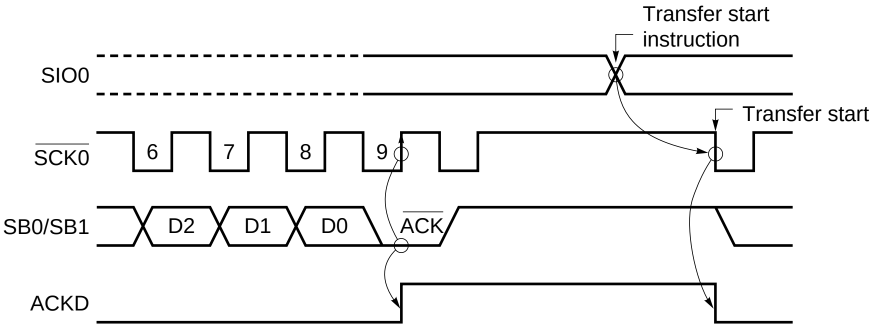
(c) When ACKE = 0 upon completion of transfer

(d) When ACKE = 1 period is short

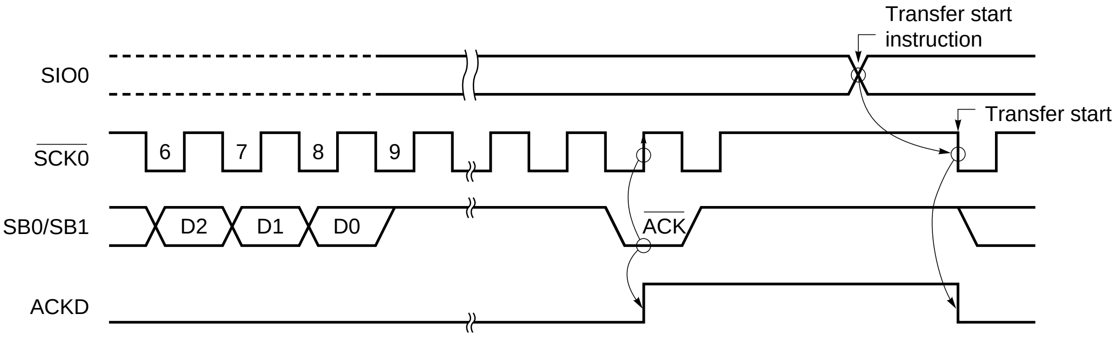
SCK0
1
2
7
8
9
SB0/SB1
D7
D6
D2
D1
D0
ACK
ACKE
ACK signal is output at 9th clock
When ACKE = 1 at this point
SCK0
6
7
8
9
SB0/SB1
D2
D1
D0
ACK
ACKE
ACK signal is output for a period of one clock just after setting
If set during this period and ACKE = 1 at the falling edge of the next SCK0
SCK0
1
2
7
8
9
SB0/SB1
D7
D6
D2
D1
D0
ACKE
ACK signal is not output
When ACKE = 0 at this point
SCK0
SB0/SB1
D2
D1
D0
ACKE
ACK signal is not output
If set and cleared during this period and ACKE = 0 at the falling edge of SCK0

Figure 14-16. ACKD Operations

(a) When $\overline{\text{ACK}}$ signal is output at 9th clock of $\overline{\text{SCK0}}$



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



(c) Clear timing when transfer start is instructed in BUSY

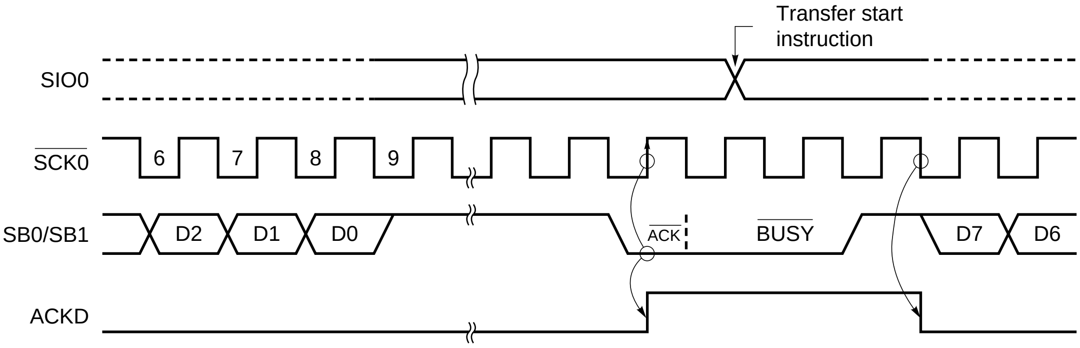


Figure 14-17. BSYE Operation

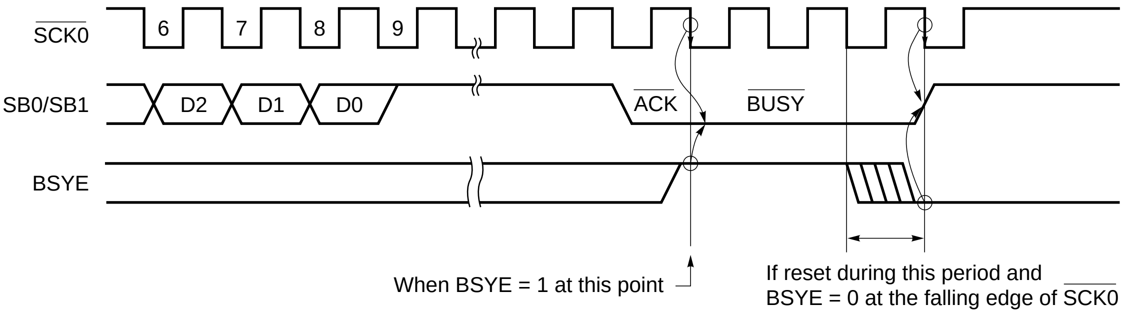

SCK0
6
7
8
9
SB0/SB1
D2
D1
D0
ACK
BUSY
BSYE
When BSYE = 1 at this point
If reset during this period and BSYE = 0 at the falling edge of SCK0

Table 14-4. Various Signals in SBI Mode (1/2)

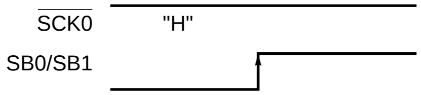
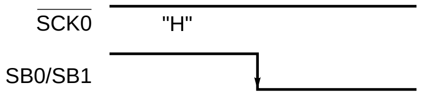
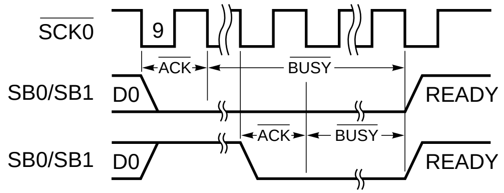
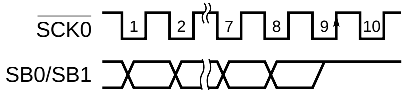
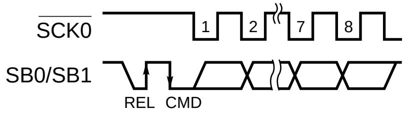
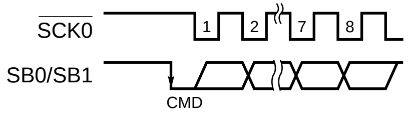
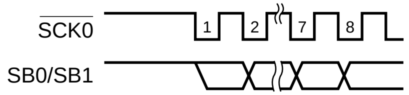
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0/SB1 rising edge when $\overline{\text{SCK0}}$ = 1		• RELT set	• RELD set • CMDD clear	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0/SB1 falling edge when $\overline{\text{SCK0}}$ = 1		• CMDT set	• CMDD set	i) Transmit data is an address after REL signal output. ii) REL signal is not output and transmit data is a command.
Acknowledge signal $(\overline{\text{ACK}})$	Master/ slave	Low-level signal to be output to SB0/SB1 during one-clock period of $\overline{\text{SCK0}}$ after completion of serial reception	[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] ACKT set	• ACKD set	Completion of reception
Busy signal $(\overline{\text{BUSY}})$	Slave	[Synchronous BUSY signal] Low-level signal to be output to SB0/SB1 following acknowledge signal		• BSYE = 1	—	Serial receive disable because of processing
Ready signal (READY)	Slave	High-level signal to be output to SB0/SB1 before serial transfer start and after completion of serial transfer		[1] BSYE = 0 [2] Execution of instruction data write SIO0 (transfer start instruction)	—	Serial receive enable

Table 14-4. Various Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Serial clock ($\overline{\text{SCK0}}$)	Master	Synchronous clock to output address/command/data, $\overline{\text{ACK}}$ signal, synchronous $\overline{\text{BUSY}}$ signal, etc. Address/command/data are transferred with the first eight synchronous clocks.		When CSIE0 = 1, execution of instruction for data write to SIO0 (serial transfer start instruction) Note 2	CSIIF0 set (rising edge of 9th clock of $\overline{\text{SCK0}}$) Note 1	Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data to be transferred in synchronization with $\overline{\text{SCK0}}$ after output of REL and CMD signals				Address value of slave device on the serial bus
Command (C7 to C0)	Master	8-bit data to be transferred in synchronization with $\overline{\text{SCK0}}$ after output of only CMD signal without REL signal output				Instructions and messages to the slave device
Data (D7 to D0)	Master/ slave	8-bit data to be transferred in synchronization with $\overline{\text{SCK0}}$ without output of REL and CMD signals				Numeric values to be processed with slave or master device

Notes 1. With WUP = 0, CSIIF0 is set always at the rising edge of the 9th clock of $\overline{\text{SCK0}}$.
With WUP = 1, CSIIF0 is set only when the received address matches the slave address register (SVA) content.

2. In $\overline{\text{BUSY}}$ state, transfer starts after the READY state is entered.

(5) Pin configuration

The serial clock pin $\overline{\text{SCK0}}$ and serial data bus pin SB0 (SB1) have the following configurations.

(a) $\overline{\text{SCK0}}$: Serial clock input/output pin

[1] Master : CMOS and push-pull output

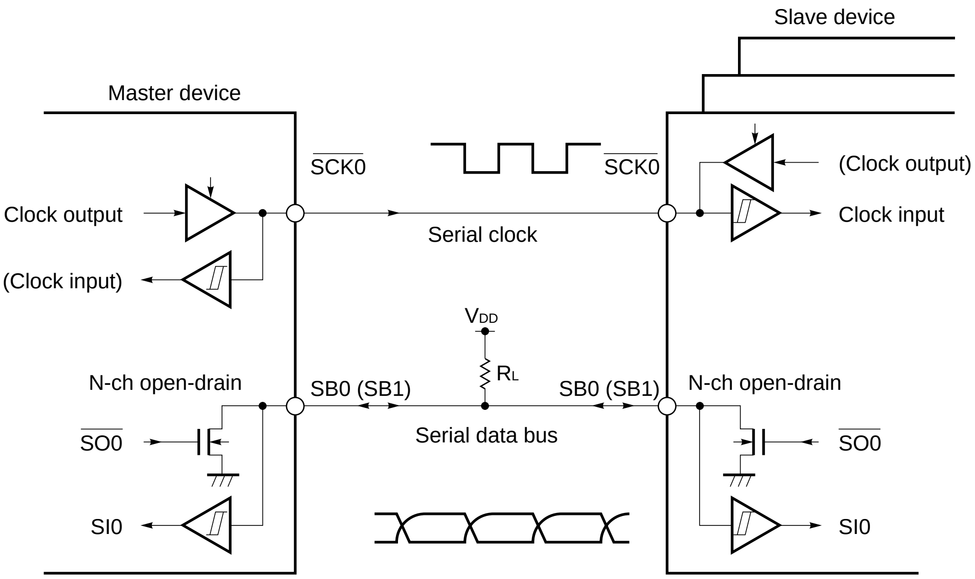
[2] Slave : Schmitt input

(b) SB0 (SB1) : Serial data input/output dual-function pin

Both master and slave devices have an N-ch open-drain output and a Schmitt input.

Because the serial data bus line has an N-ch open-drain output, an external pull-up resistor is necessary.

Figure 14-18. Pin Configuration



Caution Because the N-ch open-drain output must be made to go into a high-impedance state during data reception, write FFH to the serial I/O shift register 0 (SIO0) in advance. The N-ch open-drain output can always go into a high-impedance state during transfer. However, when wake-up function specify bit (WUP) = 1, the N-ch open-drain output always goes into a high-impedance state. Thus, it is not necessary to write FFH to SIO0.

★ **(6) Address match detection method**

In the SBI mode, the master transmits a slave address to select a specific slave device.

Coincidence of the addresses can be automatically detected by hardware. CSIIF0 is set only when the slave address transmitted by the master coincides with the address set to SVA when the wake-up function specify bit (WUP) = 1.

If bit 5 (SIC) of the interrupt timing specify register (SINT) is set to 1, the wake-up function cannot be used even if WUP is set to 1 (an interrupt request signal is generated when bus release is detected). To use the wake-up function, clear SIC to 0.

Cautions 1. Slave selection/non-selection is detected by matching of the slave address received after bus release (RELD = 1).

For this match detection, match interrupt request (CSIIFO) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

2. When detecting selection/non-selection without the use of interrupt request with WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address match detection method.

(7) Error detection

In the SBI mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, the serial I/O shift register 0 (SIO0). Thus, transmit errors can be detected in the following two ways.

(a) Comparison of SIO0 data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Use of the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If “1”, normal transmission is judged to have been carried out. If “0”, a transmit error is judged to have occurred.

(8) Communication operation

In the SBI mode, the master device selects normally one slave device as the communication target from among two or more devices by outputting an “address” to the serial bus.

After the communication target device has been determined, commands and data are transmitted/ received and serial communication is realized between the master and slave devices.

Figures 14-19 to 14-22 show data communication timing charts.

Shift operation of the shift register is carried out at the falling edge of the serial clock ($\overline{\text{SCK0}}$). Transmit data is latched into the SO0 latch and is output with MSB set as the first bit from the SB0/P25 or SB1/P26 pin.

Receive data input to the SB0 (or SB1) pin at the rising edge of $\overline{\text{SCK0}}$ is latched into the shift register.

Figure 14-19. Address Transmission from Master Device to Slave Device (WUP = 1)

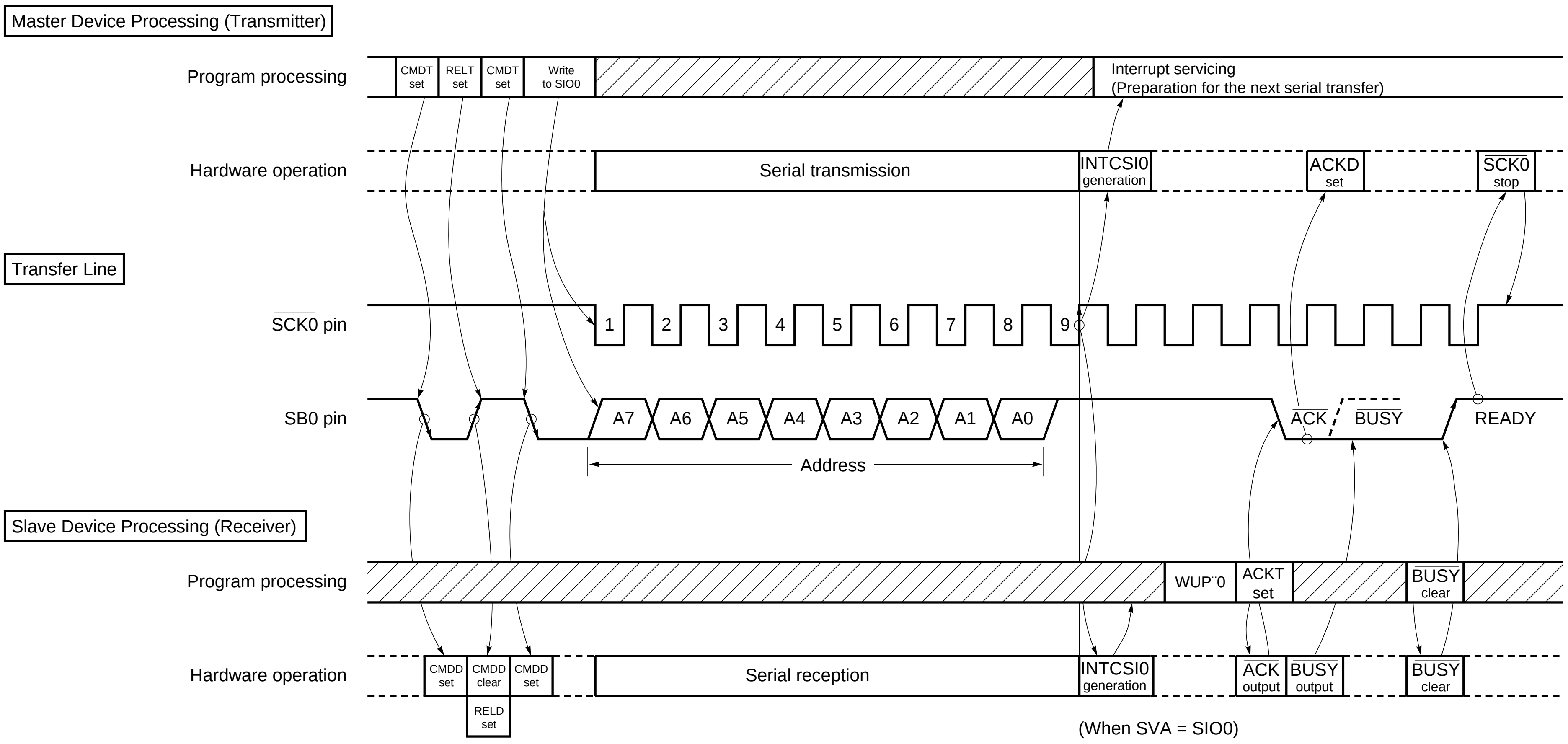


Figure 14-20. Command Transmission from Master Device to Slave Device

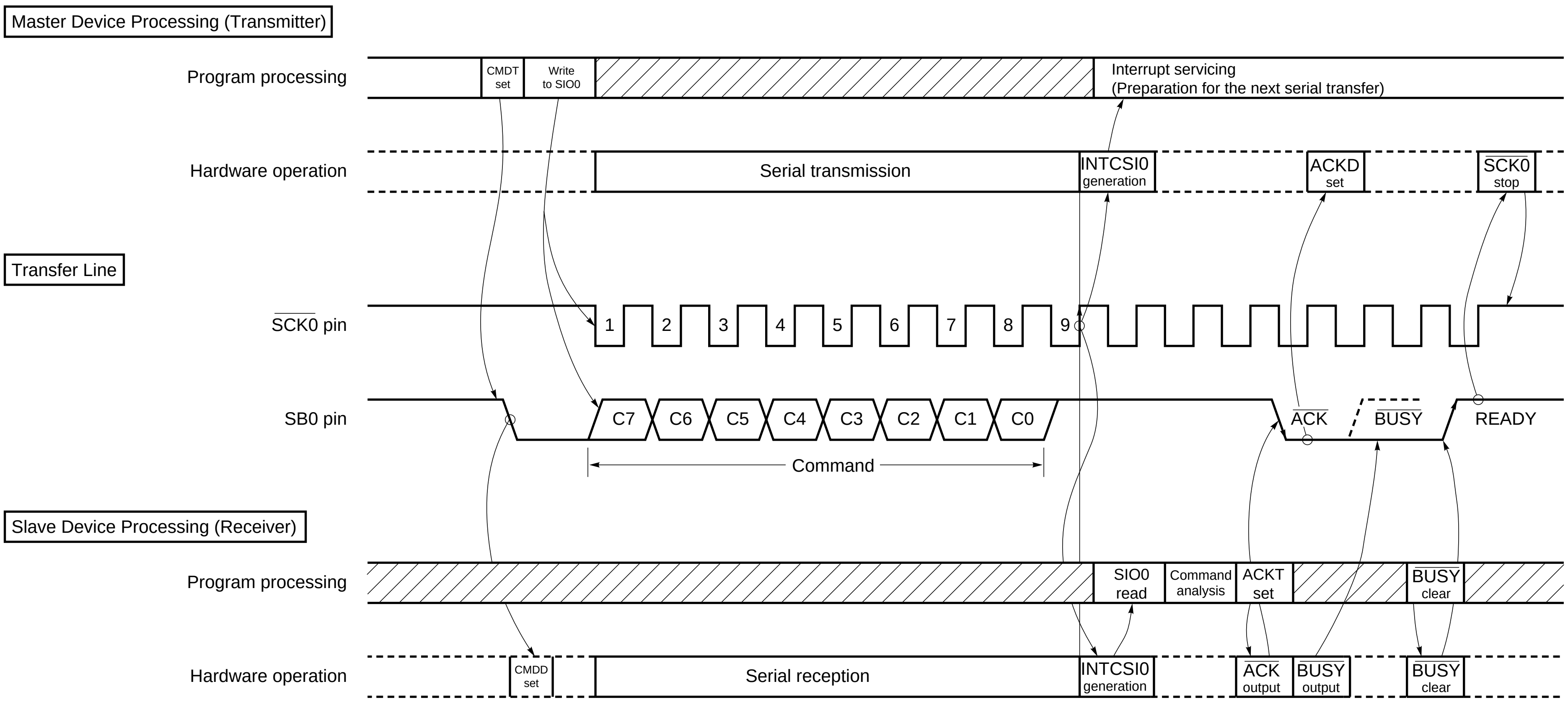


Figure 14-21. Data Transmission from Master Device to Slave Device

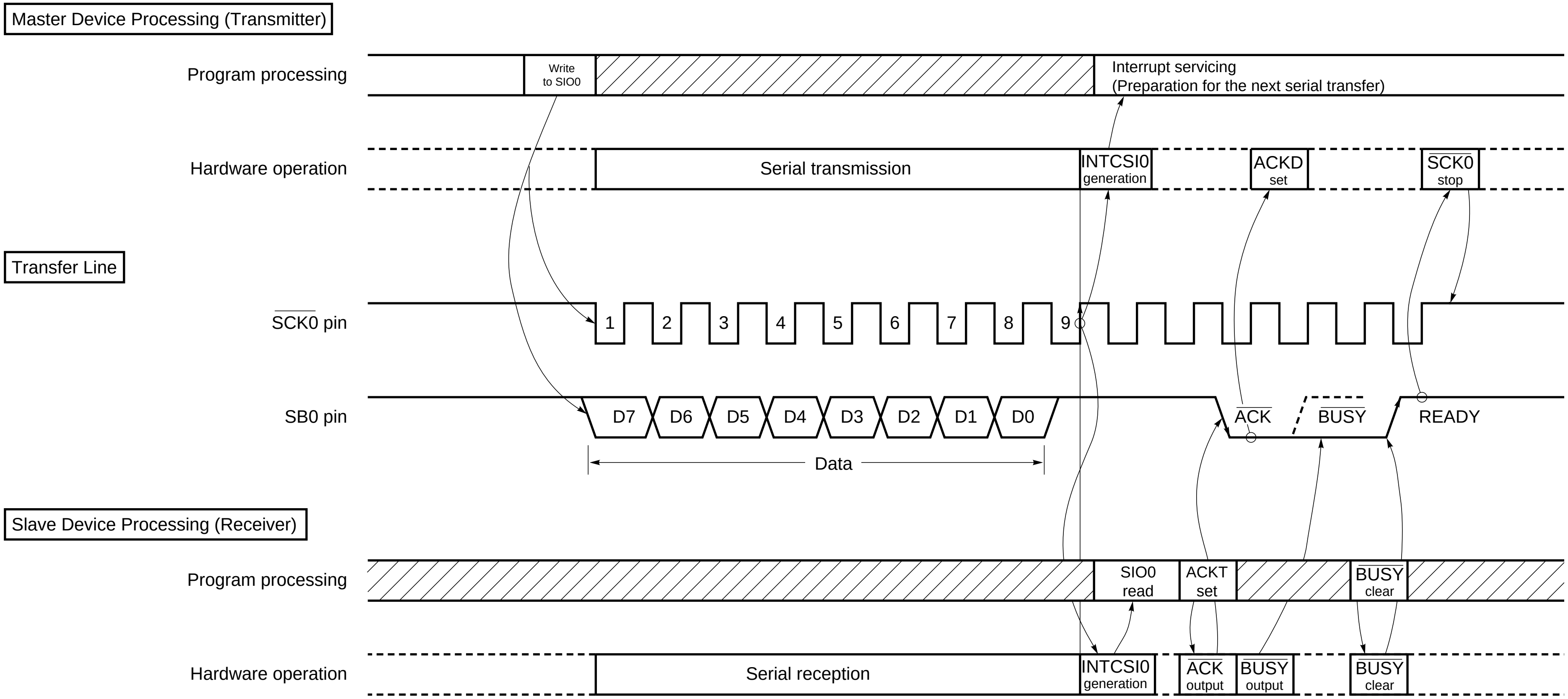
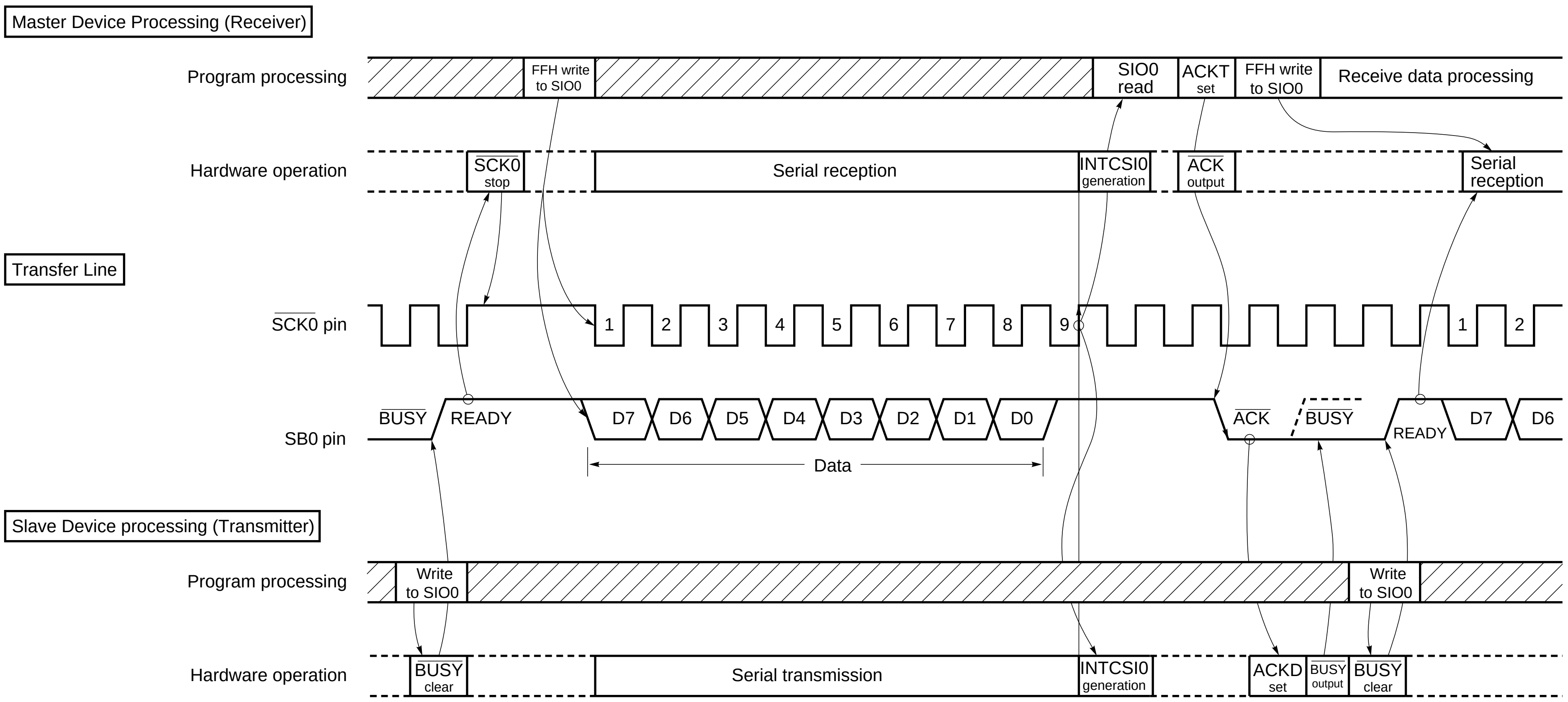


Figure 14-22. Data Transmission from Slave Device to Master Device



(9) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0)= 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must be made to go into a high-impedance state during data reception, write FFH to SIO0 in advance. However, when make-up function specify bit (WUP) = 1, the N-ch open-drain output always goes into a high-impedance state. Thus, it is not necessary to write FFH to SIO0.

3. If data is written to SIO0 when the slave is busy, the data is not lost.
When the busy state is cleared and SB0 (or SB1) input is set to the high level (READY) state, transfer starts.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

★ For pin (SB0 or SB1) which is to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after $\overline{\text{RESET}}$ input.

[1] Set the P25 and P26 output latches to 1.

[2] Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.

[3] Reset the P25 and P26 output latches from 1 to 0.

(10) Judging busy status of slave

When device is in the master mode, follow the procedure below to judge whether slave device is in the busy state or not.

[1] Detect acknowledge signal ($\overline{\text{ACK}}$) or interrupt request signal generation.

[2] Set the port mode register PM25 (or PM26) of the SB0/P25 (or SB1/P26) pin into the input mode.

[3] Read out the pin state (when the pin level is high, the READY state is set).

After the detection of the READY state, set the port mode register to 0 and return to the output mode.

(11) SBI mode precautions

(a) Slave selection/non-selection is detected by match detection of the slave address received after bus release (RELD = 1).
For this match detection, match interrupt (CSIIF0) of the address to be generated with WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when WUP = 1.

(b) When detecting selection/non-selection without the use of interrupt with WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address match detection method.

(c) If WUP is set to 1 during $\overline{\text{BUSY}}$ signal output, $\overline{\text{BUSY}}$ is not cleared. In SBI, the $\overline{\text{BUSY}}$ signal continues to be output after $\overline{\text{BUSY}}$ clear instruction generation to the falling edge of the next serial clock ($\overline{\text{SCK0}}$). Before setting WUP to 1, be sure to clear $\overline{\text{BUSY}}$ and then check that the SB0 (SB1) has become high-level.

(d) For pins which are to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after $\overline{\text{RESET}}$ input.
[1] Set the P25 and P26 output latches to 1.
[2] Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.
[3] Reset the P25 and P26 output latches from 1 to 0.

14.4.4 2-wire serial I/O mode operation

The 2-wire serial I/O mode can cope with any communication format by program.

Communication is basically carried out with two lines of serial clock ($\overline{\text{SCK0}}$) and serial data input/output (SB0 or SB1).

(1) Register setting

The 2-wire serial I/O mode is set with the serial operating mode register 0 (CSIM0), the serial bus interface control register (SBIC) and the interrupt timing specify register (SINT).

(a) Serial operating mode register 0 (CSIM0)

CSIM0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM0 to 00H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After 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	CSIM 01	CSIM 00	Serial Interface Channel 0 Clock Selection
	0	×	Input clock to $\overline{\text{SCK0}}$ pin from off-chip
	1	0	8-bit timer register 2 (TM2) output
	1	1	Clock specified with bits 0 to 3 of timer clock select register 3 (TCL3)

R/W	CSIM 04	CSIM 03	CSIM 02	PM25	P25	PM26	P26	PM27	P27	Operating Mode	Start Bit	SI0/P25 Pin Function	SO0/P26 Pin Function	$\overline{\text{SCK0}}$/P27 Pin Function
	0	×	3-wire serial I/O mode (Refer to **14.4.2 3-wire serial I/O mode operation**)											
	1	0	SBI mode (Refer to **14.4.3 SBI mode operation**)											
	1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode	MSB	P25 (CMOS input/output)	SB1/P26 (N-ch open-drain input/output)	$\overline{\text{SCK0}}$ (N-ch open-drain input/output)
			1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain input/output)	P26 (CMOS input/ output)	

R/W	WUP	Wake-up Function Control Note 3
	0	Interrupt request signal generation with each serial transfer in any mode
	1	Interrupt request signal generation when the address received after bus release (when CMDD = RELD = 1) matches the slave address register in SBI mode

R	COI	Slave Address Comparison Result Flag Note 4
	0	Slave address register not equal to serial I/O shift register 0 data
	1	Slave address register equal to serial I/O shift register 0 data

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enable

Notes **1.** Bit 6 (COI) is a Read-Only bit.

2. Can be used freely as port function.

★ **3.** Be sure to set WUP to 0 when the 2-wire serial I/O mode.

4. When CSIE0 = 0, COI becomes 0.

Remark ×: don’t care

PMxx: Port mode register

Pxx: Port output latch

(b) Serial bus interface control register (SBIC)

SBIC is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears SBIC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W

R/W	RELT	When RELT = 1, SO latch is set to 1. After SO latch setting, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R/W	CMDT	When CMDT = 1, SO latch is cleared to 0. After SO latch clearance, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

Bit 7 of the serial operating mode register 0 (CSIM0)

(c) Interrupt timing specify register (SINT)

SINT is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears SINT to 00H.

Symbol	7	⑥	⑤	④	3	2	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	0	0	0	0	FF63H	00H	R/W Note 1

R/W

SVAM	SVA Bit to be Used as Slave Address
0	Bits 0 to 7
1	Bits 1 to 7

R/W

SIC	INTCSI0 Interrupt Factor Selection
0	CSIIF0 is set upon termination of serial interface channel 0 transfer
1	CSIIF0 is set upon bus release detection or termination of serial interface channel 0 transfer

R

CLD	$\overline{\text{SCK0}}$ Pin Level Note 2
0	Low Level
1	High Level

Notes **1.** Bit 6 (CLD) is a Read-Only bit.
2. When CSIE0 = 0, CLD becomes 0.

Caution Be sure to set bits 0 to 3 to 0.

Remark SVA: Slave address register
CSIIF0: Interrupt request flag corresponding to INTCSI0
CSIE0: Bit 7 of the serial operating mode register 0 (CSIM0)

(2) Communication operation

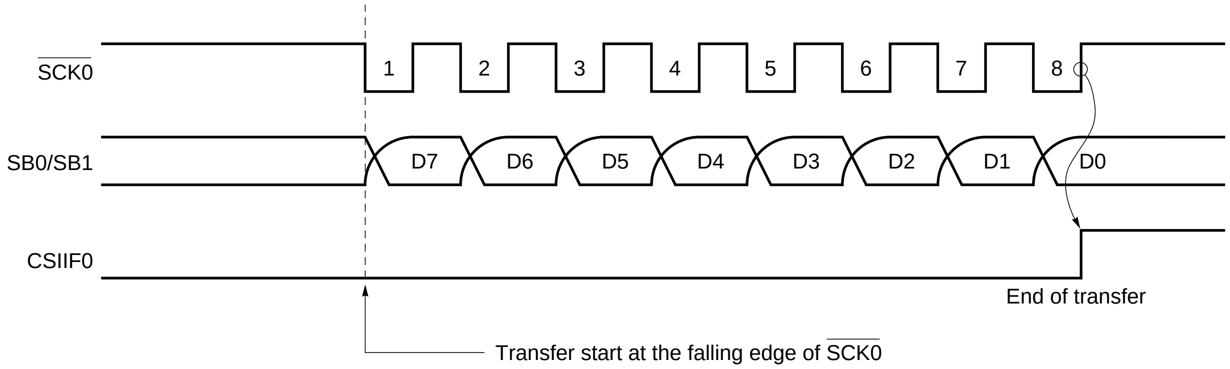
The 2-wire serial I/O mode is used for data transmission/reception in 8-bit units. Data transmission/reception is carried out bit-wise in synchronization with the serial clock.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out in synchronization with the falling edge of the serial clock ($\overline{\text{SCK0}}$).

The transmit data is held in the SO0 latch and is output from the SB0/P25 (or SB1/P26) pin with MSB set at start. The receive data input from the SB0 (or SB1) pin is latched into the shift register at the rising edge of $\overline{\text{SCK0}}$.

Upon termination of 8-bit transfer, the shift register operation stops automatically and the interrupt request flag (CSIIF0) is set.

Figure 14-23. 2-Wire Serial I/O Mode Timings



The SB0 (or SB1) pin specified for the serial data bus serves for N-ch open-drain input/output and thus it must be externally pulled up. Because it is necessary to go into a high-impedance state during data reception, write FFH to SIO0 in advance.

The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting bits 0 (RELT) and 1 (CMDT) of the serial bus interface control register (SBIC).

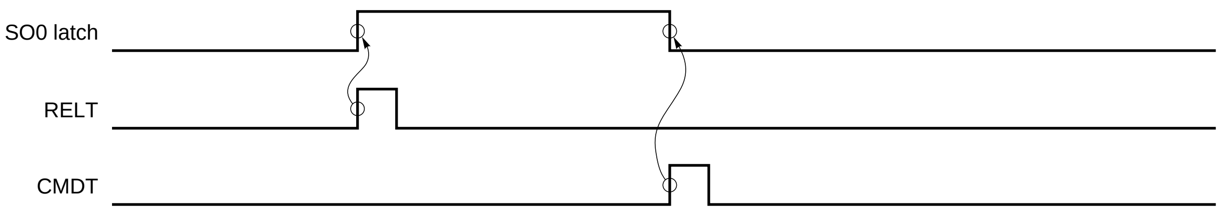
However, do not carry out this manipulation during serial transfer.

Control the $\overline{\text{SCK0}}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **14.4.5 $\overline{\text{SCK0}}$ pin output manipulation**).

(3) Various signals

Figure 14-24 shows RELT and CMDT operations.

Figure 14-24. RELT and CMDT Operations



(4) Transfer start

Serial transfer is started by setting transfer data to the serial I/O shift register 0 (SIO0) when the following two conditions are satisfied.

- Serial interface channel 0 operation control bit (CSIE0)= 1
- Internal serial clock is stopped or $\overline{\text{SCK0}}$ is at high level after 8-bit serial transfer.

Cautions 1. If CSIE0 is set to “1” after data write to SIO0, transfer does not start.

2. Because the N-ch open-drain output must made to go into a high-impedance state during data reception, write FFH to SIO0 in advance.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

(5) Error detection

In the 2-wire serial I/O mode, the serial bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, the serial I/O shift register 0 (SIO0). Thus, transmit error can be detected in the following two ways.

(a) Comparison of SIO0 data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Use of the slave address register (SVA)

Transmit data is set to both SIO0 and SVA and is transmitted. After termination of transmission, COI bit (match signal coming from the address comparator) of the serial operating mode register 0 (CSIM0) is tested. If “1”, normal transmission is judged to have been carried out. If “0”, a transmit error is judged to have occurred.

14.4.5 $\overline{\text{SCK0}}$ pin output manipulation

Because the $\overline{\text{SCK0}}$/P27 pin incorporates an output latch, static output is also possible by software in addition to normal serial clock output.

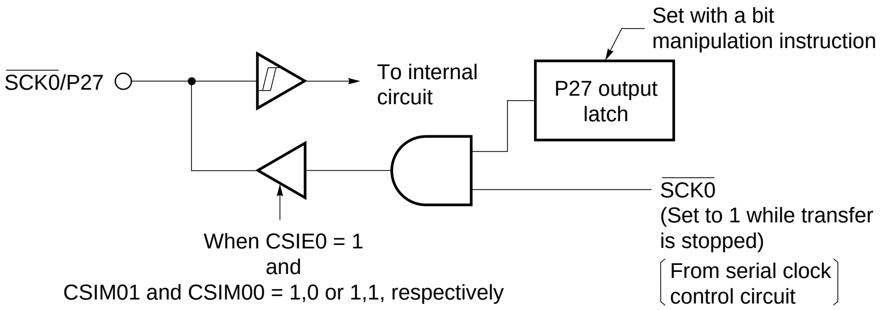
P27 output latch manipulation enables any value of $\overline{\text{SCK0}}$ to be set by software (SI0/SB0 and SO0/SB1 pin to be controlled with bits 0 (RELT) and 1 (CMDT) of the serial bus interface control register (SBIC).

$\overline{\text{SCK0}}$/P27 pin output manipulating procedure is described below.

[1] Set the serial operating mode register 0 (CSIM0) ($\overline{\text{SCK0}}$ pin enabled for serial operation in the output mode). $\overline{\text{SCK0}}$ = 1 with serial transfer suspended.

[2] Manipulate the P27 output latch with a bit manipulation instruction.

Figure 14-25. $\overline{\text{SCK0}}$/P27 Pin Configuration


$\overline{\text{SCK0}}$/P27
To internal circuit
Set with a bit manipulation instruction
P27 output latch
$\overline{\text{SCK0}}$
(Set to 1 while transfer is stopped)
From serial clock control circuit
When CSIE0 = 1
and
CSIM01 and CSIM00 = 1,0 or 1,1, respectively

CHAPTER 15 SERIAL INTERFACE CHANNEL 1

15.1 Serial Interface Channel 1 Functions

Serial interface channel 1 employs the following three modes.

Table 15-1. Difference of Serial Interface Channel 1 Mode

Operation Mode	Related Pin	Features	Usage
Operation stop mode	—	• Used when serial transfer is not carried out. • Power consumption can be reduced.	—
3-wire serial I/O mode (MSB-/LSB-first switchable)	$\overline{\text{SCK1}}$ (serial clock), SO1 (serial output) and SI1 (serial input)	• Input and output lines are independent and they can transfer/receive at the same time, so the data transfer processing time is short. • The start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB.	These modes are valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K and 17K Series.
3-wire serial I/O mode with automatic transmit/ receive function (MSB-/LSB-first switchable)	$\overline{\text{SCK1}}$ (serial clock), SO1 (serial output), SI1 (serial input)	• Mode with same function as 3-wire serial I/O mode above plus automatic transmit/receive function • Can transmit/receive data with a maximum of 64 bytes. Therefore, this function enables the hardware to transmit/receive data to/from the OSD (On Screen Display) device and device with on-chip display controller/driver independently of the CPU thus the software load can be reduced.	

15.2 Serial Interface Channel 1 Configuration

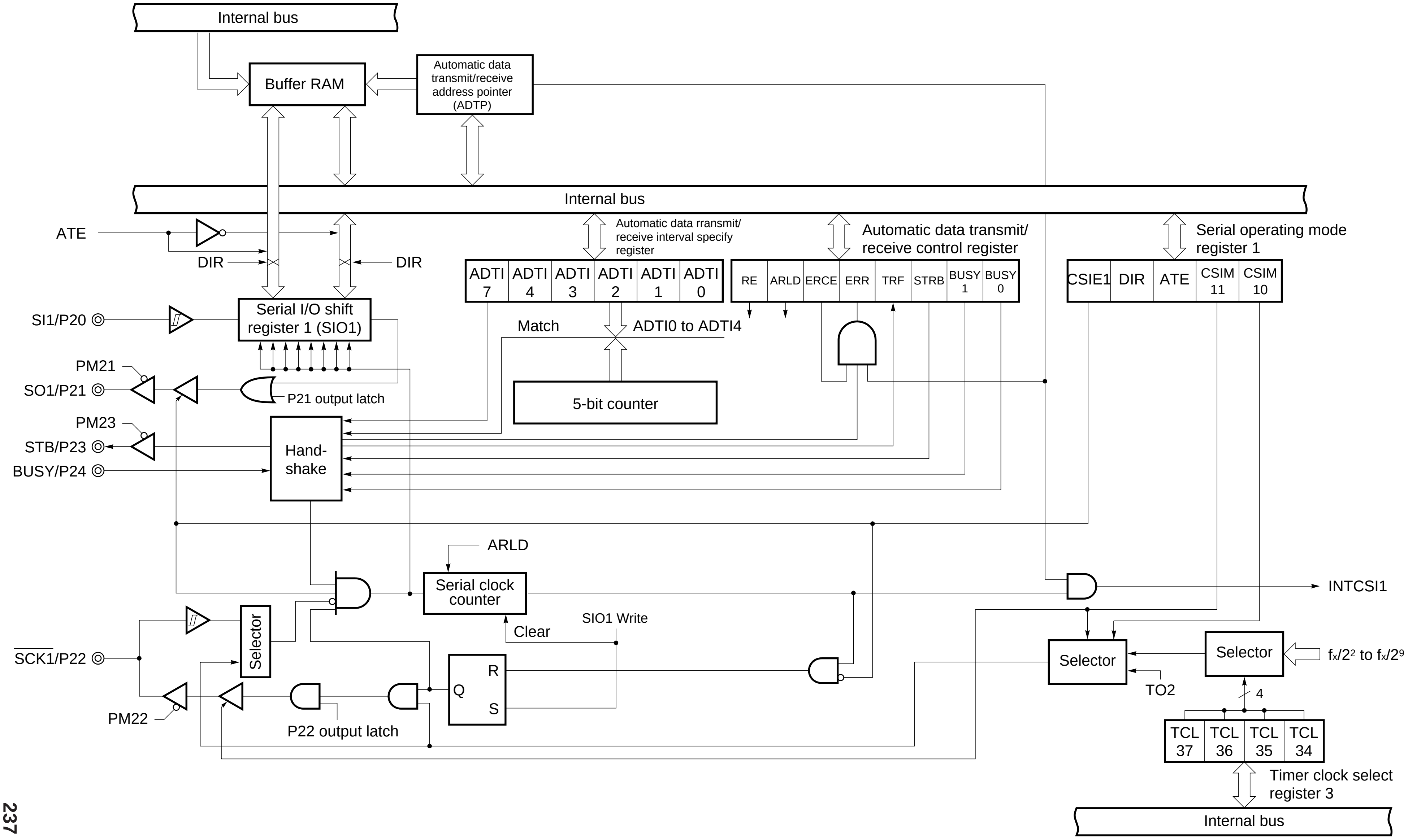
Serial interface channel 1 consists of the following hardware.

Table 15-2. Serial Interface Channel 1 Configuration

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 operating mode register 1 (CSIM1) Automatic data transmit/receive control register (ADTC) Automatic data transmit/receive interval specify register (ADTI) Port mode register 2 (PM2) Note

Note Refer to **Figure 4-5 P20, P21, P23 to P26 Block Diagram** and **Figure 4-6 P22 and P27 Block Diagram**.

Figure 15-1. Serial Interface Channel 1 Block Diagram



(1) Serial I/O shift register 1 (SIO1)

This is an 8-bit register to carry out parallel/serial conversion and to carry out serial transmission/reception (shift operation) in synchronization with the serial clock.

SIO1 is set with an 8-bit memory manipulation instruction.

When value in bit 7 (CSIE1) of serial operating mode register 1 (CSIM1) is 1, writing data to SIO1 starts serial operation.

In transmission, data written to SIO1 is output to the serial output (SO1). In reception, data is read from the serial input (SI1) to SIO1.

$\overline{\text{RESET}}$ input makes SIO1 undefined.

Caution Do not write data to SIO1 while the automatic transmit/receive function is activated.

(2) Automatic data transmit/receive address pointer (ADTP)

This register stores value of (the number of transmit data bytes –1) while the automatic transmit/receive function is activated.

It is decremented automatically with data transmission/reception.

ADTP is set with an 8-bit memory manipulation instruction. The high-order 3 bits must be set to 0.

$\overline{\text{RESET}}$ input clears ADTP to 00H.

Caution Do not write data to ADTP while the automatic transmit/receive function is activated.

(3) Serial clock counter

This counter counts the serial clocks to be output and input during transmission/reception and to check whether 8-bit data has been transmitted/received.

15.3 Serial Interface Channel 1 Control Registers

The following four types of registers are used to control serial interface channel 1.

- Timer clock select register 3 (TCL3)
- Serial operating mode register 1 (CSIM1)
- Automatic data transmit/receive control register (ADTC)
- Automatic data transmit/receive interval specify register (ADTI)

(1) Timer clock select register 3 (TCL3)

This register sets the serial clock of serial interface channel 1.

TCL3 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCL3 to 88H.

Remark Besides setting the serial clock of serial interface channel 1, TCL3 sets the serial clock of serial interface channel 0.

Figure 15-2. Timer Clock Select Register 3 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
TCL3	TCL37	TCL36	TCL35	TCL34	TCL33	TCL32	TCL31	TCL30	FF43H	88H	R/W

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection
0	1	1	0	$f_X/2^2$ (1.25 MHz)
0	1	1	1	$f_X/2^3$ (625 kHz)
1	0	0	0	$f_X/2^4$ (313 kHz)
1	0	0	1	$f_X/2^5$ (156 kHz)
1	0	1	0	$f_X/2^6$ (78.1 kHz)
1	0	1	1	$f_X/2^7$ (39.1 kHz)
1	1	0	0	$f_X/2^8$ (19.5 kHz)
1	1	0	1	$f_X/2^9$ (9.8 kHz)
Other than above				Setting prohibited

Caution If a value which is not the same as the previous value must be written to TCL3, stop the serial transfer before writing the value.

Remarks 1. f_X: Main system clock frequency

2. Figures in parentheses apply to operation with f_X = 5.0 MHz

(2) Serial operating mode register 1 (CSIM1)

This register sets serial interface channel 1 serial clock, operating mode, operation enable/stop and automatic transmit/receive operation enable/stop.

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Figure 15-3. Serial Operating Mode Register 1 Format

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial Interface Channel 1 Clock Selection
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin Note 1
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start 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 stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1 Note 3 (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. If the external clock input has been selected with CSIM11 set to 0, set bits 1 (BUSY1) and 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. Can be used as P20 when only transmitter is used. (Set bit 7 (RE) of ADTC to 0.)

Remark ×: don't care

PMxx: Port mode register

Pxx: Port output latch

(3) Automatic data transmit/receive control register (ADTC)

This register sets automatic receive enable/disable, the operating mode, strobe output enable/disable, busy input enable/disable, error check enable/disable and displays automatic transmit/receive execution and error detection.

ADTC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADTC to 00H.

Figure 15-4. Automatic Data Transmit/Receive Control Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Busy Input Control
0	×	Not using busy input
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB	Strobe Output Control
0	Strobe output disable
1	Strobe output enable

R

TRF	Status of Automatic Transmit/Receive Function Note 2
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error Detection of Automatic Transmit/Receive Function
0	No error (This bit is set to 0 when data is written to SIO1)
1	Error occurred

R/W

ERCE	Error Check Control of Automatic Transmit/ Receive Function
0	Error check disable
1	Error check enable (only when BUSY1 = 1)

R/W

ARLD	Operating Mode Selection of Automatic Transmit/ Receive Function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive Control of Automatic Transmit/Receive Function
0	Receive disable
1	Receive enable

Notes 1. Bits 3 and 4 (TRF and ERR) are Read-Only bits.

2. The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1 (interrupt request flag).

Caution When an external clock input is selected with bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) set to 0, set STRB and BUSY1 of ADTC to 0, 0.

Remark ×: don't care

(4) Automatic data transmit/receive interval specify register (ADTI)

This register sets the automatic data transmit/receive function data transfer interval.

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

$\overline{\text{RESET}}$ input clears ADTI to 00H.

Figure 15-5. Automatic Data Transmit/Receive Interval Specify Register Format (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADTI Note 1
1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_X = 5.0-MHz Operation)	
					Minimum Note 2	Maximum Note 2
0	0	0	0	0	36.8 μs + 0.5/f_{SCK}	40.0 μs + 1.5/f_{SCK}
0	0	0	0	1	62.4 μs + 0.5/f_{SCK}	65.6 μs + 1.5/f_{SCK}
0	0	0	1	0	88.0 μs + 0.5/f_{SCK}	91.2 μs + 1.5/f_{SCK}
0	0	0	1	1	113.6 μs + 0.5/f_{SCK}	116.8 μs + 1.5/f_{SCK}
0	0	1	0	0	139.2 μs + 0.5/f_{SCK}	142.4 μs + 1.5/f_{SCK}
0	0	1	0	1	164.8 μs + 0.5/f_{SCK}	168.0 μs + 1.5/f_{SCK}
0	0	1	1	0	190.4 μs + 0.5/f_{SCK}	193.6 μs + 1.5/f_{SCK}
0	0	1	1	1	216.0 μs + 0.5/f_{SCK}	219.2 μs + 1.5/f_{SCK}
0	1	0	0	0	241.6 μs + 0.5/f_{SCK}	244.8 μs + 1.5/f_{SCK}
0	1	0	0	1	267.2 μs + 0.5/f_{SCK}	270.4 μs + 1.5/f_{SCK}
0	1	0	1	0	292.8 μs + 0.5/f_{SCK}	296.0 μs + 1.5/f_{SCK}
0	1	0	1	1	318.4 μs + 0.5/f_{SCK}	321.6 μs + 1.5/f_{SCK}
0	1	1	0	0	344.0 μs + 0.5/f_{SCK}	347.2 μs + 1.5/f_{SCK}
0	1	1	0	1	369.6 μs + 0.5/f_{SCK}	372.8 μs + 1.5/f_{SCK}
0	1	1	1	0	395.2 μs + 0.5/f_{SCK}	398.4 μs + 1.5/f_{SCK}
0	1	1	1	1	420.8 μs + 0.5/f_{SCK}	424.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/fSCK, the minimum interval time is 2/fSCK.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic data transmit/receive function.

2. Zero must be set in bits 5 and 6.

★ **3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 15.4.3 (4) (a) Busy control option) is invalid.**

Remarks 1. f_X : Main system clock frequency

2. f_{SCK} : Serial clock frequency

Figure 15-5. Automatic Data Transmit/Receive Interval Specify Register Format (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_X = 5.0-MHz Operation)	
					Minimum [Note]	Maximum [Note]
1	0	0	0	0	446.4 μs + 0.5/f_{SCK}	449.6 μs + 1.5/f_{SCK}
1	0	0	0	1	472.0 μs + 0.5/f_{SCK}	475.2 μs + 1.5/f_{SCK}
1	0	0	1	0	497.6 μs + 0.5/f_{SCK}	500.8 μs + 1.5/f_{SCK}
1	0	0	1	1	523.2 μs + 0.5/f_{SCK}	526.4 μs + 1.5/f_{SCK}
1	0	1	0	0	548.8 μs + 0.5/f_{SCK}	552.0 μs + 1.5/f_{SCK}
1	0	1	0	1	574.4 μs + 0.5/f_{SCK}	577.6 μs + 1.5/f_{SCK}
1	0	1	1	0	600.0 μs + 0.5/f_{SCK}	603.2 μs + 1.5/f_{SCK}
1	0	1	1	1	625.6 μs + 0.5/f_{SCK}	628.8 μs + 1.5/f_{SCK}
1	1	0	0	0	651.2 μs + 0.5/f_{SCK}	654.4 μs + 1.5/f_{SCK}
1	1	0	0	1	676.8 μs + 0.5/f_{SCK}	680.0 μs + 1.5/f_{SCK}
1	1	0	1	0	702.4 μs + 0.5/f_{SCK}	705.6 μs + 1.5/f_{SCK}
1	1	0	1	1	728.0 μs + 0.5/f_{SCK}	731.2 μs + 1.5/f_{SCK}
1	1	1	0	0	753.6 μs + 0.5/f_{SCK}	756.8 μs + 1.5/f_{SCK}
1	1	1	0	1	779.2 μs + 0.5/f_{SCK}	782.4 μs + 1.5/f_{SCK}
1	1	1	1	0	804.8 μs + 0.5/f_{SCK}	808.0 μs + 1.5/f_{SCK}
1	1	1	1	1	830.4 μs + 0.5/f_{SCK}	833.6 μs + 1.5/f_{SCK}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic data transmit/receive function.

2. Zero must be set in bits 5 and 6.

★ **3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 15.4.3 (4) (a) Busy control option) is invalid.**

Remarks 1. f_X : Main system clock frequency

1. f_{SCK} : Serial clock frequency

15.4 Serial Interface Channel 1 Operations

The following three operating modes are available to the serial interface channel 1.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

15.4.1 Operation stop mode

Serial transfer is not carried out in the operation stop mode. Thus, power consumption can be reduced. The serial I/O shift register 1 (SIO1) does not carry out shift operation either, and thus it can be used as an ordinary 8-bit register.

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 input/output ports.

(1) Register setting

The operation stop mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIE1	CSIM 11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter 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 stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note 2 1	Note 2 ×	0	0	1	×	Operation enable	Count operation	SI1 Note 2 (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. Can be used freely as port function.

2. Can be used as P20 (CMOS input/output) when only transmitter is used. (Set bit 7 (RE) of the automatic data transmit/receive control register (ADTC) to 0.)

Remark ×: don’t care

PMxx: Port mode register

Pxx: Port output latch

15.4.2 3-wire serial I/O mode operation

The 3-wire serial I/O mode is valid for connection of peripheral I/O units and display controllers which incorporate a conventional synchronous serial interface as is the case with the 75X/XL, 78K and 17K Series.

Communication is carried out with three lines of serial clock ($\overline{\text{SCK1}}$), serial output (SO1) and serial input (SI1).

(1) Register setting

The 3-wire serial I/O mode is set with the serial operating mode register 1 (CSIM1).

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	CSIM10	FF68H	00H	R/W

CSIM11	CSIM10	Serial Interface Channel 1 Clock Selection
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin Note 1
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start Bit	SI1 Pin Function	SO1 Pin function
0	MSB	SI1/P20 (Input)	SO1 (CMOS output)
1	LSB		

CSIE1	CSIM11	PM20	P20	PM21	P21	PM22	P22	Shift Register 1 Operation	Serial Clock Counter Operation Control	SI1/P20 Pin Function	SO1/P21 Pin Function	$\overline{\text{SCK1}}$/P22 Pin Function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1 Note 3 (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. If the external clock input has been selected with CSIM11 set to 0, set bits 1 (BUSY1) and 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. Can be used as P20 when only transmitter is used. (Set bit 7 (RE) of ADTC to 0.)

Remark ×: don’t care

PMxx: Port mode register

Pxx: Port output latch

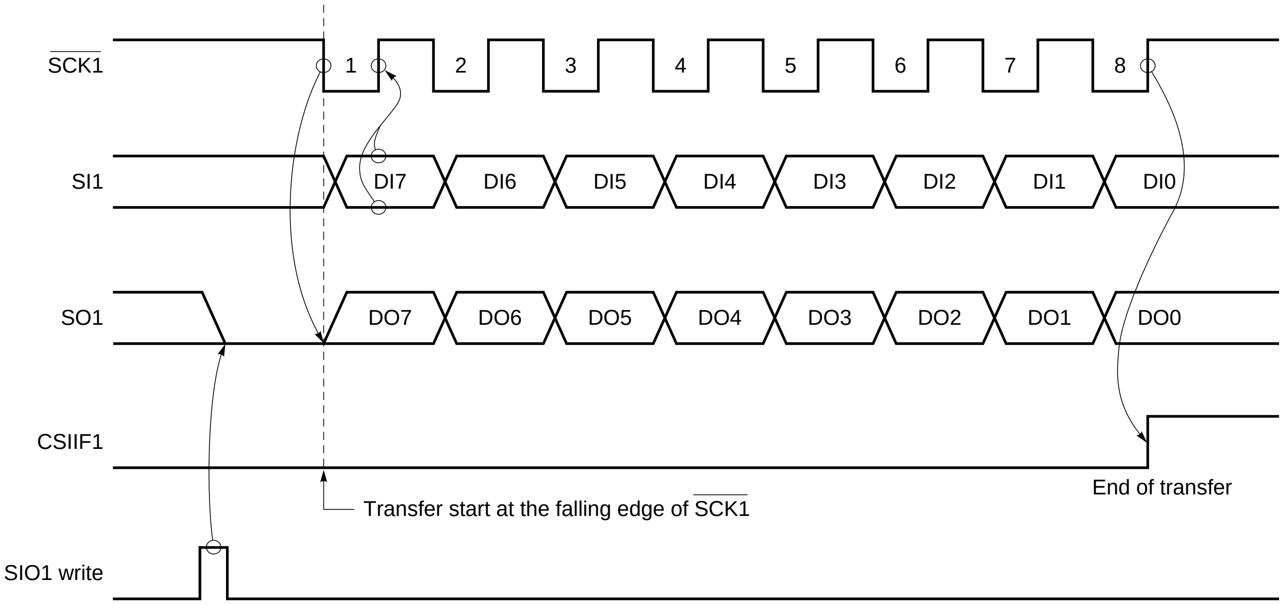
(2) Communication operation

The 3-wire serial I/O mode is used for data transmission/reception in 8-bit units. Bit-wise data transmission/reception is carried out in synchronization with the serial clock.

Shift operation of the serial I/O shift register 1 (SIO1) is carried out at the falling edge of the serial clock ($\overline{\text{SCK1}}$). The transmit data is held in the SO1 latch and is output from the SO1 pin. The receive data input to the SI1 pin is latched into SIO1 at the rising edge of $\overline{\text{SCK1}}$.

Upon termination of 8-bit transfer, the SIO1 operation stops automatically and the interrupt request flag (CSIIF1) is set.

Figure 15-6. 3-Wire Serial I/O Mode Timings



Caution SO1 pin becomes low level by SIO1 write.

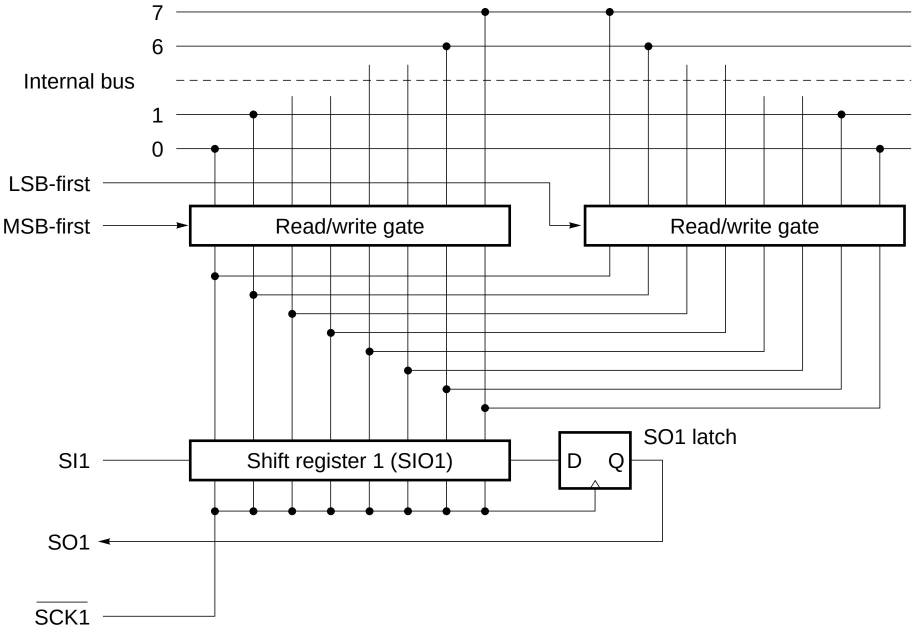
★ **(3) MSB/LSB switching as the start bit**

The 3-wire serial I/O mode enables to select transfer to start from MSB or LSB.

Figure 15-7 shows the configuration of the serial I/O shift register 1 (SIO1) and internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

MSB/LSB switching as the start bit can be specified with bit 6 (DIR) of the serial operating mode register 1 (CSIM1).

Figure 15-7. Circuit of Switching in Transfer Bit Order



Start bit switching is realized by switching the bit order for data write to SIO1. The SIO1 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

★ **(4) Transfer start**

Serial transfer is started by setting transfer data to the serial I/O shift register 1 (SIO1) when the following two conditions are satisfied.

- Serial interface channel 1 operation control bit (CSIE1) = 1
- Internal serial clock is stopped or $\overline{\text{SCK1}}$ is a high level after 8-bit serial transfer.

Caution If CSIE1 is set to "1" after data write to SIO1, transfer does not start.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF1) is set.

15.4.3 3-wire serial I/O mode operation with automatic transmit/receive function

This 3-wire serial I/O mode is used for transmission/reception of a maximum of 64-byte data without the use of software. Once transfer is started, the data prestored in the RAM can be transmitted by the set number of bytes, and data can be received and stored in the RAM by the set number of bytes.

Handshake signals (STB and BUSY) are supported by hardware to transmit/receive data continuously. OSD (On Screen Display) LSI and peripheral LSI including LCD controller/driver can be connected without difficulty.

(1) Register setting

The 3-wire serial I/O mode with automatic transmit/receive function is set with the serial operating mode register 1 (CSIM1), the automatic data transmit/receive control register (ADTC) and the automatic data transmit/receive interval specify register (ADTI).

(a) Serial operating mode register 1 (CSIM1)

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM1 to 00H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM 11	CSIM 10	FF68H	00H	R/W

CSIM 11	CSIM 10	Serial Interface Channel 1 Clock Selection
0	×	Clock externally input to $\overline{\text{SCK1}}$ pin Note 1
1	0	8-bit timer register 2 (TM2) output
1	1	Clock specified with bits 4 to 7 of timer clock select register 3 (TCL3)

ATE	Serial Interface Channel 1 Operating Mode Selection
0	3-wire serial I/O mode
1	3-wire serial I/O mode with automatic transmit/receive function

DIR	Start 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 stop	Clear	P20 (CMOS input/output)	P21 (CMOS input/output)	P22 (CMOS input/output)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1 Note 3 (Input)	SO1 (CMOS output)	$\overline{\text{SCK1}}$ (Input)
	1					0	1					$\overline{\text{SCK1}}$ (CMOS output)

Notes 1. If the external clock input has been selected with CSIM11 set to 0, set bits 1 (BUSY1) and 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. Can be used as P20 when only transmitter is used. (Set bit 7 (RE) of ADTC to 0.)

Remark ×: don’t care
PMxx: Port mode register
Pxx: Port output latch

(b) Automatic data transmit/receive control register (ADTC)

ADTC is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADTC to 00H.

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
ADTC	RE	ARLD	ERCE	ERR	TRF	STRB	BUSY1	BUSY0	FF69H	00H	R/W Note 1

R/W

BUSY1	BUSY0	Busy Input Control
0	x	Not using busy input
1	0	Busy input enable (active high)
1	1	Busy input enable (active low)

R/W

STRB	Strobe Output Control
0	Strobe output disable
1	Strobe output enable

R

TRF	Status of Automatic Transmit/Receive Function Note 2
0	Detection of termination of automatic transmission/ reception (This bit is set to 0 upon suspension of automatic transmission/reception or when ARLD = 0.)
1	During automatic transmission/reception (This bit is set to 1 when data is written to SIO1.)

R

ERR	Error Detection of Automatic Transmit/Receive Function
0	No error (This bit is set to 0 when data is written to SIO1)
1	Error occurred

R/W

ERCE	Error Check Control of Automatic Transmit/ Receive Function
0	Error check disable
1	Error check enable (only when BUSY1 = 1)

R/W

ARLD	Operating Mode Selection of Automatic Transmit/ Receive Function
0	Single operating mode
1	Repetitive operating mode

R/W

RE	Receive Control of Automatic Transmit/Receive Function
0	Receive disable
1	Receive enable

Notes 1. Bits 3 and 4 (TRF and ERR) are Read-Only bit.

2. The termination of automatic transmission/reception should be judged by using TRF, not CSIIF1 (interrupt request flag).

Caution When an external clock input is selected with bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) set to 0, set STRB and BUSY1 of ADTC to 0, 0 (handshake control cannot be executed when the external clock is input).

Remark ×: don’t care

(c) Automatic data transmit/receive interval specify register (ADTI)

This register sets the data transfer interval of the automatic data transmit/receive function.

ADTI is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADTI to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI7	Data Transfer Interval Control
0	No control of interval by ADTI Note 1
1	Control of interval by ADTI (ADTI0 to ADTI4)

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_X = 5.0-MHz Operation)	
					Minimum Note 2	Maximum Note 2
0	0	0	0	0	36.8 μs + 0.5/f_{SCK}	40.0 μs + 1.5/f_{SCK}
0	0	0	0	1	62.4 μs + 0.5/f_{SCK}	65.6 μs + 1.5/f_{SCK}
0	0	0	1	0	88.0 μs + 0.5/f_{SCK}	91.2 μs + 1.5/f_{SCK}
0	0	0	1	1	113.6 μs + 0.5/f_{SCK}	116.8 μs + 1.5/f_{SCK}
0	0	1	0	0	139.2 μs + 0.5/f_{SCK}	142.4 μs + 1.5/f_{SCK}
0	0	1	0	1	164.8 μs + 0.5/f_{SCK}	168.0 μs + 1.5/f_{SCK}
0	0	1	1	0	190.4 μs + 0.5/f_{SCK}	193.6 μs + 1.5/f_{SCK}
0	0	1	1	1	216.0 μs + 0.5/f_{SCK}	219.2 μs + 1.5/f_{SCK}
0	1	0	0	0	241.6 μs + 0.5/f_{SCK}	244.8 μs + 1.5/f_{SCK}
0	1	0	0	1	267.2 μs + 0.5/f_{SCK}	270.4 μs + 1.5/f_{SCK}
0	1	0	1	0	292.8 μs + 0.5/f_{SCK}	296.0 μs + 1.5/f_{SCK}
0	1	0	1	1	318.4 μs + 0.5/f_{SCK}	321.6 μs + 1.5/f_{SCK}
0	1	1	0	0	344.0 μs + 0.5/f_{SCK}	347.2 μs + 1.5/f_{SCK}
0	1	1	0	1	369.6 μs + 0.5/f_{SCK}	372.8 μs + 1.5/f_{SCK}
0	1	1	1	0	395.2 μs + 0.5/f_{SCK}	398.4 μs + 1.5/f_{SCK}
0	1	1	1	1	420.8 μs + 0.5/f_{SCK}	424.0 μs + 1.5/f_{SCK}

Notes 1. The interval is dependent only on CPU processing.

2. The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than $2/f_{SCK}$, the minimum interval time is $2/f_{SCK}$.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic data transmit/receive function.

2. Zero must be set in bits 5 and 6.

★ **3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 15.4.3 (4) (a) Busy control option) is invalid.**

Remarks 1. f_X : Main system clock frequency

2. f_{SCK} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_X = 5.0-MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	446.4 μs + 0.5/f_{SCK}	449.6 μs + 1.5/f_{SCK}
1	0	0	0	1	472.0 μs + 0.5/f_{SCK}	475.2 μs + 1.5/f_{SCK}
1	0	0	1	0	497.6 μs + 0.5/f_{SCK}	500.8 μs + 1.5/f_{SCK}
1	0	0	1	1	523.2 μs + 0.5/f_{SCK}	526.4 μs + 1.5/f_{SCK}
1	0	1	0	0	548.8 μs + 0.5/f_{SCK}	552.0 μs + 1.5/f_{SCK}
1	0	1	0	1	574.4 μs + 0.5/f_{SCK}	577.6 μs + 1.5/f_{SCK}
1	0	1	1	0	600.0 μs + 0.5/f_{SCK}	603.2 μs + 1.5/f_{SCK}
1	0	1	1	1	625.6 μs + 0.5/f_{SCK}	628.8 μs + 1.5/f_{SCK}
1	1	0	0	0	651.2 μs + 0.5/f_{SCK}	654.4 μs + 1.5/f_{SCK}
1	1	0	0	1	676.8 μs + 0.5/f_{SCK}	680.0 μs + 1.5/f_{SCK}
1	1	0	1	0	702.4 μs + 0.5/f_{SCK}	705.6 μs + 1.5/f_{SCK}
1	1	0	1	1	728.0 μs + 0.5/f_{SCK}	731.2 μs + 1.5/f_{SCK}
1	1	1	0	0	753.6 μs + 0.5/f_{SCK}	756.8 μs + 1.5/f_{SCK}
1	1	1	0	1	779.2 μs + 0.5/f_{SCK}	782.4 μs + 1.5/f_{SCK}
1	1	1	1	0	804.8 μs + 0.5/f_{SCK}	808.0 μs + 1.5/f_{SCK}
1	1	1	1	1	830.4 μs + 0.5/f_{SCK}	833.6 μs + 1.5/f_{SCK}

Note The data transfer interval includes an error. The data transfer minimum and maximum intervals are found from the following expressions (n: Value set in ADTI0 to ADTI4). However, if a minimum which is calculated by the following expressions is smaller than 2/f_{SCK}, the minimum interval time is 2/f_{SCK}.

$$\text{Minimum} = (n + 1) \times \frac{2^7}{f_X} + \frac{56}{f_X} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n + 1) \times \frac{2^7}{f_X} + \frac{72}{f_X} + \frac{1.5}{f_{SCK}}$$

Cautions 1. ADTI should not be written to during operation of the automatic data transmit/receive function.

2. Zero must be set in bits 5 and 6.

★ **3. To control the data transfer interval by means of automatic transmission/reception with ADTI, busy control (refer to 15.4.3 (4) (a) Busy control option) is invalid.**

Remarks 1. f_X : Main system clock frequency

2. f_{SCK} : Serial clock frequency

(2) Automatic transmit/receive data setting

(a) Transmit data setting

[1] Write transmit data from the least significant address FAC0H of buffer RAM (up to FAFFH at maximum).
The transmit data should be in the order from high-order address to low-order address.

[2] Set to the automatic data transmit/receive address pointer (ADTP) the value obtained by subtracting 1 from the number of transmit data bytes.

(b) Automatic transmit/receive mode setting

[1] Set bits 7 (CSIE1) and 5 (ATE) of the serial operating mode register 1 (CSIM1) to 1.

[2] Set bit 7 (RE) of the automatic data transmit/receive control register (ADTC) to 1.

[3] Set a data transmit/receive interval in the automatic data transmit /receive interval specify register (ADTI).

[4] Write any value to the serial I/O shift register 1 (SIO1) (transfer start trigger).

Caution Writing any value to SIO1 orders the start of automatic transmit/receive operation and the written value has no meaning.

The following operations are automatically carried out when (a) and (b) are carried out.

- After the buffer RAM data specified with ADTP is transferred to SIO1, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address specified with ADTP.
- ADTP is decremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTP decremental output becomes 00H and address FAC0H data is output (end of automatic transmission/reception).
- When automatic transmission/reception is terminated, bit 3 (TRF) of ADTC is cleared to 0.

(3) Communication operation

(a) Basic transmission/reception mode

This transmission/reception mode is the same as the 3-wire serial I/O mode in which specified number of data are transmitted/received in 8-bit units.

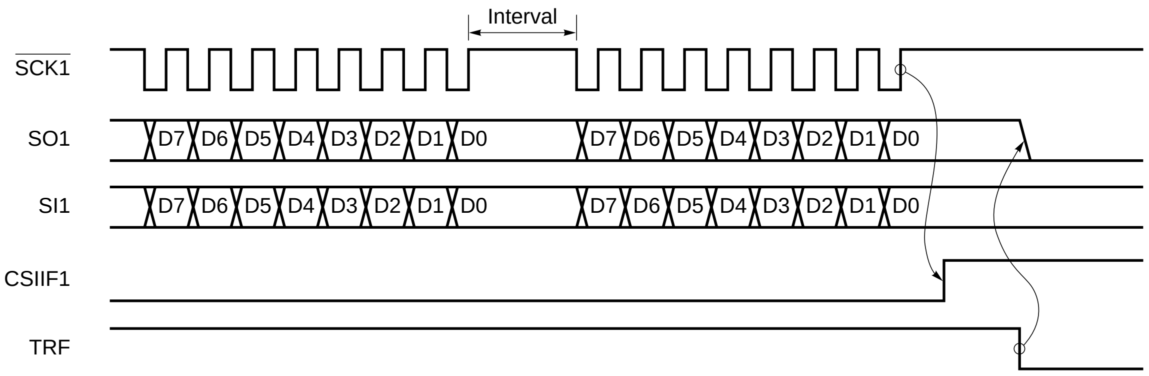
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. To judge completion, however, use bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) instead of the CSIIF1.

If busy control and strobe control are not executed, the P23/STB and P24/BUSY pins can be used as normal input/output ports.

Figure 15-8 shows the basic transmission/reception mode operation timings, Figure 15-9 shows the operation flowchart. The operation of the buffer RAM in 6-byte transmission/reception mode is shown in Figure 15-10.

Figure 15-8. Basic Transmission/Reception Mode Operation Timings



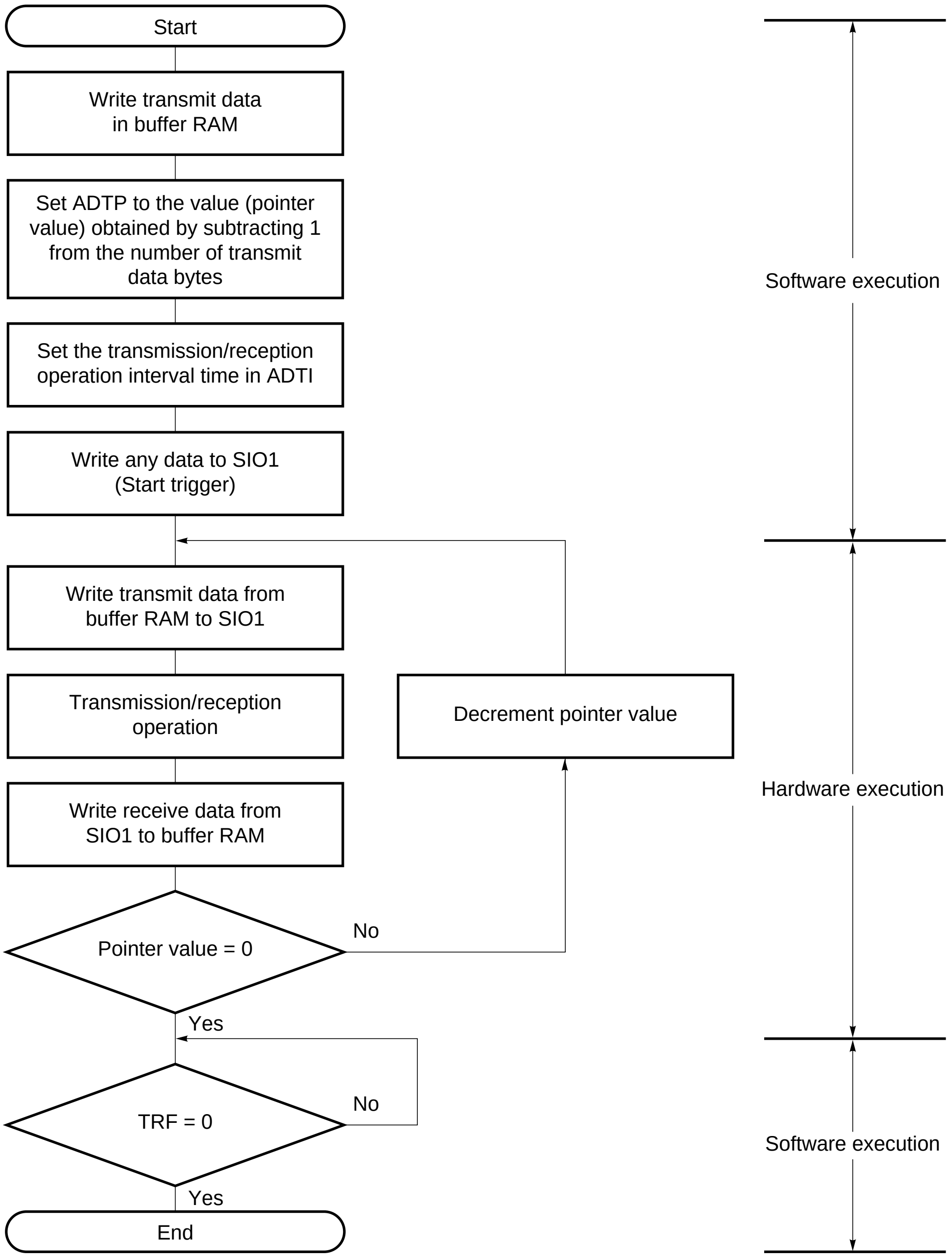
Cautions 1. Because, in the basic transmission/reception mode, the automatic transmit/receive function writes/reads data to/from the buffer RAM after 1-byte transmission/reception, an interval is inserted till the next transmission/reception.
As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic data transmit/receive interval).

2. When TRF is cleared, SO1 pin becomes low level.

Remarks 1. CSIIF1: Interrupt request flag

2. TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

Figure 15-9. Basic Transmission/Reception Mode Flowchart


Start
Write transmit data in buffer RAM
Set ADTP to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI
Write any data to SIO1 (Start trigger)
Write transmit data from buffer RAM to SIO1
Transmission/reception operation
Write receive data from SIO1 to buffer RAM
Pointer value = 0
No
Yes
Decrement pointer value
TRF = 0
No
Yes
End
Software execution
Hardware execution
Software execution

Remarks 1. ADTP : Automatic data transmit/receive address pointer

2. ADTI : Automatic data transmit/receive interval specify register

3. SIO1 : Serial I/O shift register 1

4. TRF : Bit 3 of the automatic data transmit/receive control register (ADTC)

In 6-byte transmission/reception (ARLD = 0, RE = 1) in basic transmit/receive mode, buffer RAM operates as follows.

(i) Before transmission/reception (refer to Figure 15-10 (a))

After arbitrary data has been written to the serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the receive data 1 (R1) is transferred from SIO1 to the buffer RAM, and the automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission/reception point (refer to Figure 15-10 (b))

Transmission/reception of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When transmission of the fourth byte is completed, the receive data 4 (R4) is transferred from SIO1 to the buffer RAM, and ADTP is decremented.

(iii) Completion of transmission/reception (refer to Figure 15-10 (c))

When transmission of the sixth byte is completed, the receive data 6 (R6) is transferred from SIO1 to the buffer RAM, and the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 15-10. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (1/2)

(a) Before transmission/reception

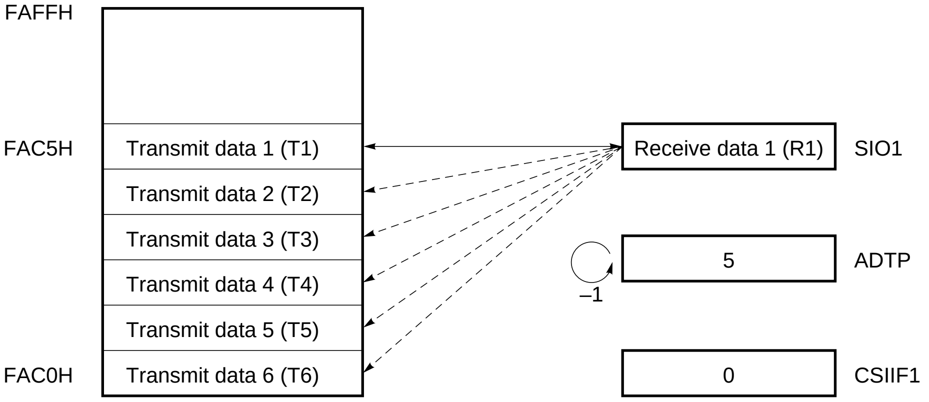
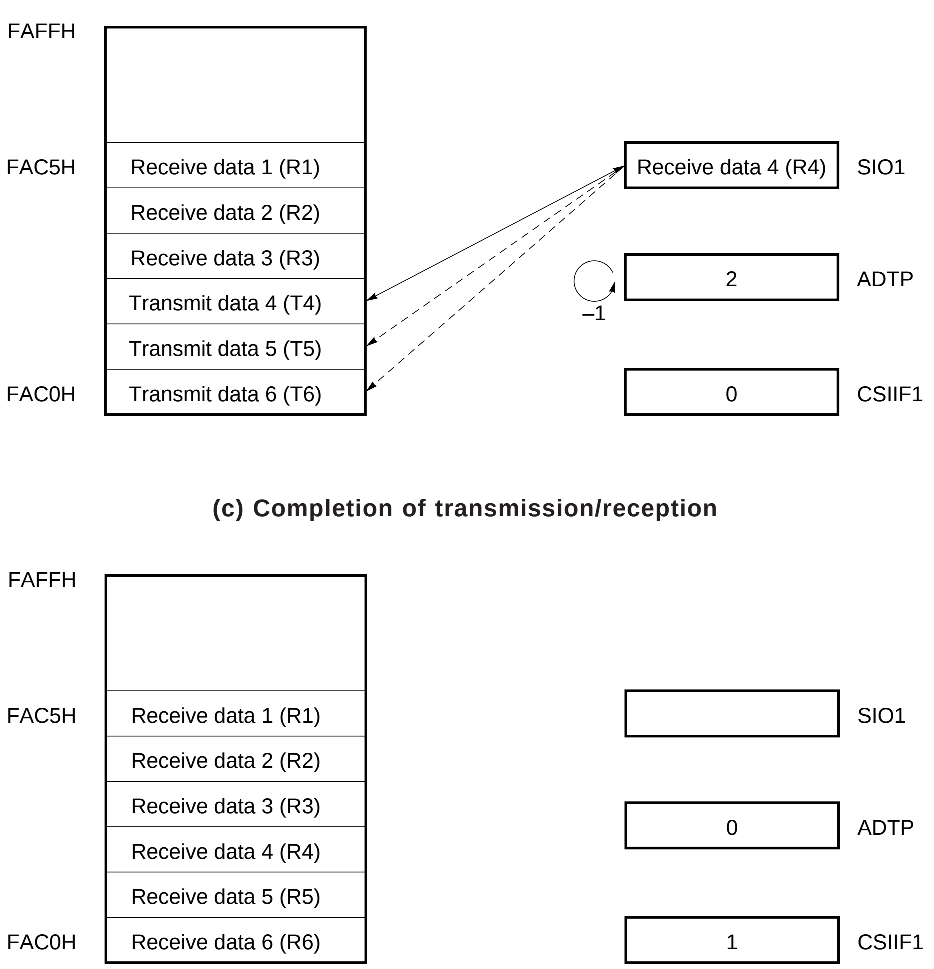


Figure 15-10. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (2/2)



(b) 4th byte transmission/reception point

(c) Completion of transmission/reception

(b) Basic transmission mode

In this mode, a specified number of 8-bit unit data are transmitted.

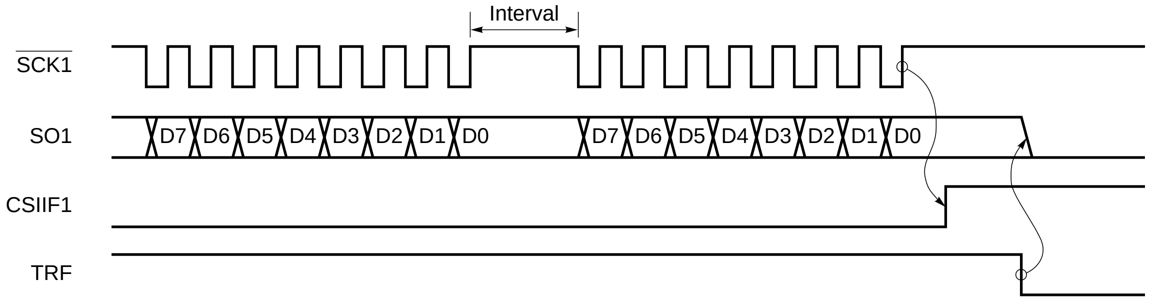
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. To judge completion, however, use bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) instead of CSIIF1.

If receive operation, busy control and strobe control are not executed, the P20/SI1, P23/STB and P24/BUSY pins can be used as normal input/output ports.

Figure 15-11 shows the basic transmission mode operation timings, Figure 15-12 shows the operation flowchart. The operation of the buffer RAM in 6-byte transmission mode is shown in Figure 15-13.

Figure 15-11. Basic Transmission Mode Operation Timings

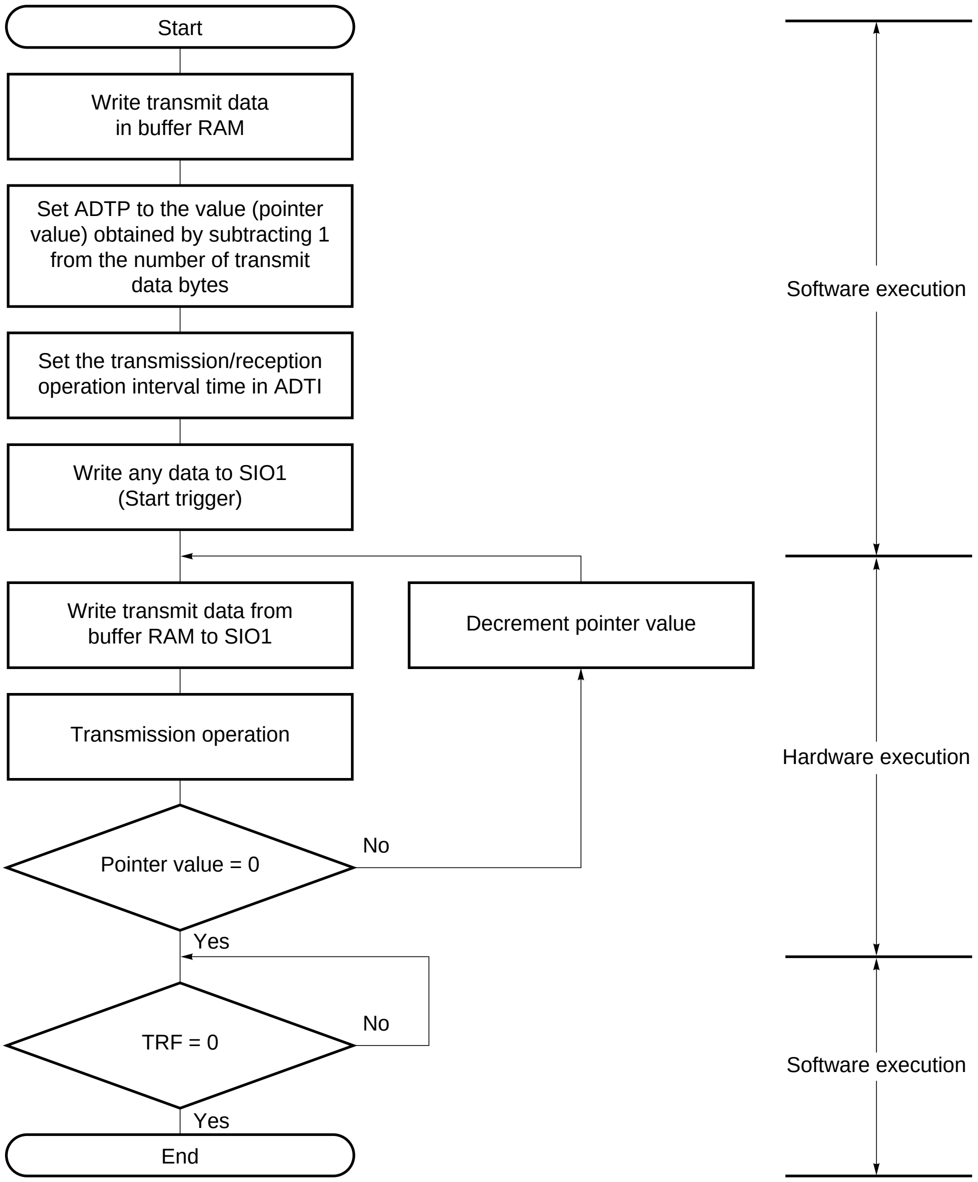


Cautions 1. Because, in the basic transmission mode, the automatic transmit/receive function reads data from the buffer RAM after 1-byte transmission, an interval is inserted till the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic data transmit/receive interval).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag
TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

Figure 15-12. Basic Transmission Mode Flowchart


Start
Write transmit data in buffer RAM
Set ADTP to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI
Write any data to SIO1 (Start trigger)
Write transmit data from buffer RAM to SIO1
Decrement pointer value
Transmission operation
Pointer value = 0
No
Yes
TRF = 0
No
Yes
End
Software execution
Hardware execution
Software execution

Remarks **1.** ADTP : Automatic data transmit/receive address pointer

2. ADTI : Automatic data transmit/receive interval specify register

3. SIO1 : Serial I/O shift register 1

4. TRF : Bit 3 of the automatic data transmit/receive control register (ADTC)

In 6-byte transmission (ARLD = 0, RE = 0) in basic transmit mode, buffer RAM operates as follows.

(i) Before transmission (refer to Figure 15-13 (a))

After arbitrary data has been written to the serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission point (refer to Figure 15-13 (b))

Transmission of the third byte is completed, and transmit data 4 (T4) is transferred from the buffer RAM to SIO1. When transmission of the fourth byte is completed, ADTP is decremented.

(iii) Completion of transmission (refer to Figure 15-13 (c))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 15-13. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (1/2)

(a) Before transmission

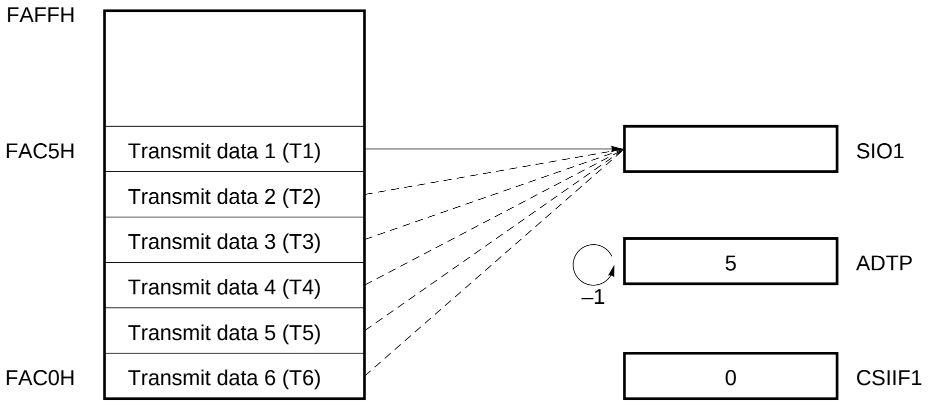
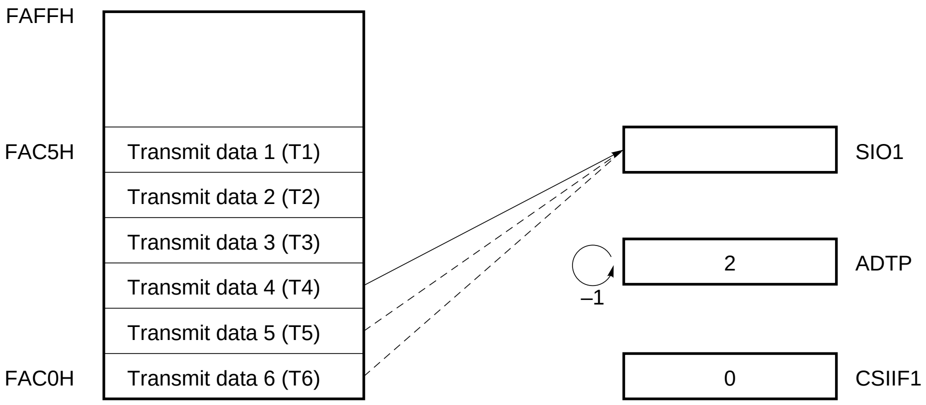
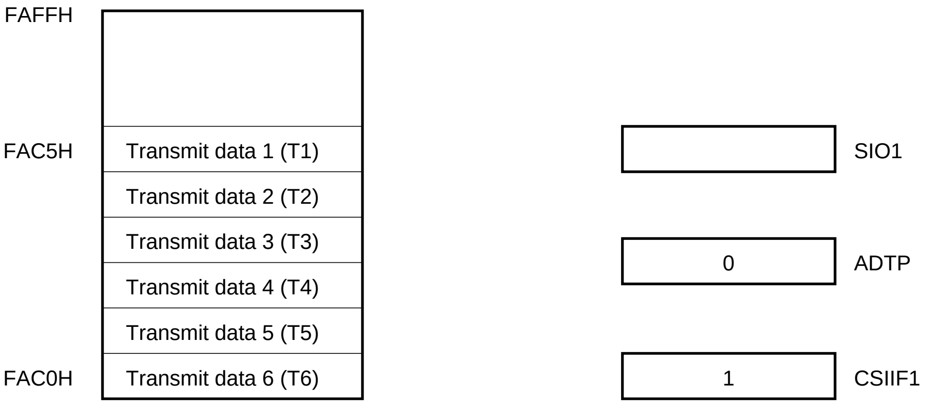


Figure 15-13. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (2/2)

(b) 4th byte transmission point



(c) Completion of transmission



(c) Repeat transmission mode

In this mode, data stored in the buffer RAM is transmitted repeatedly.

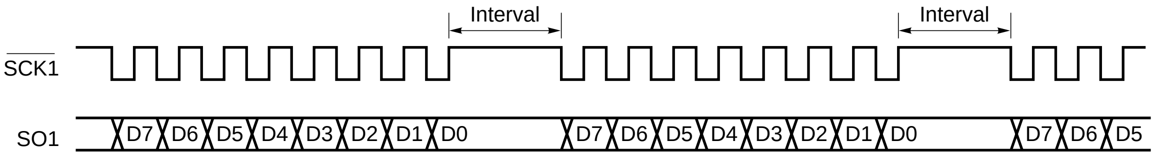
Serial transfer is started when any data is written to the serial I/O shift register 1 (SIO1) while bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) is set to 1.

Unlike the basic transmission mode, after the last byte (data in address FAC0H) has been transmitted, the interrupt request flag (CSIIF1) is not set, the value at the time when the transmit function was started is set in the automatic data transmit/receive address pointer (ADTP) again, and the buffer RAM contents are transmitted again.

When a reception operation, busy control and strobe control are not performed, the P20/SI1, P23/STB and P24/BUSY pins can be used as normal input/output ports.

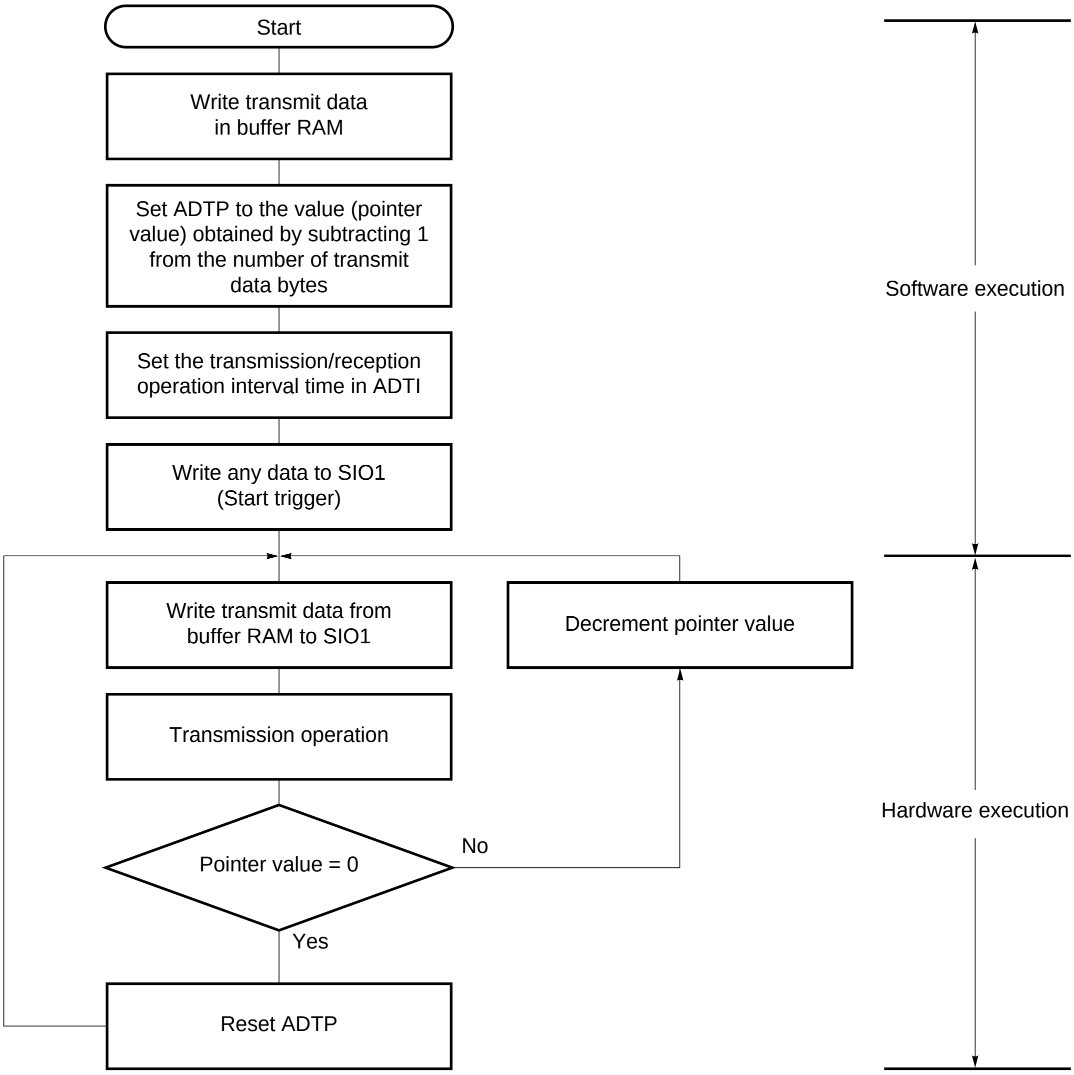
The repeat transmission mode operation timing is shown in Figure 15-14, and the operation flowchart in Figure 15-15. The operation of the buffer RAM in 6-byte repeat transmission mode is shown in Figure 15-16.

Figure 15-14. Repeat Transmission Mode Operation Timings



Caution Since, in the repeat transmission mode, a read is performed on the buffer RAM after the transmission of one byte, the interval is included in the period up to the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon the CPU operation and the value of the automatic data transmit/receive interval specify register (ADTI) (see (5) Automatic data transmit/receive interval).

Figure 15-15. Repeat Transmission Mode Flowchart


Start
Write transmit data in buffer RAM
Set ADTP to the value (pointer value) obtained by subtracting 1 from the number of transmit data bytes
Set the transmission/reception operation interval time in ADTI
Write any data to SIO1 (Start trigger)
Write transmit data from buffer RAM to SIO1
Decrement pointer value
Transmission operation
Pointer value = 0
No
Yes
Reset ADTP
Software execution
Hardware execution

Remarks **1.** ADTP : Automatic data transmit/receive address pointer

2. ADTI : Automatic data transmit/receive interval specify register

3. SIO1 : Serial I/O shift register 1

When data of 6 bytes are transmitted in repeat transmit mode (ARLD = 1, RE = 0), the buffer RAM operates as follows:

(i) Before transmission (refer to Figure 15-16 (a))

After arbitrary data has been written to the serial I/O shift register 1 (SIO1) (start trigger: this data is not transferred), transmit data 1 (T1) is transferred from the buffer RAM to SIO1. When transmission of the first byte is completed, the automatic data transmit/receive address pointer (ADTP) is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) Upon completion of transmission of 6th bytes (refer to Figure 15-16 (b))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is not set.

★ The first pointer value is set again to ADTP.

(iii) 7th byte transmission point (refer to Figure 15-16 (c))

Transmit data 1 (T1) is transferred from the buffer RAM to SIO1 again. When transmission of the first byte is completed, ADTP is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

Figure 15-16. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (1/2)

(a) Before transmission

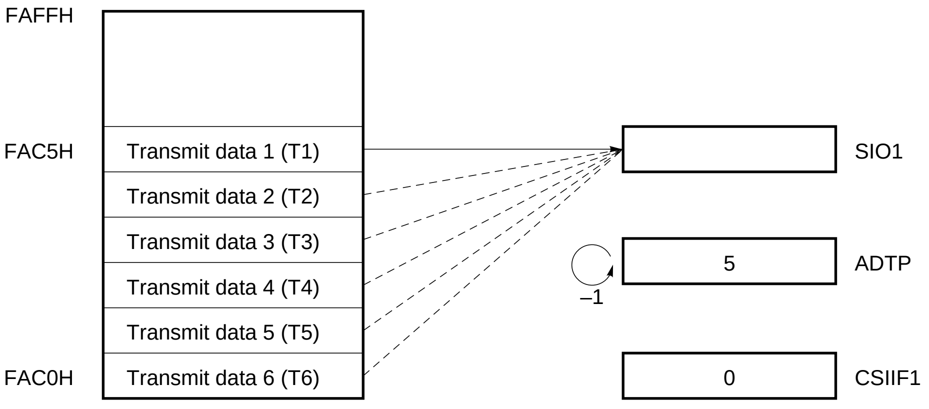
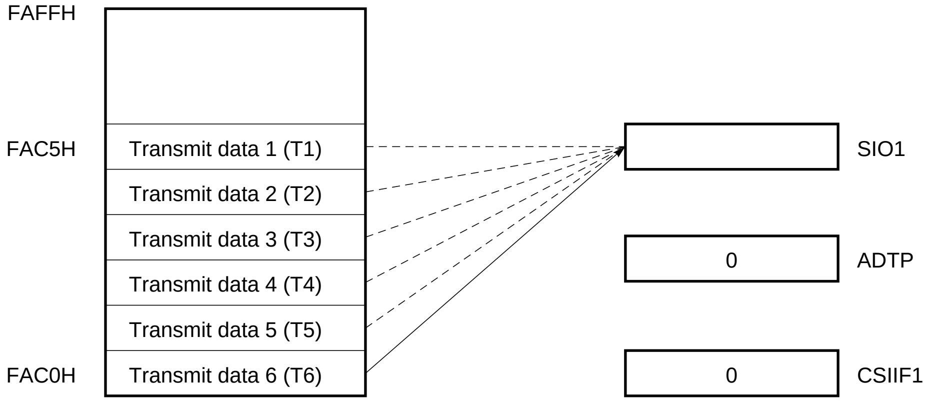
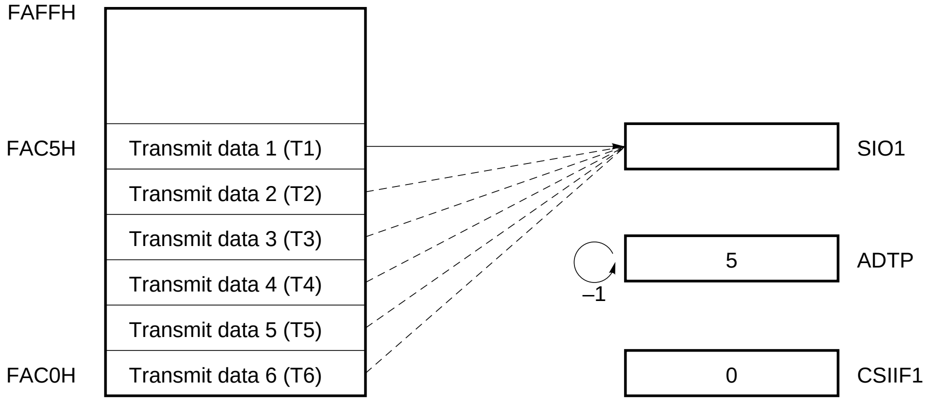


Figure 15-16. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (2/2)

(b) Upon completion of transmission of 6 bytes



(c) 7th byte transmission point



(d) Automatic transmission/reception suspending and restart

Automatic transmission/reception can be temporarily suspended by setting bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1) to 0.

If during 8-bit data transfer, the transmission/reception is not suspended if bit 7 (CSIE1) is set to 0. It is suspended upon completion of 8-bit data transfer.

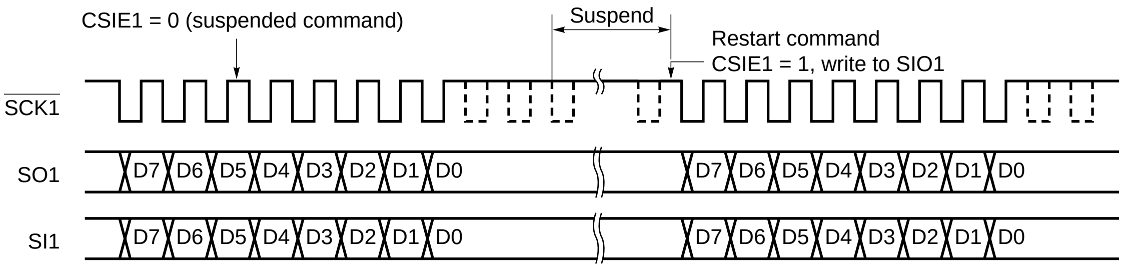
When suspended, bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) is set to 0 after transfer of the 8th bit, and all the port pins used with the serial interface pins for dual function (P20/SI1, P21/SO1, P22/$\overline{\text{SCK1}}$, P23/STB and P24/BUSY) are set to the port mode.

Automatic transmission/reception can be restarted and the remaining data can be transferred by setting CSIE1 to 1 and writing any data to the serial I/O shift register 1 (SIO1).

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, transfer is suspended and the HALT mode is set if during 8-bit data transfer. When the HALT mode is cleared, automatic transmission/reception is restarted at the suspended point.

★ **2. When suspending automatic transmission/reception, do not change the operating mode to 3-wire serial I/O mode while TRF = 1.**

Figure 15-17. Automatic Transmission/Reception Suspension and Restart



Remark CSIE1: Bit 7 of the serial operating mode register 1 (CSIM1)

★ **(4) Synchronization control**

Busy control and strobe control are to synchronize transmission/reception data between the master device and slave device.

By using these functions, a bit slippage in data being transmitted/received can be detected.

(a) Busy control option

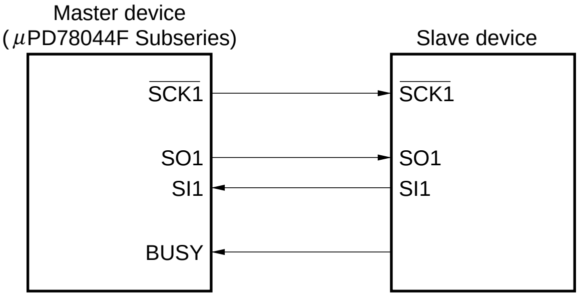
Busy control is to allow a slave device to output a busy signal to the master device, so that the master device puts serial transmission/reception into a wait state while the busy signal is active.

To use the busy control option, the following conditions must be satisfied:

- Set bit 5 (ATE) of the serial operating mode register 1 (CSIM1) to 1.
- Set bit 1 (BUSY1) of the automatic data transmit/receive control register (ADTC) to 1.

Figure 15-18 shows the system configuration of the master device and a slave device when the busy control option is used.

Figure 15-18. System Configuration with Busy Control Option



The master device inputs the busy signal output by the slave device to the BUSY/P24 pin. It samples the input busy signal in synchronization with the fall of the serial clock. Even if the busy signal becomes active while 8-bit data is transmitted or received, transmission/reception is not put into a wait state. If the busy signal is active at the rising edge of the serial clock two clocks after transmission or reception of 8-bit data has been completed, the busy signal becomes valid. After that, transmission or reception is put into a wait state while the busy signal is active.

The active level of the busy signal is specified by bit 0 (BUSY0) of ADTC, as follows:

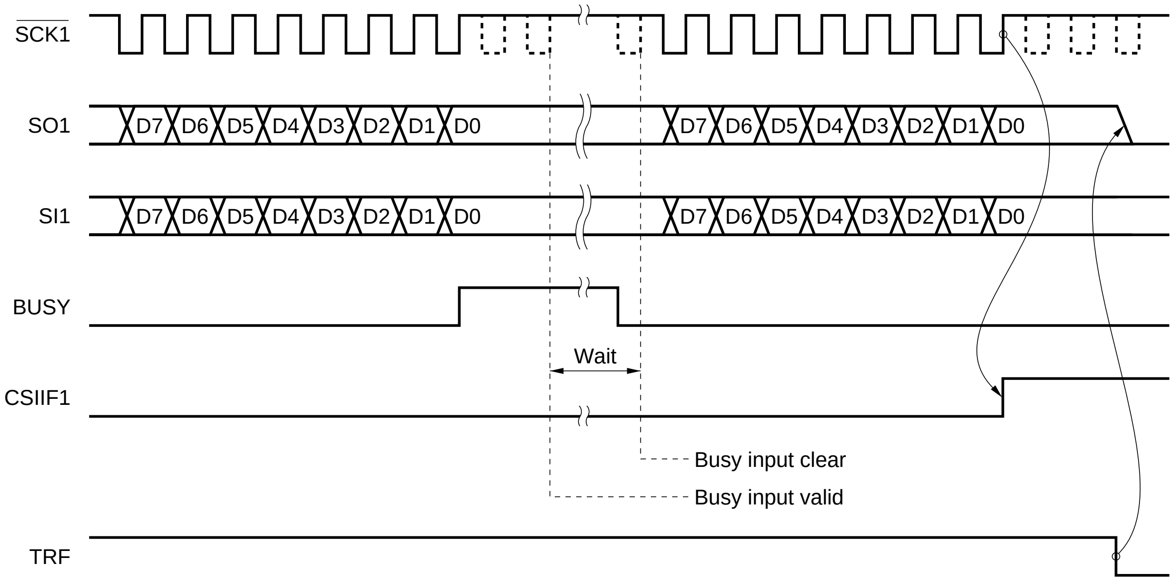
BUSY0 = 0: Active high
BUSY0 = 1: Active low

When using the busy control option, select the internal clock as the serial clock. Busy control cannot be executed with the external clock.

Figure 15-19 shows the operation timings when using the busy control option.

Caution Busy control cannot be executed when the interval time is controlled by using the automatic data transmit/receive interval specify register (ADTI).
If an attempt is made to execute both control operations at the same time, busy control is invalid.

Figure 15-19. Operation Timings when Using Busy Control Option (BUSY0 = 0)

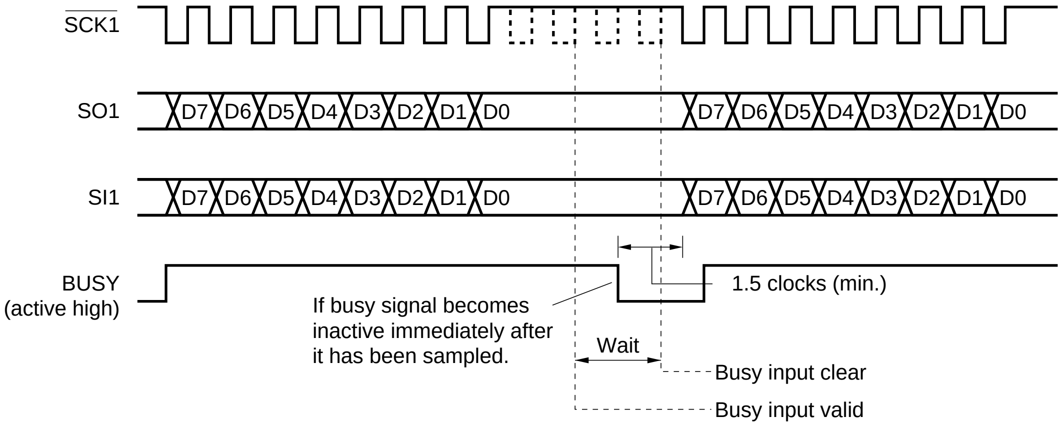


Caution When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag
TRF: Bit 3 of the automatic data transmit/receive control register (ADTC)

When the busy signal becomes inactive, the wait is cleared. If the sampled busy signal is inactive, transmission/reception of 8-bit data is started at the falling edge of the next clock.
Note that, because the busy signal is asynchronous with the serial clock, it takes the master device up to 1 clock to sample the busy signal even if the slave device has made the busy signal inactive. In addition, it takes 0.5 clock until data transfer is started after the signal has been sampled.
To clear the wait, therefore, it is necessary for the slave device to keep the busy signal inactive for at least 1.5 clocks. In Figures 15 to 20, the busy signal becomes active as soon as transmission/reception has been started.

Figure 15-20. Busy Signal and Clearing Wait (BUSY0 = 0)



(b) Busy & strobe control option

Strobe control is to synchronize data transmission/reception between the master device and a slave device. The master device outputs a strobe signal from the STB/P23 pin on completion of transmission/reception of 8-bit data. This strobe signal informs the slave device of the data transmission/reception completion timing of the master device. Therefore, synchronization can be established even if bit slippage occurs due to noise carried on the serial clock, keeping bit slippage from affecting transmission of the next byte.

To use the strobe control option, the following conditions must be satisfied:

- Set bit 5 (ATE) of the serial operating mode register 1 (CSIM1) to 1.
- Set bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 1.

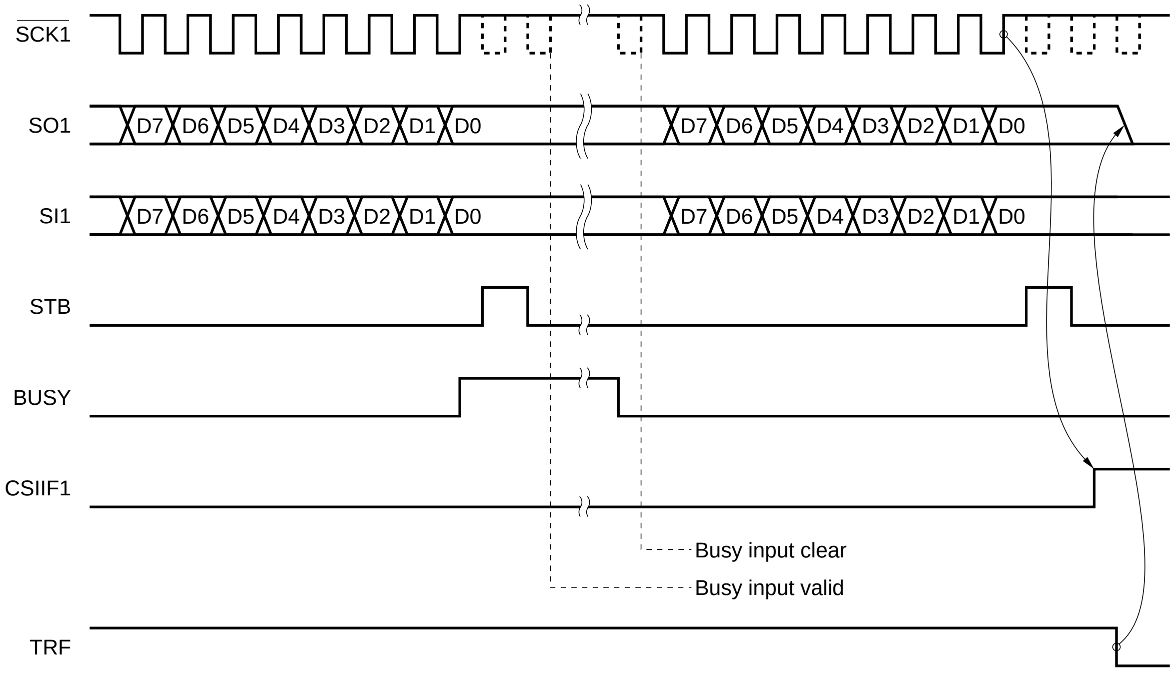
Usually, the busy control and strobe control options are simultaneously used for handshaking. In this case, the strobe signal is output from the STB/P23 pin and the BUSY/P24 pin is sampled. While a busy signal is input to the pin, transmission/reception can be put into a wait state.

If strobe control is not executed, the P23/STB pin can be used as a normal I/O port pin.

Figure 15-21 shows the operation timings when using the busy & strobe control option.

When the strobe control option is used, the interrupt request flag that is set on completion of transmission/reception (CSIIF1) is set after the strobe signal has been output.

Figure 15-21. Operation Timings when Using Busy & Strobe Control Option (BUSY0 = 0)



Caution When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF1: Interrupt request flag
TRF: Bit 3 the automatic data transmit/receive control register (ADTC)

(c) Bit slippage detection function with busy signal

During automatic transmission/reception, bit slippage may take place in the serial clock of the slave device due to the noise carried on the serial clock signal output by the master device. Unless the strobe control option is used at this time, the bit slippage affects transmission of the next byte. In such a case, the master device can detect the bit slippage by using the busy control option and checking the busy signal during transmission.

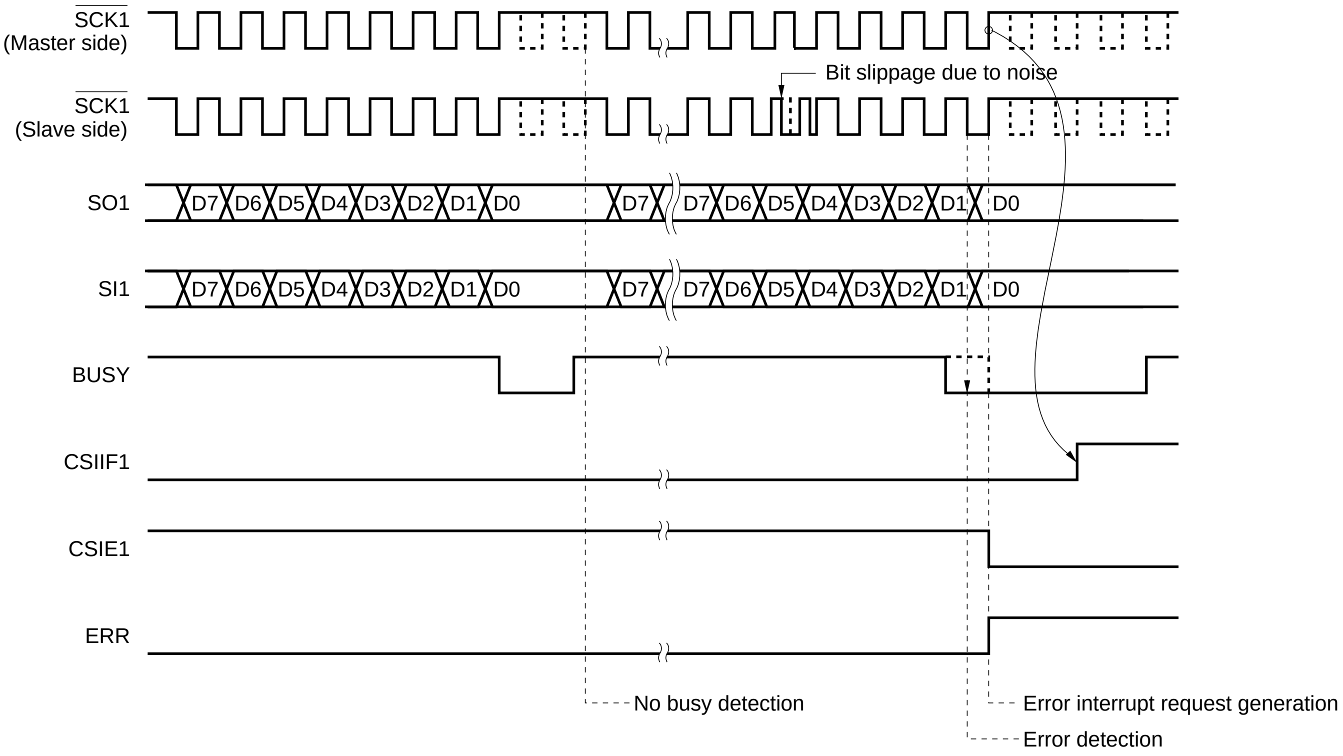
The bit slippage is detected by using the busy signal as follows:

The slave outputs a busy signal after the 8th rise of the serial clock during data transmission/reception (at this time, make the busy signal inactive within two clocks not to make the master device put transmission/reception into a wait state).

The master samples the busy signal in synchronization with the fall of the serial clock. If bit slippage does not occur, the busy signal is found to be inactive after it has been sampled eight times. If the busy signal is found to be active when it has been sampled, it is assumed that bit slippage has occurred, and error processing is performed (by setting bit 4 (ERR) of the automatic data transmit/receive control register (ADTC) to 1).

Figure 15-22 shows the operation timings of the bit slippage detection function using the busy signal.

Figure 15-22. Operation Timings of Bit Slippage Detection Function Using Busy Signal (BUSY0 = 1)


$\overline{SCK1}$
(Master side)
$\overline{SCK1}$
(Slave side)
Bit slippage due to noise
SO1
D7 D6 D5 D4 D3 D2 D1 D0 D7 D7 D6 D5 D4 D3 D2 D1 D0
SI1
D7 D6 D5 D4 D3 D2 D1 D0 D7 D7 D6 D5 D4 D3 D2 D1 D0
BUSY
CSIIF1
CSIE1
ERR
No busy detection
Error interrupt request generation
Error detection

Remark CSIIF1 : Interrupt request flag
CSIE1 : Bit 7 of the serial operating mode register 1 (CSIM1)
ERR : Bit 4 of the automatic data transmit/receive control register (ADTC)

(5) Automatic data transmit/receive interval

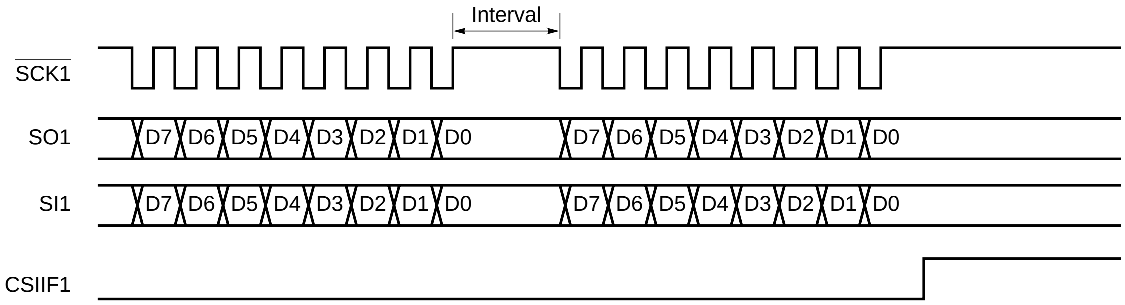
When the automatic data transmit/receive function is used, one byte is transmitted/received and then the read/write operations from/to the buffer RAM are performed, therefore an interval is inserted before the next data transmission/receiving.

When the automatic data transmit/receive function is performed by an internal clock, since the read/write operations from/to the buffer RAM are done in parallel with CPU processing, the interval depends on the CPU processing at the moment of serial clock's eighth positive-edge timing and the value which is set in the automatic data transmit/receive interval specify register (ADTI). Whether or not the interval depends on the ADTI can be selected by setting bit 7 of ADTI (ADTI7).

When ADTI7 is set to 0, the interval depends only on the CPU processing. When ADTI7 is set to 1, the interval is the value determined by the contents of the ADTI or the other value determined by CPU processing, whichever is greater.

When the automatic data transmit/receive function is performed by an external clock, it must be chosen so that the interval may be longer than the value shown in (b).

Figure 15-23. Automatic Data Transmit/Receive Interval



Remark CSIIF1: Interrupt request flag

(a) In case the automatic data transmit/receive function is performed by an internal clock

The internal clock operation is performed when bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) is set to 1.

In this case, the interval is determined as follows by CPU processing.

When bit 7 (ADTI7) of the automatic data transmit/receive interval specify register (ADTI) is set to 0, the interval is determined by CPU processing. When ADTI7 is set to 1, the interval is determined by the contents of ADTI or by CPU processing, whichever is greater. For the interval determined by ADTI, see the format shown in Figure 15-5. Automatic Data Transmit/Receive Interval Specify Register Format.

Table 15-3. Interval Time by CPU Processing (During Internal Clock Operation)

CPU Processing	Interval
When using multiplication instruction	MAX. (2.5 T_{SCK}, 13 T_{CPU})
When using division instruction	MAX. (2.5 T_{SCK}, 20 T_{CPU})
External, access 1 wait mode	MAX. (2.5 T_{SCK}, 9 T_{CPU})
Other than above	MAX. (2.5 T_{SCK}, 7 T_{CPU})

T_{SCK} : 1/f_{SCK}

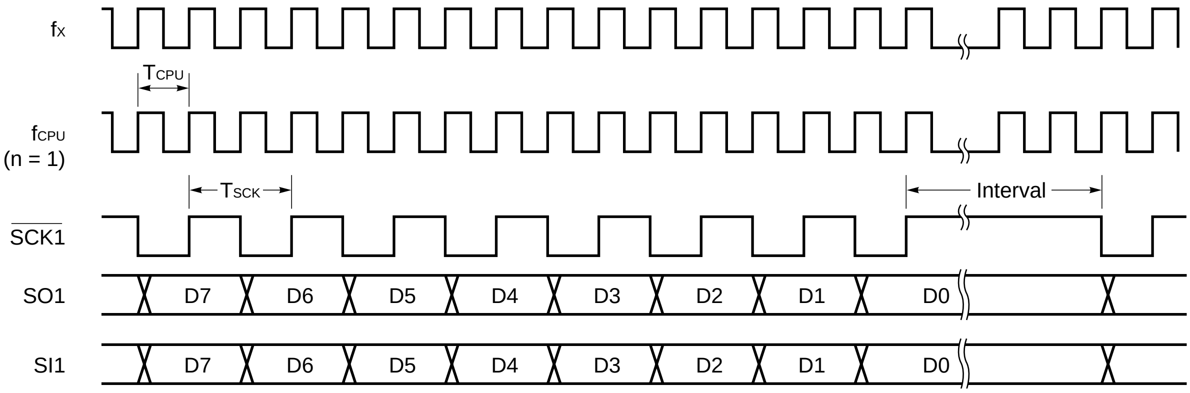
f_{SCK} : Serial clock frequency

T_{CPU} : 1/f_{CPU}

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC)

MAX. (a, b) : a or b, whichever is greater

Figure 15-24. Operation Timing with Automatic Data Transmit/Receive Function Performed by Internal Clock



f_X : Main system clock oscillation frequency

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC)

T_{CPU} : 1/f_{CPU}

T_{SCK} : 1/f_{SCK}

f_{SCK} : Serial clock frequency

(b) In case the automatic data transmit/receive function is performed by an external clock

The external clock operation is performed when bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) is cleared to 0.

When the automatic data transmit/receive function is used by the external clock, it must be selected so that the interval may be longer than the values shown as follows:

Table 15-4. Interval Time by CPU Processing (During External Clock Operation)

CPU Processing	Interval
When using multiplication instruction	13 TCPU or more
When using division instruction	20 TCPU or more
External access 1 wait mode	9 TCPU or more
Other than above	7 TCPU or more

T_{CPU} : $1/f_{CPU}$

f_{CPU} : CPU clock (set by bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC))

CHAPTER 16 FIP CONTROLLER/DRIVER

16.1 FIP Controller/Driver Functions

The functions of the FIP controller/driver incorporated in the μPD78044F Subseries are as follows.

(1) Automatically outputs the segment signals (DMA operation) and digit signals by automatically reading data displayed.

(2) Controls 9- to 24-segment and 2- to 16-digit FIPs (fluorescent indicator panel) using display mode registers 0 and 1 (DSPM0 and DSPM1).

(3) Pins not used for FIP display can be used as output and input/output ports.

(4) Luminance can be adjusted in 8 levels using display mode register 1 (DSPM1).

(5) Incorporates hardware for key scan application.

- Generates interruption signals (INTKS) indicating key scan timing.
- Outputs key scan signals from segment output pins by setting key scan data to port 11 and port 12.
- Detects timings at which key scan data are output by the key scan flag (KSF).

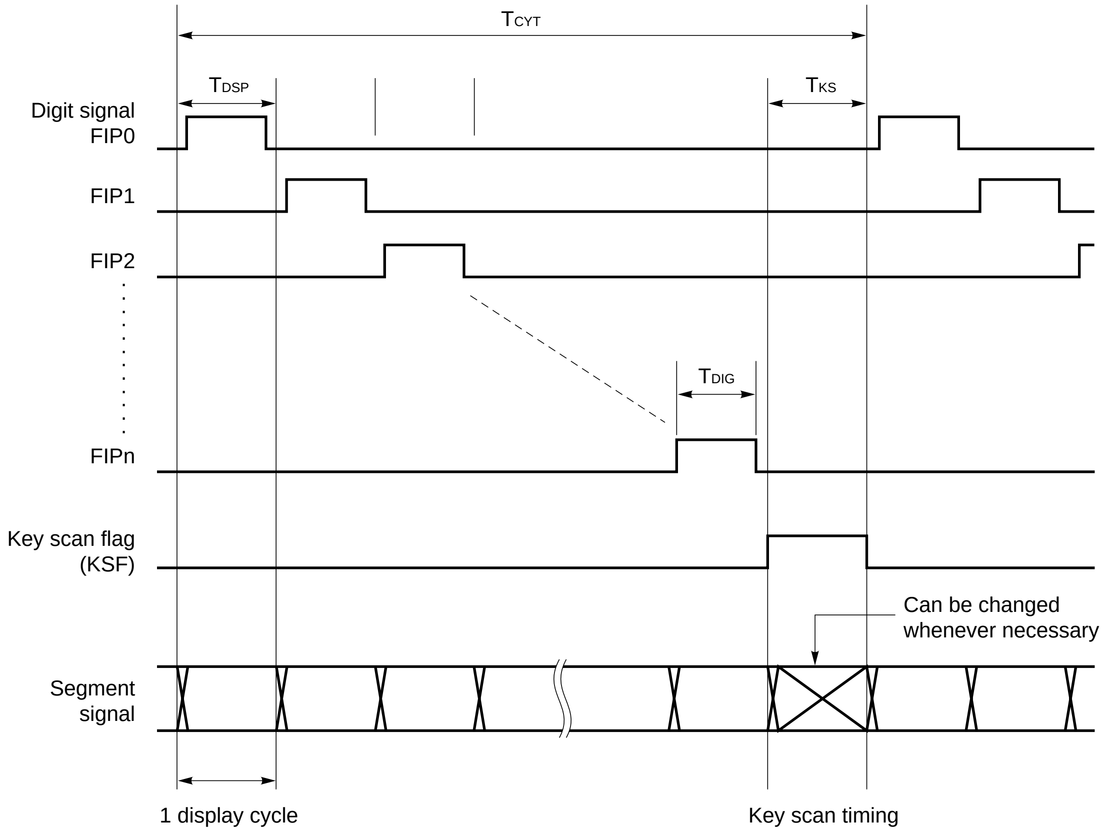
(6) Incorporates a high voltage-resistant output buffer that can directly drive the FIP.

(7) The display output pin can be connected to a pull-down resistor by mask option.

Cautions 1. The FIP controller/driver can be operated only when f_X or $f_X/2$ is selected for the CPU clock. When performing FIP display, set bit 4 (CSS) of the processor control register (PCC) to 0 and bits 2 to 0 (PCC2 to PCC0) of the PCC register to 000 or 001. Data will not be displayed properly if clocks other than these are used (including the subsystem clock). To stop the main oscillation, be sure to specify "display stop" by setting bits 7 to 4 (DIGS3 to DIGS0) of the display mode register 1 (DSPM1) to 0000.

2. When using the FIP controller/driver, set the pins of ports 11 and 12 used as segment outputs to the output mode (set 0 to the bits corresponding to port mode registers 11 and 12).

Figure 16-1. FIP Controller Operation Timings



n : Displayed digit – 1 (Digits 2 to 16 can be selected using the display mode register 1 (DSPM1))

T_{DSP} : 1 display cycle (1024/f_X (204.8 μs: at 5.0-MHz operation) or 2048/f_X (409.6 μs: at 5.0-MHz operation)

T_{KS} : Key scan timing ($T_{KS} = T_{DSP}$)

T_{CYT} : Display cycle ($T_{CYT} = T_{DSP} \times$ (Displayed digit + 1))

T_{DIG} : Width of digit signal pulse (Can be selected from 8 types using the display mode register 1 (DSPM1))

All 34 display output pins are used as port pins as well. These pins are used as port pins when display stop is set using bits 4 to 7 (DIGS0 to DIGS3) of the display mode register 1 (DSPM1).

Even when display is enabled, display output pins not used for outputting digit signals and segment signals can be used as port pins.

Table 16-1. Relation between Display Output Pins and Port Pins

Display Pin Name	Dual Port Name	Input/Output
FIP0, FIP1	P80, P81	For output port
FIP2 to FIP9	P90 to P97	For output port
FIP10 to FIP17	P100 to P107	For output port
FIP18 to FIP25	P110 to P117	I/O port
FIP26 to FIP33	P120 to P127	I/O port

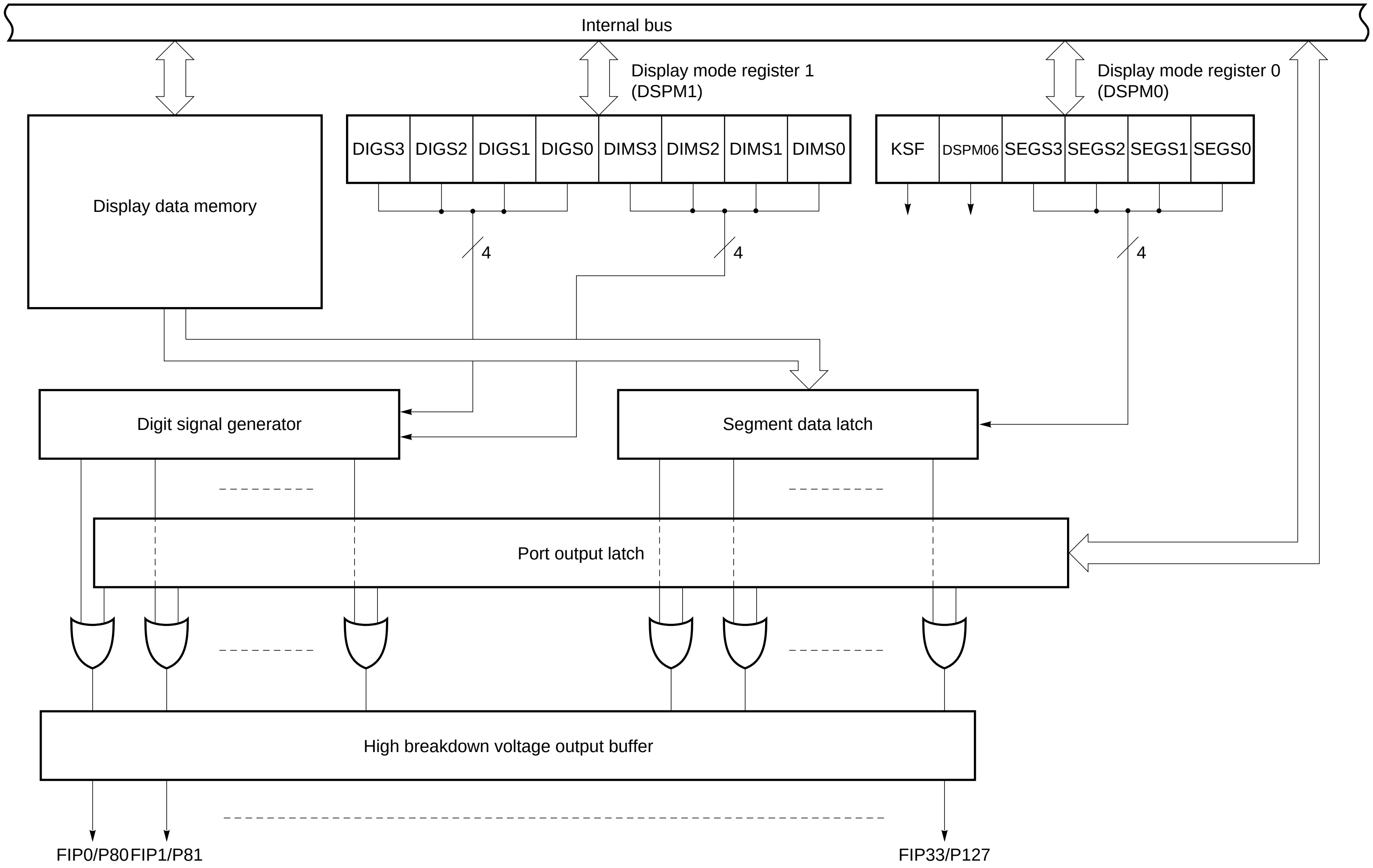
16.2 FIP Controller/Driver Configuration

The FIP controller/driver consists of the following hardware.

Table 16-2. FIP Controller/Driver Configuration

Item	Configuration
Display output	34 pins (segments: 9 to 24, digits: 2 to 16)
Control register	Display mode register 0 (DSPM0) Display mode register 1 (DSPM1)

Figure 16-2. FIP Controller/Driver Block Diagram



16.3 FIP Controller/Driver Control Registers

16.3.1 Control registers

There are two types of registers for controlling the FIP controller/driver.

- Display mode register 0 (DSPM0)
- Display mode register 1 (DSPM1)

(1) Display mode register 0 (DSPM0) (Refer to **Figure 16-3**.)

This register sets the display segment number and the mode for the subsystem clock noise elimination circuit, and displays the key scan timing.

The DSPM0 is set with an 8-bit memory manipulation instruction. However, only bit 7 (KSF) can be read using the 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears DSPM0 to 00H.

(2) Display mode register 1 (DSPM1) (Refer to **Figure 16-4**.)

This register sets the display digit number and the cut width of the digit signal, and selects the display cycle (T_{DSP}).

When bit 0 (DIMS0) is set to 1 and the display cycle to 2048/fx (409.6 μs: at 5.0-MHz operation), light leakage is reduced.

As the display cycle approaches the commercial power supply frequency when the display digit is increased, the display will flicker. In this case, select 1024/f_X (204.8 μs: at 5.0-MHz operation). If light leaks, adjust the cut width of the digit signal using bits 1 to 3 (DIMS1 to DIMS3).

DSPM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears DSPM1 to 00H.

Figure 16-3. Display Mode Register 0 Format

Symbol	⑦	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM0	KSF	DSPM06	0	0	SEGS3	SEGS2	SEGS1	SEGS0	FFA0H	00H	R/W Note 1

SEGS3	SEGS2	SEGS1	SEGS0	Display Segment
0	0	0	0	9 segments
0	0	0	1	10 segments
0	0	1	0	11 segments
0	0	1	1	12 segments
0	1	0	0	13 segments
0	1	0	1	14 segments
0	1	1	0	15 segments
0	1	1	1	16 segments
1	0	0	0	17 segments
1	0	0	1	18 segments
1	0	1	0	19 segments
1	0	1	1	20 segments
1	1	0	0	21 segments
1	1	0	1	22 segments
1	1	1	0	23 segments
1	1	1	1	24 segments

DSPM06	Subsystem Clock Noise Reduction Circuit Mode Setting Note 2
0	2.5 MHz < f_X ≤ 5.0 MHz
1	1.25 MHz < f_X ≤ 2.5 MHz

KSF	Timing Status
0	Display timing
1	Key scan timing

Notes 1. Bit 7 (KSF) is for read-only.

2. Set the values according to the main system clock frequency (fx) used. The noise elimination circuit is effective during FIP display operations.

Caution When using a main system clock below 1.25 MHz together with the FIP controller/driver, be sure to use the main system clock (TCL24 = 0) for counting the clock timer.

Remark f_X : Main system clock oscillation frequency

Figure 16-4. Display Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
DSPM1	DIGS3	DIGS2	DIGS1	DIGS0	DIMS3	DIMS2	DIMS1	DIMS0	FFA1H	00H	R/W

DIMS0	Display Cycle Selection
0	1024/f_X is 1 display cycle. (1 display cycle = 204.8 μs: when operated at 5.0 MHz)
1	2048/f_X is 1 display cycle. (1 display cycle = 409.6 μs: when operated at 5.0 MHz)

DIMS3	DIMS2	DIMS1	Digit Signal Cut Width
0	0	0	1/16
0	0	1	2/16
0	1	0	4/16
0	1	1	6/16
1	0	0	8/16
1	0	1	10/16
1	1	0	12/16
1	1	1	14/16

DIGS3	DIGS2	DIGS1	DIGS0	Display Digit
0	0	0	0	Display stopped (Static display)[Note]
0	0	0	1	2 digits
0	0	1	0	3 digits
0	0	1	1	4 digits
0	1	0	0	5 digits
0	1	0	1	6 digits
0	1	1	0	7 digits
0	1	1	1	8 digits
1	0	0	0	9 digits
1	0	0	1	10 digits
1	0	1	0	11 digits
1	0	1	1	12 digits
1	1	0	0	13 digits
1	1	0	1	14 digits
1	1	1	0	15 digits
1	1	1	1	16 digits

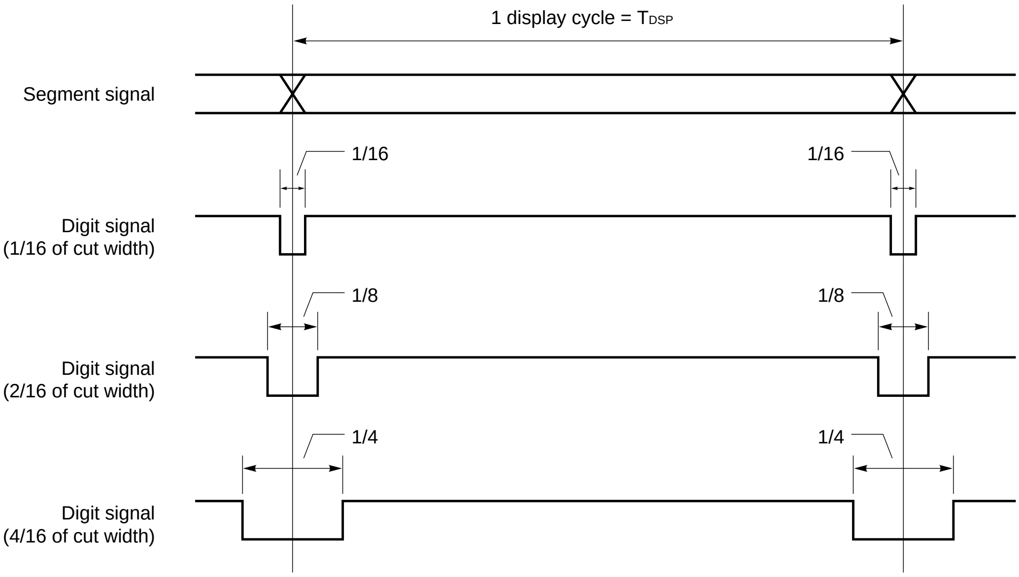
Note When setting display stop, static display can be set by setting the port output latch.

Remark f_X : Main system clock oscillation frequency

16.3.2 1 Display period and blanking width

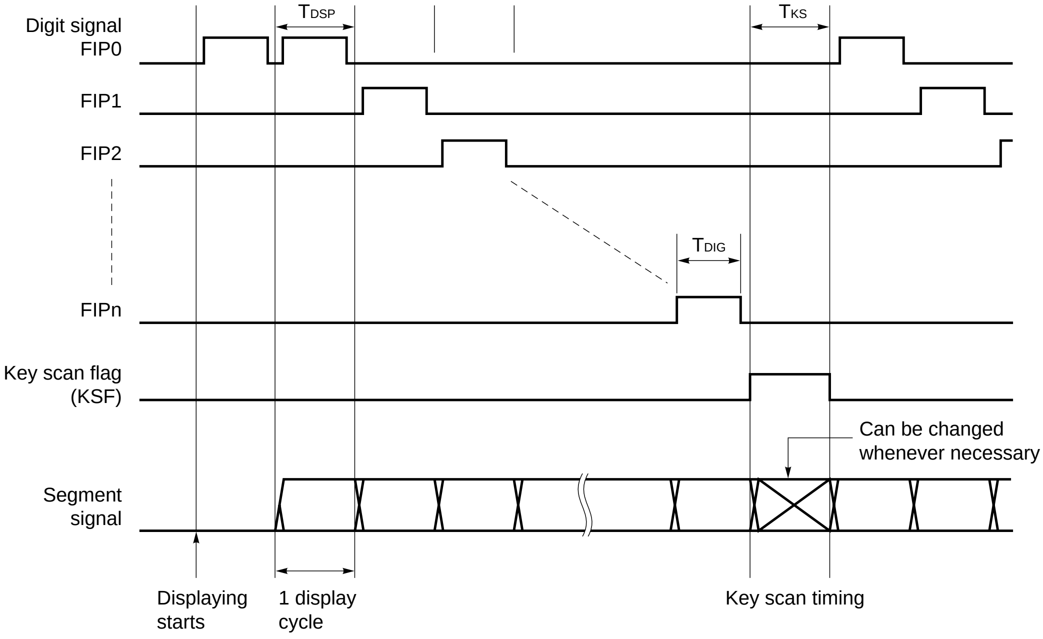
The digit signal is equally cut at the beginning and end of the display period by the cut width set by bits 1 to 3 (DIMS1 to DIMS3) of the display mode register 1 (DSPM1).

Figure 16-5. Cut Width of Digit Signal



Remark FIP0 is output twice when displaying is started from the display stop status.

Figure 16-6. FIP Controller Display Start Timings



n : Displayed digit – 1 (Digits 2 to 16 can be selected using the display mode register 1 (DSPM1))

T_{DSP} : 1 display cycle (1024/f_X (204.8 μs: at 5.0-MHz operation) or 2048/f_X (409.6 μs: at 5.0-MHz operation))

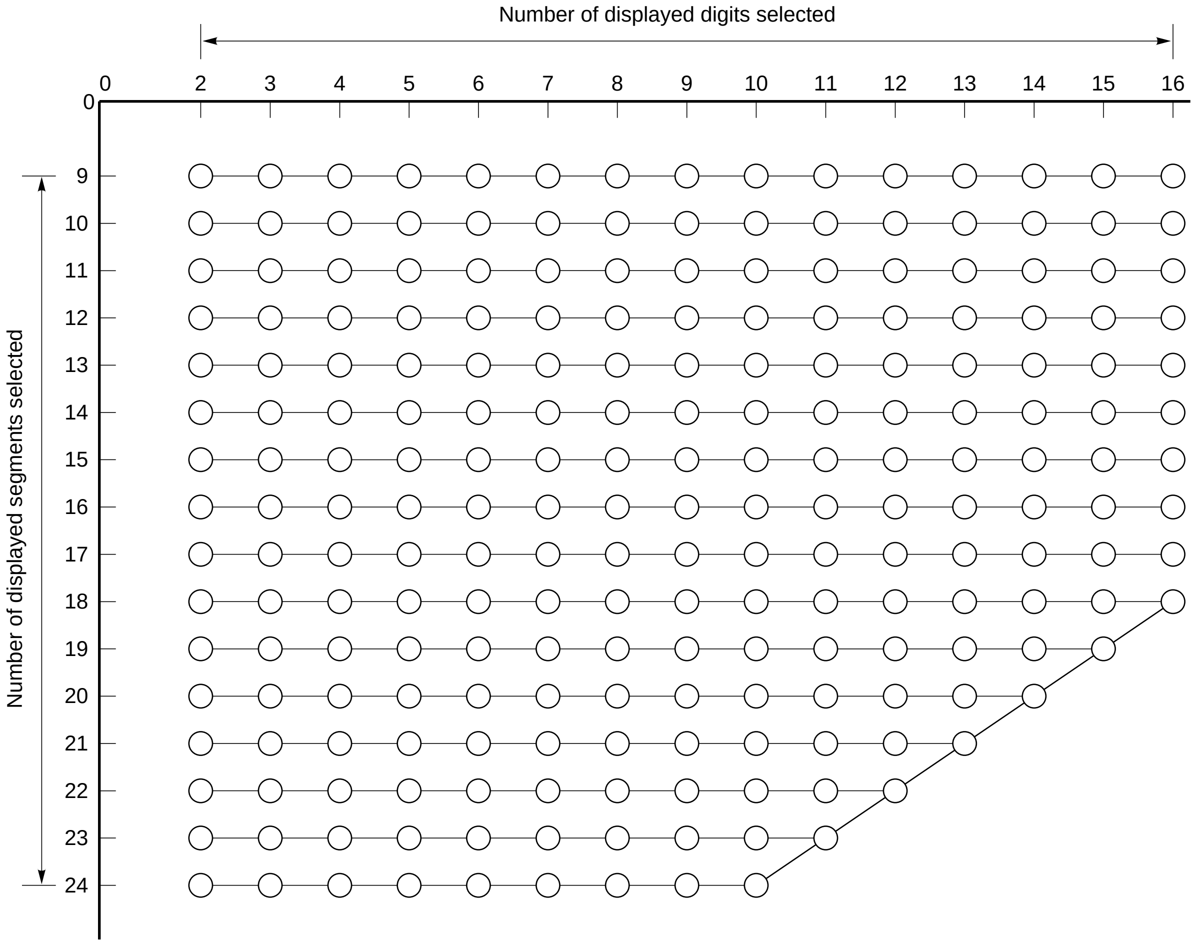
T_{KS} : Key scan timing (T_{KS} = T_{DSP})

T_{DIG} : Width of digit signal pulse (Can be selected from 8 types using the display mode register 1 (DSPM1))

16.4 Selecting the Display Mode

The number of segments and digits displayed by the FIP controller/driver depend on the display mode set.

Figure 16-7. Selection of Display Mode


Number of displayed digits selected
0
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
0
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Number of displayed segments selected

Caution When the total number of digits and segments together exceed 34, the digits will have the priority.

16.5 Display Mode and Display Output

The on-chip controller/driver assigns pins FIP0/P80 to FIP33/P127 to digit signals and segment signals (in this order). The number assigned is specified by the display mode registers 0 and 1 (DSPM0 and DSPM1). The remaining pins are assigned as general ports.

The pin configuration for a 14-segment display is shown below as an example.

Figure 16-8. Pin Configuration for 14-Segment Display

Pin Name	Number of Displayed Digits Selected								
	Display stop	2	3	--------	9	10	11	--------	16
FIP0/P80	P80	T0∨P80	T0∨P80		T0∨P80	T0∨P80	T0∨P80		T0∨P80
FIP1/P81	P81	T1∨P81	T1∨P81		T1∨P81	T1∨P81	T1∨P81		T1∨P81
FIP2/P90	P90	P90	T2∨P82		T2∨P90	T2∨P90	T2∨P90		T2∨P90
FIP3/P91	P91	P91	P91		T3∨P91	T3∨P91	T3∨P91		T3∨P91
FIP4/P92	P92	P92	P92		T4∨P92	T4∨P92	T4∨P92		T4∨P92
FIP5/P93	P93	P93	P93		T5∨P93	T5∨P93	T5∨P93		T5∨P93
FIP6/P94	P94	P94	P94		T6∨P94	T6∨P94	T6∨P94		T6∨P94
FIP7/P95	P95	P95	P95		T7∨P95	T7∨P95	T7∨P95		T7∨P95
FIP8/P96	P96	P96	P96		T8∨P96	T8∨P96	T8∨P96		T8∨P96
FIP9/P97	P97	P97	P97		P97	T9∨P97	T9∨P97		T9∨P97
FIP10/P100	P100	S0∨P100	S0∨P100		S0∨P100	S0∨P100	T10∨P100		T10∨P100
FIP11/P101	P101	S1∨P101	S1∨P101		S1∨P101	S1∨P101	S0∨P101		T11∨P101
FIP12/P102	P102	S2∨P102	S2∨P102		S2∨P102	S2∨P102	S1∨P102		T12∨P102
FIP13/P103	P103	S3∨P103	S3∨P103		S3∨P103	S3∨P103	S2∨P103		T13∨P103
FIP14/P104	P104	S4∨P104	S4∨P104		S4∨P104	S4∨P104	S3∨P104		T14∨P104
FIP15/P105	P105	S5∨P105	S5∨P105		S5∨P105	S5∨P105	S4∨P105		T15∨P105
FIP16/P106	P106	S6∨P106	S6∨P106	--------	S6∨P106	S6∨P106	S5∨P106	--------	S0∨P106
FIP17/P107	P107	S7∨P107	S7∨P107		S7∨P107	S7∨P107	S6∨P107		S1∨P107
FIP18/P110	P110	S8 ∨ P110	S8 ∨ P110		S8 ∨ P110	S8 ∨ P110	S7 ∨ P110		S2 ∨ P110
FIP19/P111	P111	S9 ∨ P111	S9 ∨ P111		S9 ∨ P111	S9 ∨ P111	S8 ∨ P111		S3 ∨ P111
FIP20/P112	P112	S10 ∨ P112	S10 ∨ P112		S10 ∨ P112	S10 ∨ P112	S9 ∨ P112		S4 ∨ P112
FIP21/P113	P113	S11 ∨ P113	S11 ∨ P113		S11 ∨ P113	S11 ∨ P113	S10 ∨ P113		S5 ∨ P113
FIP22/P114	P114	S12 ∨ P114	S12 ∨ P114		S12 ∨ P114	S12 ∨ P114	S11 ∨ P114		S6 ∨ P114
FIP23/P115	P115	S13 ∨ P115	S13 ∨ P115		S13 ∨ P115	S13 ∨ P115	S12 ∨ P115		S7 ∨ P115
FIP24/P116	P116	P116	P116		P116	P116	S13 ∨ P116		S8 ∨ P116
FIP25/P117	P117	P117	P117		P117	P117	P117		S9 ∨ P117
FIP26/P120	P120	P120	P120		P120	P120	P120		S10 ∨ P120
FIP27/P121	P121	P121	P121		P121	P121	P121		S11 ∨ P121
FIP28/P122	P122	P122	P122		P122	P122	P122		S12 ∨ P122
FIP29/P123	P123	P123	P123		P123	P123	P123		S13 ∨ P123
FIP30/P124	P124	P124	P124		P124	P124	P124		P124
FIP31/P125	P125	P125	P125		P125	P125	P125		P125
FIP32/P126	P126	P126	P126		P126	P126	P126		P126
FIP33/P127	P127	P127	P127		P127	P127	P127		P127

Remark ∨ : OR

16.6 Display Data Memory

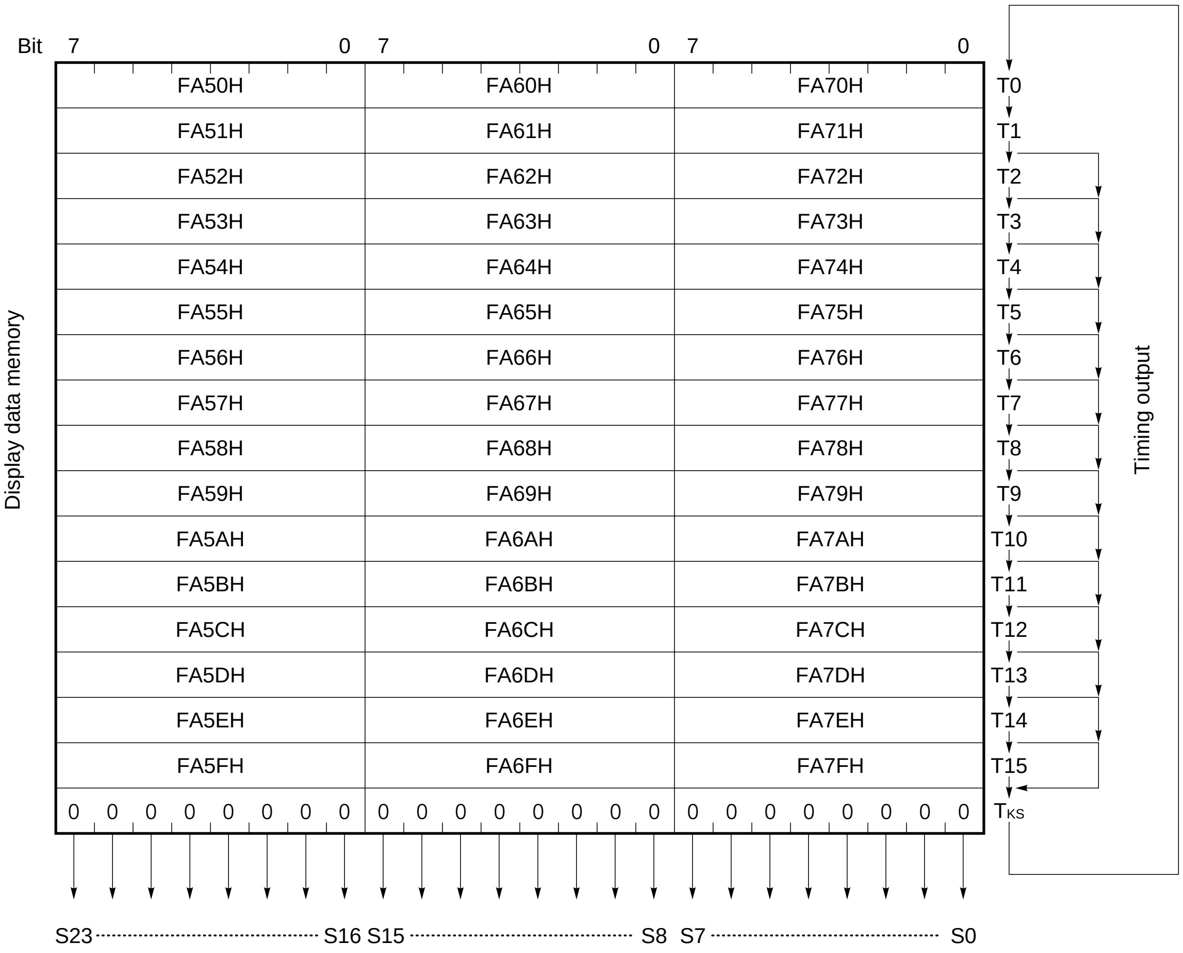
The display data memory is the area for storing the segment data to be displayed.

This memory is mapped at addresses FA50H to FA7FH. To display data on the FIP, the FIP controller reads the data stored in this memory regardless of the type of operations performed by the CPU (DMA operations).

The area not used for the display data can be used as a normal RAM.

At the key scan timing (T_{KS}), all segment outputs and digit outputs become “0” and the output latch data of ports 11 and 12 are output to FIP18/P110 to FIP33/P127.

Figure 16-9. Contents of Display Data Memory and Relation with Segment Output



Bit 7 … 0	Bit 7 … 0	Bit 7 … 0	
FA50H	FA60H	FA70H	T0
FA51H	FA61H	FA71H	T1
FA52H	FA62H	FA72H	T2
FA53H	FA63H	FA73H	T3
FA54H	FA64H	FA74H	T4
FA55H	FA65H	FA75H	T5
FA56H	FA66H	FA76H	T6
FA57H	FA67H	FA77H	T7
FA58H	FA68H	FA78H	T8
FA59H	FA69H	FA79H	T9
FA5AH	FA6AH	FA7AH	T10
FA5BH	FA6BH	FA7BH	T11
FA5CH	FA6CH	FA7CH	T12
FA5DH	FA6DH	FA7DH	T13
FA5EH	FA6EH	FA7EH	T14
FA5FH	FA6FH	FA7FH	T15
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	T_{KS}

16.7 Key Scan Flags and Key Scan Data

16.7.1 Key scan flags

The key scan flag (KSF) is set to 1 in the key scan timing and reset automatically to 0 in the display timing.

The KSF is mapped at bit 7 of the display mode register 0 (DSPM0) and can be tested one bit at a time. It cannot be written.

By testing the KSF, it can be determined if it is in the key scan timing and if the data input using keys is correct.

16.7.2 Key scan data

The data stored in ports 11 and 12 are output from pins FIP18 to FIP33 in the key scan timing.

By changing the data output from ports 11 and 12 during the key scan timing, key scan can be performed using these pins FIP18 to FIP33.

Caution Be sure to set the output latches of ports 11 and 12 to 0 before the key scan timing is completed.

16.8 Light Leakage of FIP

Light may leak when a FIP is driven using the μPD78044F Subseries. Two possible causes are as follows.

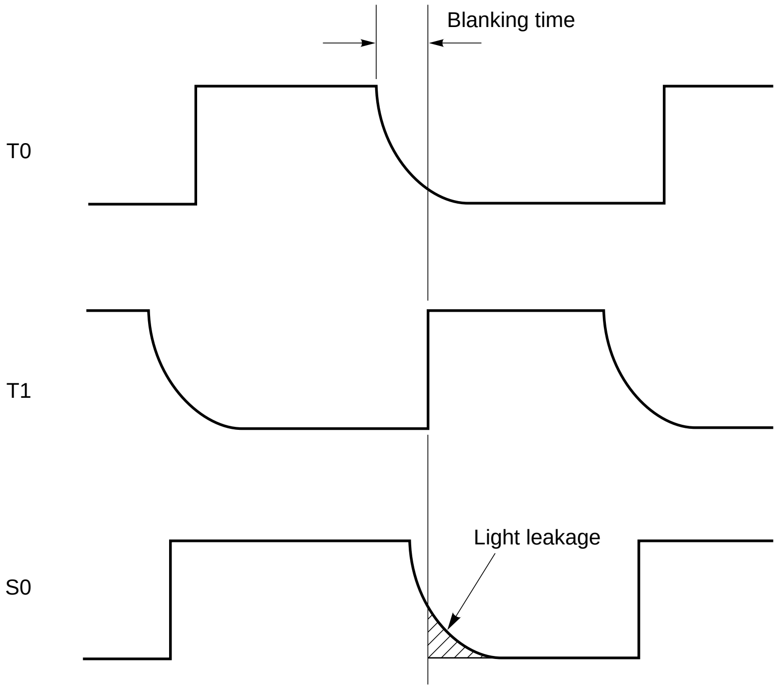
(1) Light leakage due to a short blanking

Figure 16-10 shows the signal waveforms when only the first digit of two digits to be displayed is lit.

As shown in this figure, when the blanking time is short, the T1 signal rises before the segment signal is eliminated, resulting in light leaking.

Generally, as approximately 20 *m*s is required for the blanking time, consider values for setting the display mode register 1 (DSPM1) carefully.

Figure 16-10. Light Leakage due to Short Blanking Time



(2) Light leakage due to capacity between segment and grid of FIP

As shown in Figure 16-12, light may leak even if the blanking time is sufficient. As shown by C_{SG} in Figure 16-11, as there is capacity between the grid and segment of the FIP, the timing signal pin voltage will be increased via C_{SG} when the segment signal turns on.

As shown in Figure 16-12, when this voltage exceeds the cut-off voltage (E_K), light will leak.

This spike noise voltage depends on the size of C_{SG} and the on-chip pull-down resistor (R_L). The greater the value of C_{SG} or the RL value, the greater is the voltage, making it easy for light to leak.

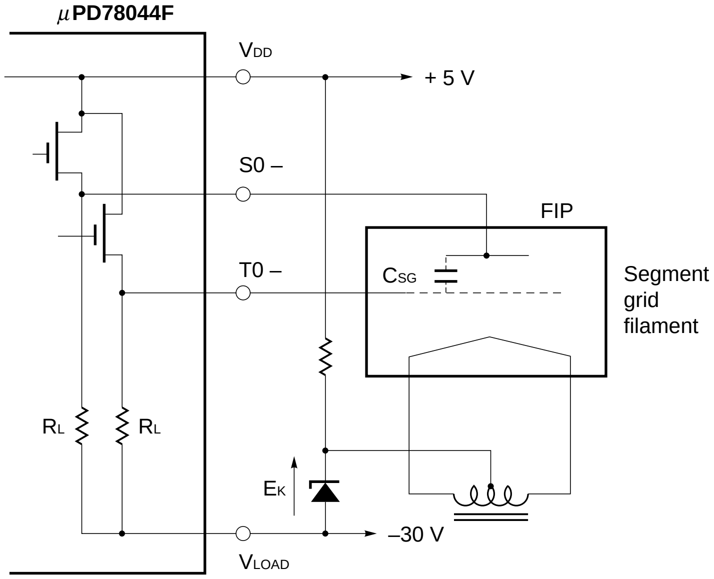
This C_{SG} value differs according to the area of the data displayed on the FIP. The greater the area, the greater the C_{SG} is. Consequently, pull-down resistor values with which light will not leak also depend on the size of the FIP.

As the value of the pull-down resistor incorporated by the mask option is comparatively great, in some cases, light leakage cannot be controlled with the resistance only.

If the quality of the display is insufficient, increase the back bias (increase the E_K), place a filter over the FIP, or attach a 10-kΩ pull-down resistor externally to the timing signal pin.

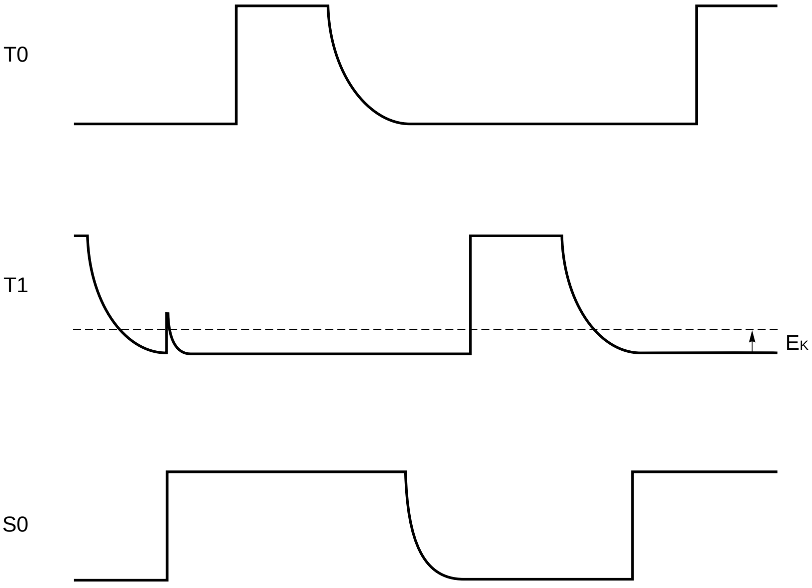
Depending on the duty cycle of the spike noise voltage for the whole display period, the easiness with which light leaks due to the C_{SG} varies. The lesser the number of digits displayed, the easier it is for light to leak. Lowering the luminance of the display is also effective.

Figure 16-11. Light Leakage due to C_{SG}



Remark E_K: Cut-off voltage
R_L: On-chip pull-down resistor

Figure 16-12. Waveform of Light Leakage due to C_{SG}



16.9 Calculating Total Power Dissipation

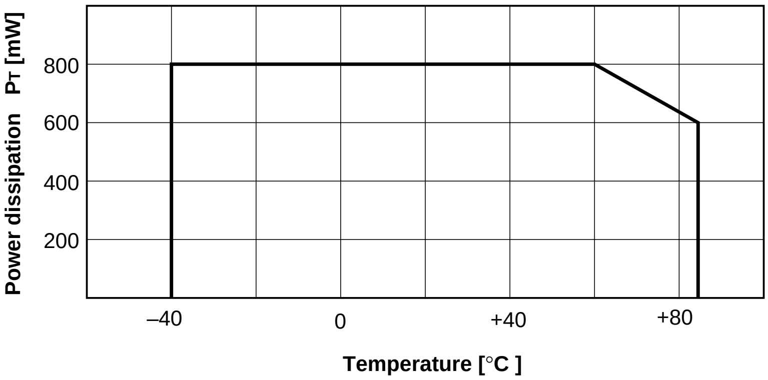
The total power dissipation of the μPD78044F Subseries is the sum of the values at the following three parts. Design your application set so that the sum is lower than the total power dissipation P_T stipulated in Figure 16-13. (The recommended operating condition is 80% or lower of the rated value.)

[1] CPU: the power consumed by CPU and calculated with V_{DD} (max.) x I_{DD} (max.)

[2] Output pins: the power consumption when the maximum current flows at display output pins

[3] Pull-down resistors: the power dissipated at the on-chip pull-down resistors connected to display output pins

Figure 16-13. Allowable Total Power Dissipation P_T (T_A = –40 to +85 °C)



The calculation example of the total power dissipation is provided below.

★ The calculation of total power dissipation of the display example in Figure 16-14 is shown below.

Example: The following conditions are assumed

V_{DD} = 5 V ± 10%, 5.0 MHz oscillation

Power supply current (I_{DD}) = 21.6 mA

Display output: 11 grids × 10 segments (when cut width = 1/16: DIMS1 – DIMS3 = 000B)

Max. 15 mA flows in grid pins.

Max. 3 mA flows in segment pins.

And display output voltage is off at key scan timing.

Display output voltage : grid V_{OD} = V_{DD} – 2 V (with 2 V voltage drop)

segment V_{OD} = V_{DD} – 0.4 V (with 0.4 V voltage drop)

FIP voltage (V_{LOAD}) = –30 V

Mask option pull-down resistor = 25 kΩ

The conditions above are applied to the expressions <1> to <3> to calculate total power dissipation.

<1> CPU power dissipation: 5.5 V × 21.6 mA = 118.8 mW

<2> Output pin power dissipation:

$$\text{Grid}\qquad (V_{DD} - V_{OD}) \times \frac{\text{Total current value of each grid}}{\text{Number of grids + 1}} \times \text{Digit width (1 – Cut width)} =$$

$$2\text{ V} \times \frac{15\text{ mA} \times 11\text{ grids}}{11\text{ grids} + 1} \times (1 - \frac{1}{16}) = 25.8\,\text{mW}$$

$$\text{Segment } (V_{DD} - V_{OD}) \times \frac{\text{Total segment current value of dots lit}}{\text{Number of grids + 1}} =$$

$$0.4\text{ V} \times \frac{3\text{ mA} \times 31\text{ dots}}{11\text{ grids} + 1} = 3.1\text{ mW}$$

<3> Pull-down resistor power dissipation:

$$\text{Grid}\qquad \frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{Number of grids}}{\text{Number of grids + 1}} \times \text{Digit width} =$$

$$\frac{(5.5\text{ V} - 2\text{ V} - (-30\text{ V}))^2}{25\text{ k}\Omega} \times \frac{11\text{ grids}}{11\text{ grids} + 1} \times (1 - \frac{1}{16}) = 38.6\text{ mW}$$

$$\text{Segment}\qquad \frac{(V_{OD} - V_{LOAD})^2}{\text{Pull-down resistor value}} \times \frac{\text{Number of dots lit}}{\text{Number of grids + 1}} =$$

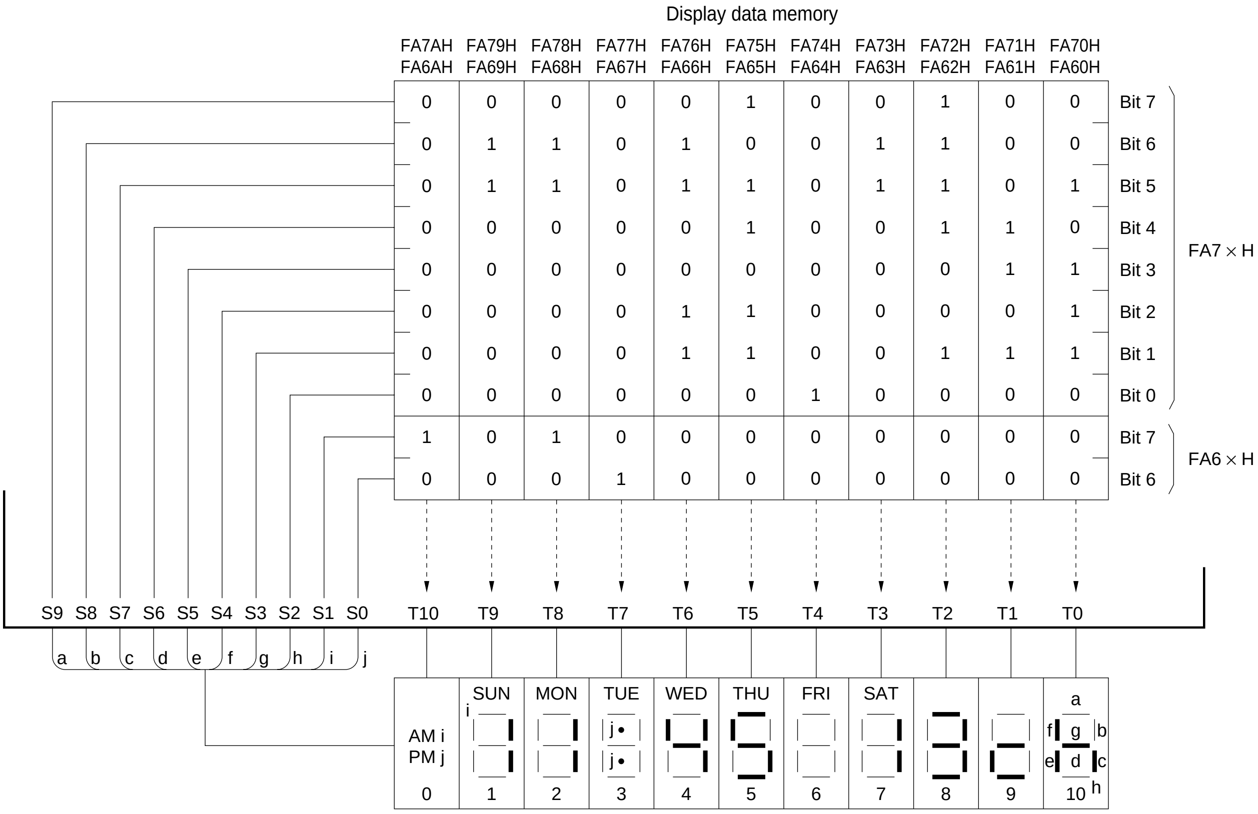
$$\frac{(5.5\text{ V} - 0.4\text{ V} - (-30\text{ V}))^2}{25\text{ k}\Omega} \times \frac{31\text{ dots}}{11\text{ grids} + 1} = 127.3\text{ mW}$$

Total power dissipation = <1> + <2> + <3> = 118.8 + 25.8 + 3.1 + 38.6 + 127.3 = 313.6 mW
(< P_T = 600 mW)

In this example, there is no problem as far as the power consumption is concerned because the total power consumption does not exceed the allowable permissible total consumption rating shown in Figure 16-13.

If the total power dissipation exceeds the of allowable total dissipation rating, the number of pull-down resistors must be reduced so that as to decrease power dissipation.

Figure 16-14. Display Data Memory Contents and Segment Outputs in 10-Segment/11-Digit Display Mode



FA7AH FA6AH	FA79H FA69H	FA78H FA68H	FA77H FA67H	FA76H FA66H	FA75H FA65H	FA74H FA64H	FA73H FA63H	FA72H FA62H	FA71H FA61H	FA70H FA60H		
0	0	0	0	0	1	0	0	1	0	0	Bit 7	FA7 × H
0	1	1	0	1	0	0	1	1	0	0	Bit 6	
0	1	1	0	1	1	0	1	1	0	1	Bit 5	
0	0	0	0	0	1	0	0	1	1	0	Bit 4	
0	0	0	0	0	0	0	0	0	1	1	Bit 3	
0	0	0	0	1	1	0	0	0	0	1	Bit 2	
0	0	0	0	1	1	0	0	1	1	1	Bit 1	
0	0	0	0	0	0	1	0	0	0	0	Bit 0	
1	0	1	0	0	0	0	0	0	0	0	Bit 7	FA6 × H
0	0	0	1	0	0	0	0	0	0	0	Bit 6	

[MEMO]

CHAPTER 17 INTERRUPT FUNCTIONS AND TEST FUNCTION

17.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is unconditionally acknowledged even in an interrupt disabled status. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

It generates a standby release signal.

One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specify flag register (PR0L, PR0H).

Multiple high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (see **Table 17-1**).

A standby release signal is generated.

Four external interrupt sources and ten internal interrupt sources are incorporated as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt to be generated by executing the BRK instruction. It is acknowledged even in an interrupt disabled state. The software interrupt does not undergo interrupt priority control.

17.2 Interrupt Sources and Configuration

A total of 16 non-maskable, maskable and software interrupt sources are incorporated in the interrupt sources (see **Table 17-1**).

Table 17-1. Interrupt Source List

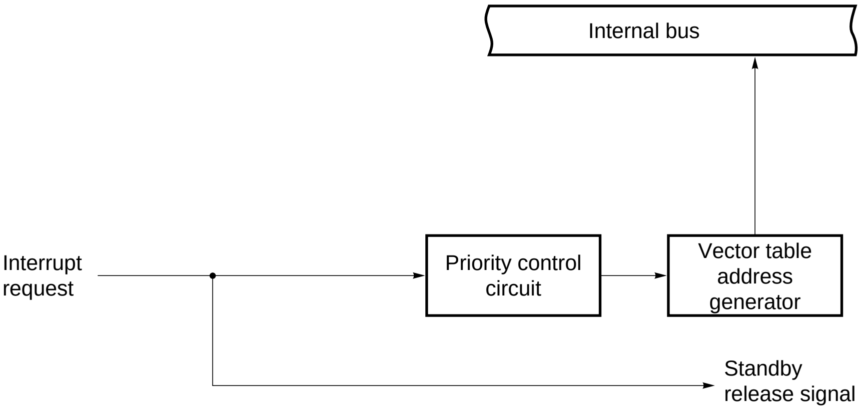
Interrupt Type	Default Priority[Note]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type [Note 2]
		Name	Trigger			
Non-maskable	—	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	(D)
	3	INTP2			000AH	
	4	INTP3			000CH	
		INTUD	6-bit up/down counter match signal generation	Internal		(B)
	5	INTCSI0	End of serial interface channel 0 transfer		000EH	
	6	INTCSI1	End of serial interface channel 1 transfer		0010H	
	7	INTTM3	Reference time interval signal from watch timer		0012H	
	8	INTTM0	16-bit timer/event counter match signal generation		0014H	
	9	INTTM1	8-bit timer/event counter 1 match signal generation		0016H	
	10	INTTM2	8-bit timer/event counter 2 match signal generation		0018H	
	11	INTAD	End of A/D converter conversion		001AH	
	12	INTKS	Key scan timing from FIP controller/driver		001CH	
Software	—	BRK	Execution of BRK instruction	—	003EH	(E)

Notes 1. Default priorities are intended for two or more simultaneously generated maskable interrupt requests. 0 is the highest priority and 12 is the lowest priority.

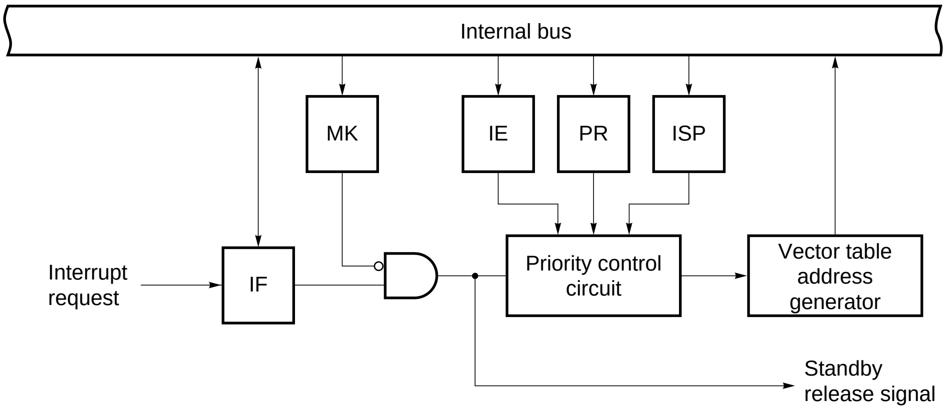
2. Basic configuration types (A) to (E) correspond to (A) to (E) on next page.

Figure 17-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

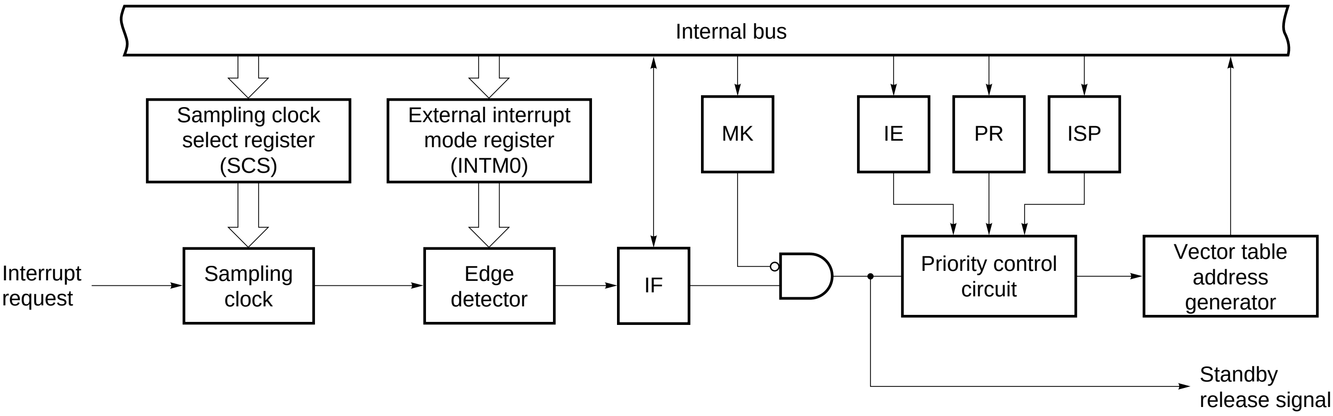
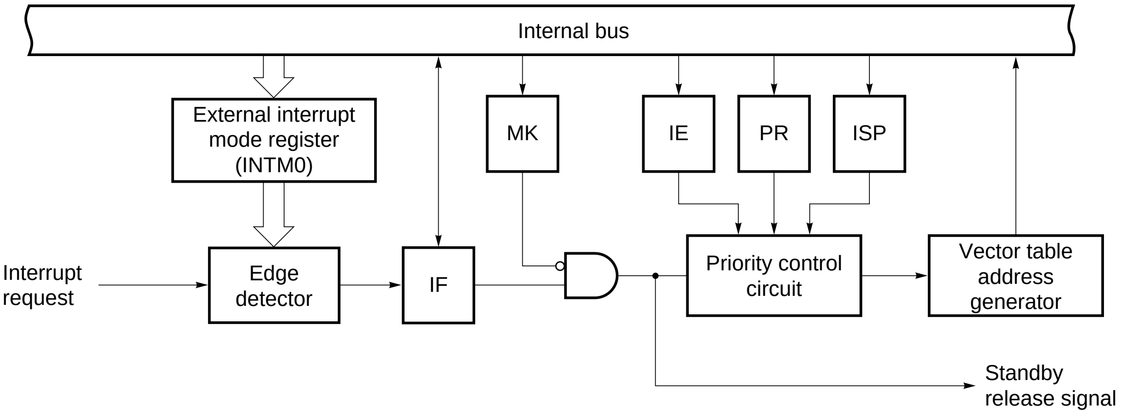
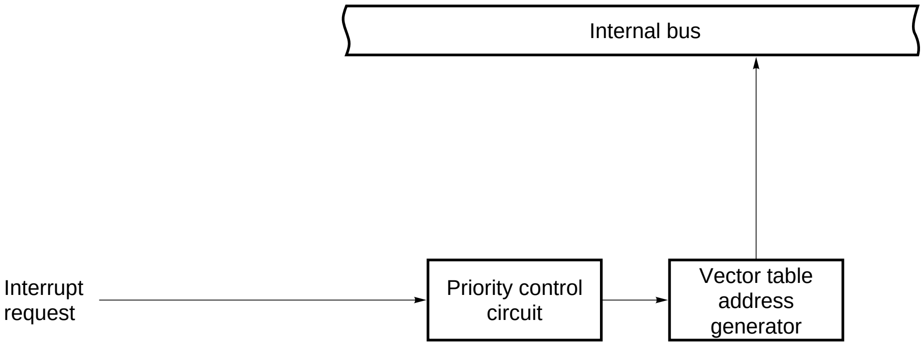


Figure 17-1. Basic Configuration of Interrupt Function (2/2)

(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 specify flag

17.3 Interrupt Function Control Registers

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H)
- Interrupt mask flag register (MK0L, MK0H)
- Priority specify flag register (PR0L, PR0H)
- External interrupt mode register (INTM0)
- Sampling clock select register (SCS)
- Program status word (PSW)

Table 17-2 gives a listing of interrupt request flags, interrupt mask flags and priority specify flag names corresponding to interrupt request sources.

Table 17-2. Various Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specify Flag	
		Register		Register		Register
INTWDT	TMIF4	IF0L	TMMK4	MK0L	TMPR4	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3/INTUD	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	
INTKS	KSIF		KSMK		KSPR	

(1) Interrupt request flag registers (IF0L, IF0H)

The interrupt request flag is set to (1) when the corresponding interrupt request is generated or an instruction is executed. It is cleared to (0) when an instruction is executed upon acknowledgment of an interrupt request or upon application of $\overline{\text{RESET}}$ input.

IF0L or IF0H is set with a 1-bit or 8-bit manipulation instruction. If these two 8-bit registers are used as a single jointed 16-bit register IF0, use a 16-bit memory manipulation instruction to set it.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 17-2. Interrupt Request Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
IF0L	TMIF3	CSIIF1	CSIIF0	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W

	7	6	⑤	④	③	②	①	⓪			
IF0H	0	0	WTIF Note	KSIF	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

xxIF	Interrupt Request Flag
0	No interrupt request signal
1	Interrupt request signal is generated: Interrupt request state

★ **Note** WTIF is test input flag. Vectored interrupt request is not generated.

Cautions 1. TMIF4 flag is R/W enabled only when a watchdog timer is used as an interval timer. If a watchdog timer will be used in watchdog timer mode 1, set TMIF4 flag to 0.

2. Be sure to clear bits 6 and 7 of IF0H to 0.

(2) Interrupt mask flag registers (MK0L, MK0H)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L or MK0H is set with a 1-bit or 8-bit memory manipulation instruction. If these two 8-bit registers are used as a single jointed 16-bit register MK0, use a 16-bit memory manipulation instruction to set it.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 17-3. Interrupt Mask Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
MK0L	TMMK3	CSIMK1	CSIMK0	PMK3	PMK2	PMK1	PMK0	TMMK4	FFE4H	FFH	R/W

	7	6	⑤	④	③	②	①	⓪			
MK0H	1	1	WTMK Note	KSMK	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

xxMK	Interrupt Serving and Standby Mode Control
0	Interrupt serving enabled, standby mode release enabled
1	Interrupt serving disabled, standby mode release disabled

★ **Note** WTMK controls standby mode release enable/disable. Does not control the interrupt function.

Cautions 1. If a watchdog timer is used in watchdog timer mode 1, the MK0 value becomes undefined when TMMK4 flag is read.

2. Because port 0 has an alternate function as the external interrupt request input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set.

Therefore, 1 should be set in the interrupt mask flag before using the output mode.

3. Be sure to set bits 6 and 7 of MK0H to 1.

(3) Priority specify flag registers (PR0L, PR0H)

The priority specify flag is used to set the corresponding maskable interrupt priority orders.

PR0L or PR0H is set with a 1-bit or 8-bit memory manipulation instruction. If these two 8-bit registers are used as a single jointed 16-bit register PR0, use a 16-bit memory manipulation instruction to set it.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 17-4. Priority Specify Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
PR0L	TMPR3	CSIPR1	CSIPR0	PPR3	PPR2	PPR1	PPR0	TMPR4	FFE8H	FFH	R/W
	7	6	5	④	③	②	①	⓪			
PR0H	1	1	1	KSPR	ADPR	TMPR2	TMPR1	TMPR0	FFE9H	FFH	R/W

xxPR	Priority Level Selection
0	High priority level
1	Low priority level

Cautions 1. When a watchdog timer is used in watchdog timer mode 1, set 1 in TMPR4 flag.

2. Be sure to set bits 5 to 7 of PR0H to 1.

(4) External interrupt mode registers (INTM0)

This register sets the valid edge for INTP0 to INTP2.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM0 value to 00H.

Remarks 1. INTP0 is also used for TI0/P00.

2. INTP3 is fixed at falling edge.

Figure 17-5. External Interrupt Mode Register Format

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

ES11	ES10	INTP0 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES21	ES20	INTP1 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES31	ES30	INTP2 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

Caution Before setting the valid edge of the INTP0/TI0/P00 pins, stop the timer operation by clearing bits 1 to 3 (TMC01 to TMC03) of the 16-bit timer mode control register (TMC0) to 0, 0, 0.

(5) Sampling clock select register (SCS)

This register is used to set the valid edge clock sampling clock to be input to INTP0. When remote controlled data reception is carried out using INTP0, digital noise is removed with sampling clocks.

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

$\overline{\text{RESET}}$ input clears SCS to 00H.

Figure 17-6. Sampling Clock Select Register Format

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

SCS1	SCS0	INTP0 Sampling Clock Selection
0	0	$f_X/2^{N+1}$
0	1	Setting prohibited
1	0	$f_X/2^6$ (78.1 kHz)
1	1	$f_X/2^7$ (39.1 kHz)

Caution: $f_X/2^{N+1}$ is a clock to be supplied to the CPU and $f_X/2^6$ and $f_X/2^7$ are clocks to be supplied to the peripheral hardware. $f_X/2^{N+1}$ stops in the HALT mode.

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

2. f_X : Main system clock frequency

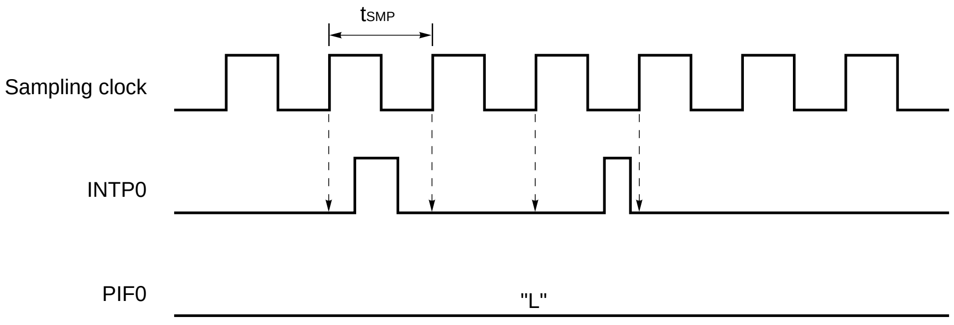
3. Values in parentheses when operated with fX = 5.0 MHz.

When the sampled INTP0 input level is active twice in succession, the noise eliminator sets the interrupt request flag (PIF0) to 1.

Figure 17-7 shows input/output timing of the noise eliminator.

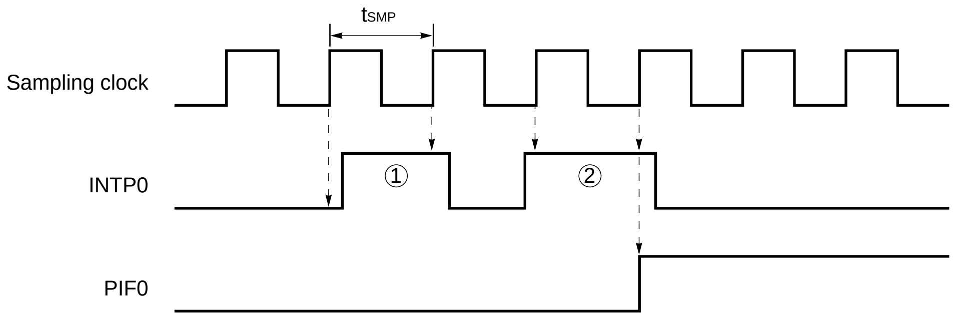
Figure 17-7. Noise Eliminator Input/Output Timing (when Rising Edge is Detected)

(a) When input is less than the sampling cycle (t_{SMP})



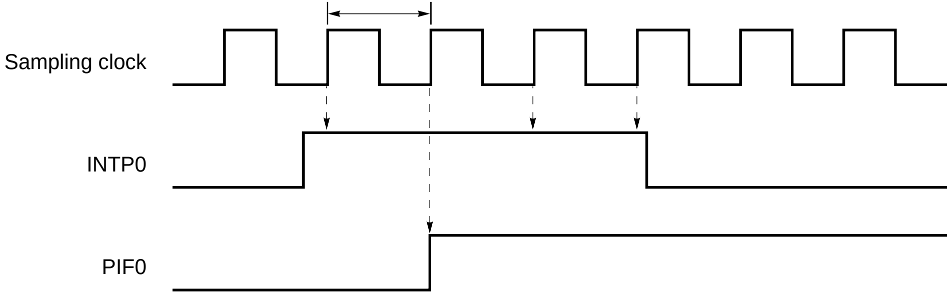
Because INTP0 level is not active (high) when it is sampled, PIF0 output remains at low level.

(b) When input is equal to or twice the sampling cycle (t_{SMP})



Because sampled INTP0 level is active (high) twice in succession in ②, PIF0 flag is set to 1.

(c) When input is twice or more than the sampling cycle (t_{SMP})



Because INTP0 level is active (high) twice in succession, PIF0 flag is set to 1.

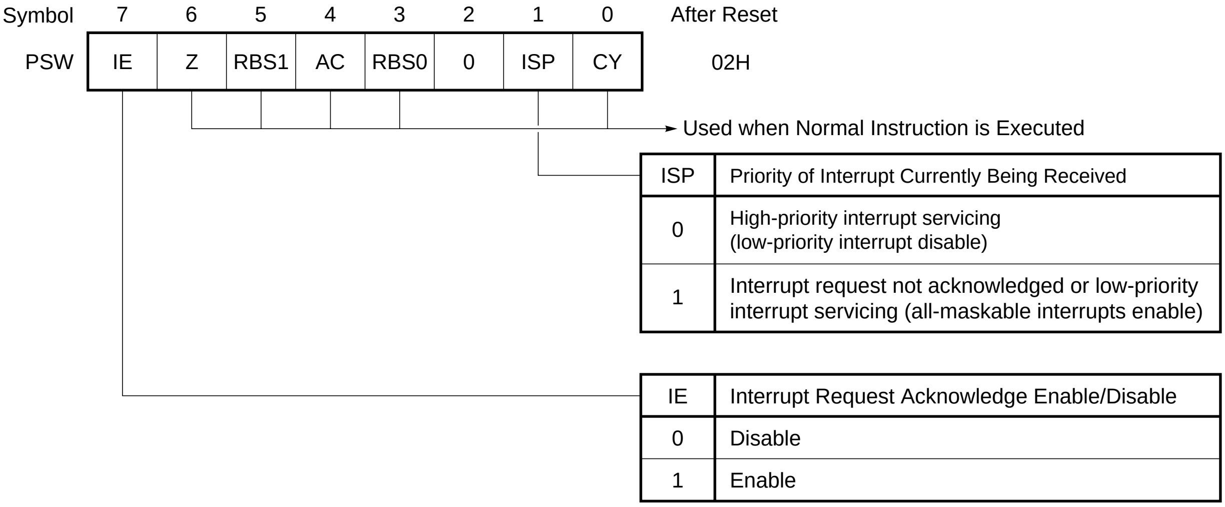
(6) Program status word (PSW)

The program status word is a register to hold the instruction execution result and the current status for interrupt request. The IE flag to set maskable interrupt enable/disable and the ISP flag to control multiple interrupt services are mapped.

Besides 8-bit unit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged or when a BRK instruction is executed, the contents of the program status word are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt is acknowledged, the contents of the priority specify flag of the acknowledged interrupt are transferred to the ISP flag. The contents of PSW can also be saved into the stack with the PUSH PSW instruction. It is reset from the stack with the RETI, RETB and POP PSW instructions.

$\overline{\text{RESET}}$ input sets PSW to 02H.

Figure 17-8. Program Status Word Format



ISP	Priority of Interrupt Currently Being Received
0	High-priority interrupt servicing (low-priority interrupt disable)
1	Interrupt request not acknowledged or low-priority interrupt servicing (all-maskable interrupts enable)

IE	Interrupt Request Acknowledge Enable/Disable
0	Disable
1	Enable

17.4 Interrupt Servicing Operations

17.4.1 Non-maskable interrupt request acknowledge operation

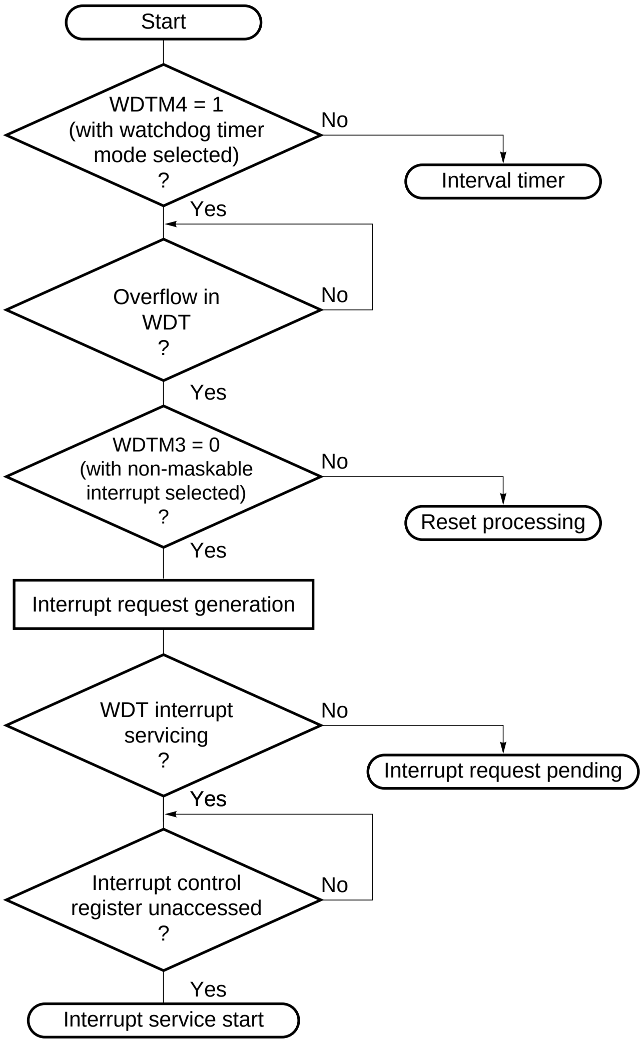
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the contents of the acknowledged interrupt are saved in the stacks, PSW and PC, in that order, the IE and ISP flags are reset to 0, and the vector table contents are loaded into PC and branched.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current execution of the non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction is executed. If a new non-maskable interrupt request is generated twice or more during non-maskable interrupt service program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt service program execution.

Figure 17-9 shows the flowchart illustrating how the non-maskable interrupt request occurs and is acknowledged. Figure 17-10 shows the acknowledge timing of the non-maskable interrupt request. Figure 17-11 shows acknowledge operation of multiple non-maskable interrupt requests.

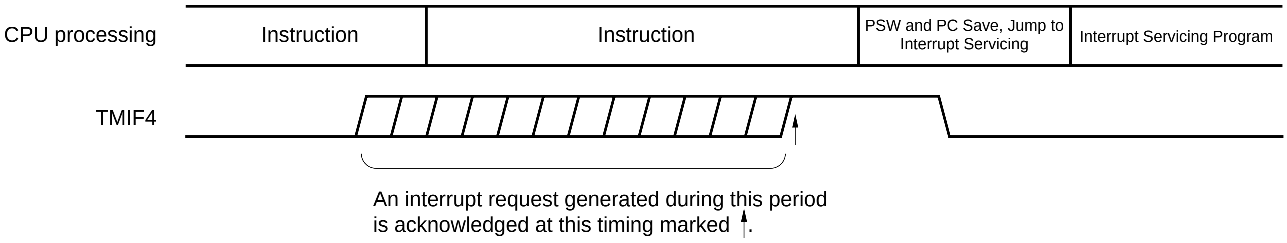
Figure 17-9. Non-Maskable Interrupt Request Occurrence and Acknowledge Flowchart



Remarks 1. WDTM : Watchdog timer mode register

2. WDT : Watchdog timer

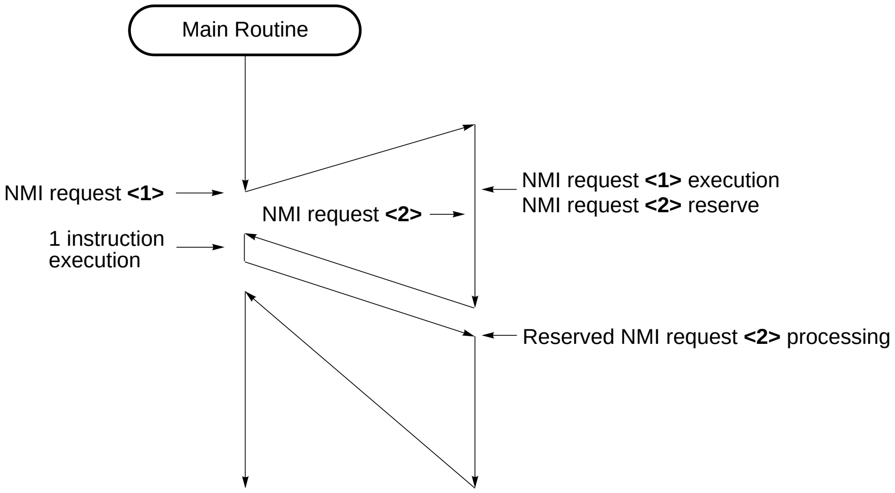
Figure 17-10. Non-Maskable Interrupt Request Acknowledge Timing


CPU processing
Instruction
Instruction
PSW and PC Save, Jump to Interrupt Servicing
Interrupt Servicing Program
TMIF4
An interrupt request generated during this period
is acknowledged at this timing marked ↑.

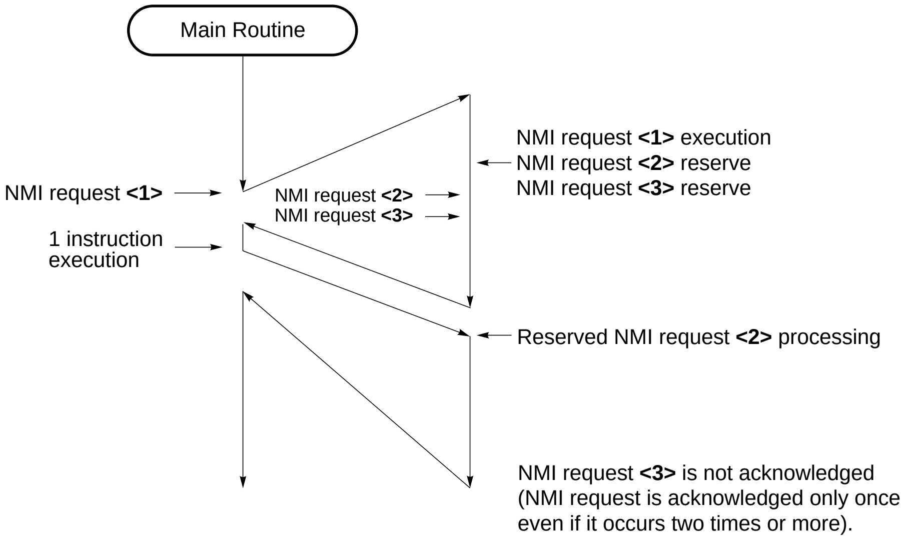
TMIF4 : Watchdog timer interrupt request flag

Figure 17-11. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a new non-maskable interrupt request is generated during non-maskable interrupt servicing program execution


Main Routine
NMI request **<1>**
1 instruction execution
NMI request **<2>**
NMI request **<1>** execution
NMI request **<2>** reserve
Reserved NMI request **<2>** processing

(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution


Main Routine
NMI request **<1>**
1 instruction execution
NMI request **<2>**
NMI request **<3>**
NMI request **<1>** execution
NMI request **<2>** reserve
NMI request **<3>** reserve
Reserved NMI request **<2>** processing
NMI request **<3>** is not acknowledged
(NMI request is acknowledged only once
even if it occurs two times or more).

17.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the interrupt request mask (MK) flag is cleared to 0. A vectored interrupt request is acknowledged in an interrupt enable state (with IE flag set to 1). However, a low-priority interrupt request is not acknowledged during high-priority interrupt service (with ISP flag reset to 0).

Wait times from maskable interrupt request generation to interrupt servicing are shown in Table 17-3.

For the timing of the acknowledge an interrupt request, refer to Figures 17-13 and 17-14.

Table 17-3. Times from Maskable Interrupt Request Generation to Interrupt Service

	Minimum Time	Maximum Time Note
When xxPR = 0	7 clocks	32 clocks
When xxPR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time is maximized.

Remark 1 clock : $\frac{1}{f_{CPU}}$ (where, f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specify flag is acknowledged first. If two or more requests are specified for the same priority, by the interrupt priority specify flag, the one with the higher default priority is acknowledged first.

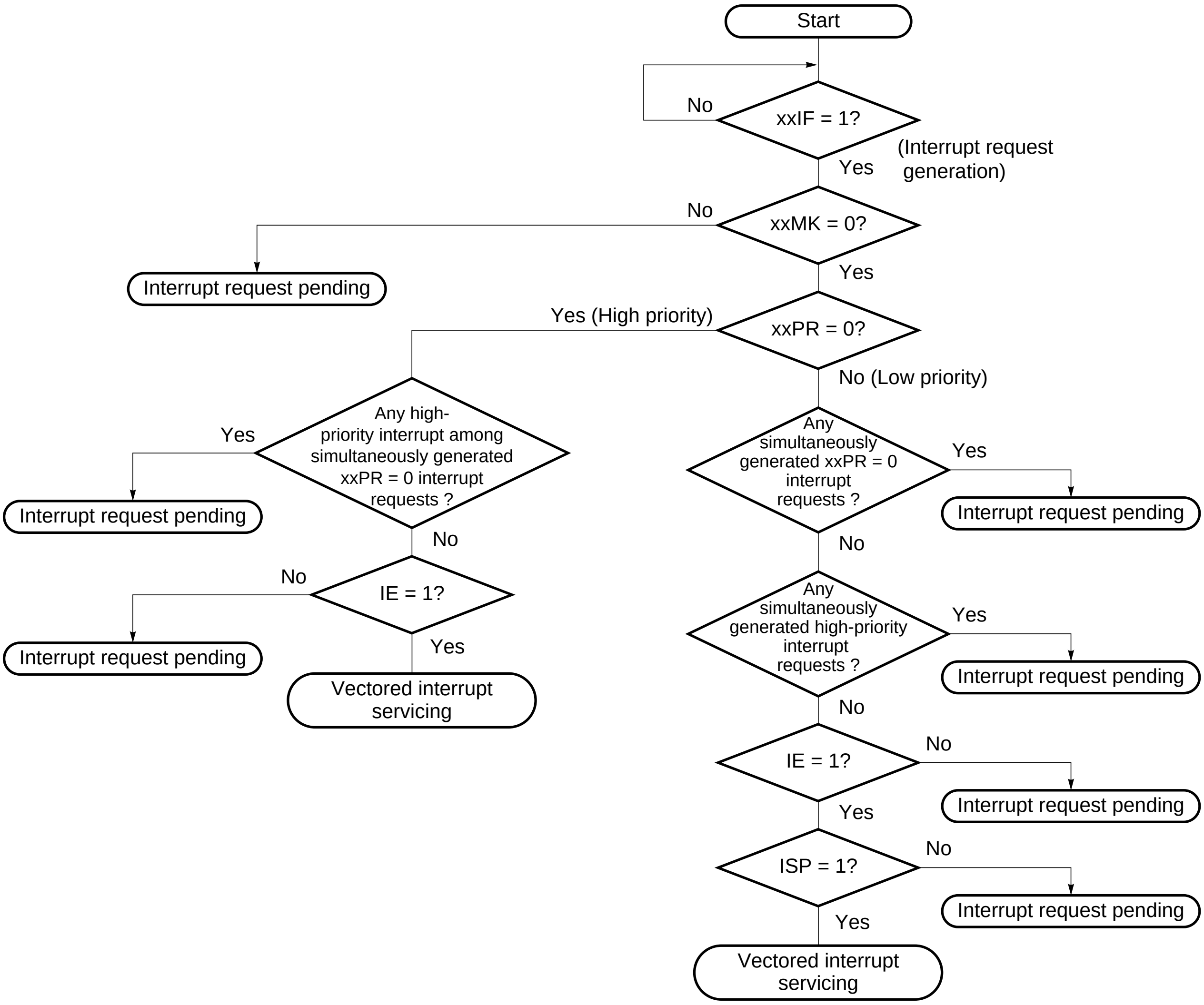
Any reserved interrupt requests are acknowledged when they become acknowledgeable.

Figure 17-12 shows interrupt request acknowledge algorithms.

If a maskable interrupt request is acknowledged, the contents of acknowledged interrupt are saved in the stacks, program status word (PSW) and program counter (PC), in that order, the IE flag is reset to 0, and the acknowledged interrupt priority specify flag contents are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

Return from the interrupt is possible with the RETI instruction.

Figure 17-12. Interrupt Request Acknowledge Processing Algorithm


Start
No
xxIF = 1?
(Interrupt request generation)
Yes
No
xxMK = 0?
Interrupt request pending
Yes
Yes (High priority)
xxPR = 0?
No (Low priority)
Yes
Any high-priority interrupt among simultaneously generated xxPR = 0 interrupt requests ?
Interrupt request pending
No
No
IE = 1?
Interrupt request pending
Yes
Vectored interrupt servicing
Any simultaneously generated xxPR = 0 interrupt requests ?
Yes
Interrupt request pending
No
Any simultaneously generated high-priority interrupt requests ?
Yes
Interrupt request pending
No
IE = 1?
No
Interrupt request pending
Yes
ISP = 1?
No
Interrupt request pending
Yes
Vectored interrupt servicing

xxIF: Interrupt request flag

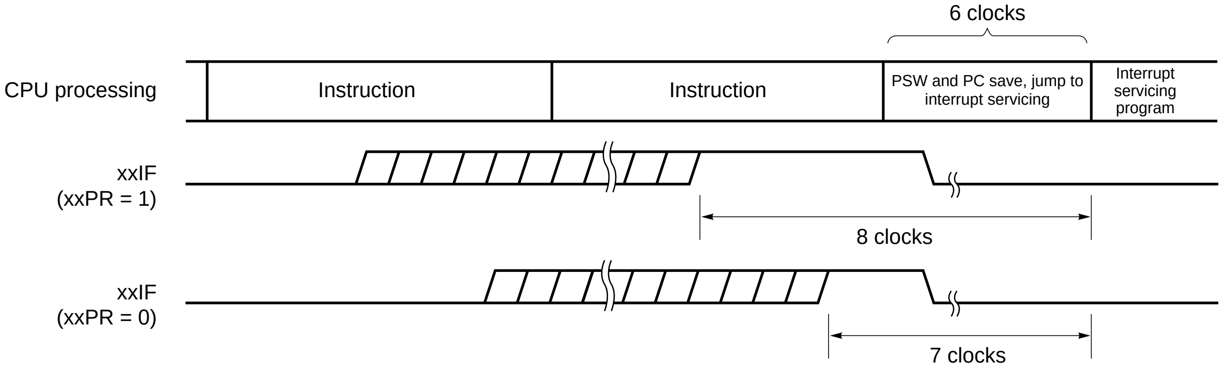
xxMK: Interrupt mask flag

xxPR: Priority specify flag

IE: Flag that controls maskable interrupt request acknowledge (1 = enable, 0 = disable)

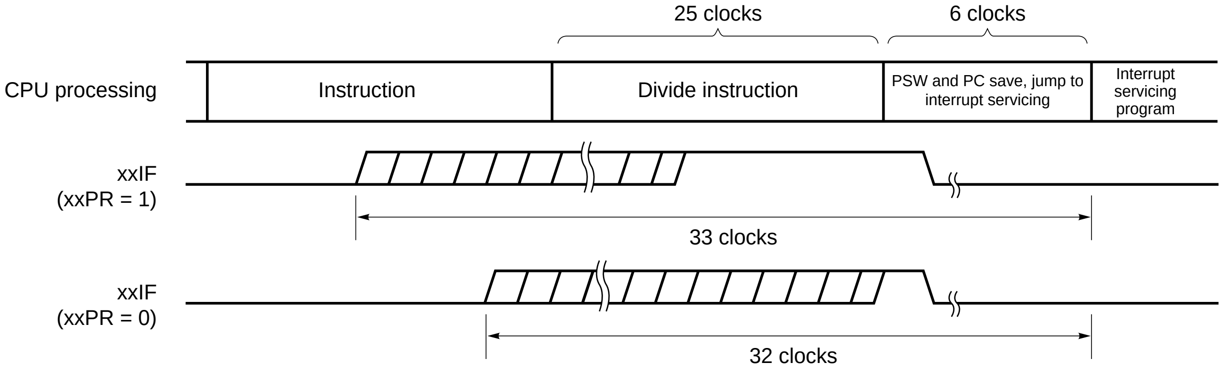
ISP: Flag indicating priority of interrupt currently processed (0 = interrupt with high priority is processed. 1 = interrupt request is not acknowledged or interrupt with low priority processed).

Figure 17-13. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (where, f_{CPU}: CPU clock)

Figure 17-14. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock : $\frac{1}{f_{CPU}}$ (where, f_{CPU}: CPU clock)

17.4.3 Software interrupt request acknowledge operation

A software interrupt request is acknowledged by BRK instruction execution. Software interrupt cannot be disabled.

If a software interrupt request is acknowledged, it is saved in the stacks, program status word (PSW) and program counter (PC), in that order, the IE flag is reset to 0 and the contents of the vector tables (003EH and 003FH) are loaded into PC and branched.

Return from the software interrupt is possible with the RETB instruction.

Caution Do not use the RETI instruction for returning from the software interrupt.

★

17.4.4 Multiple interrupt servicing

Acknowledging another interrupt request while an interrupt is processed is called multiple interrupts.

A multiple interrupt is not generated unless the interrupt request is enabled (IE = 1) (except the non-maskable interrupt). When an interrupt request is acknowledged, the other interrupts are disabled (IE = 0). To enable a multiple interrupt, therefore, the IE flag must be set to 1 by executing the EI instruction during interrupt processing and the interrupt must be enabled.

Even if the interrupt request is enabled, some multiple instructions are not acknowledged. However, the multiple instructions are controlled by the programmable priority. An interrupt has two types of priorities: default priority and programmable priority. The multiple interrupt is controlled by the programmable priority.

In the EI status, if an interrupt request having the same or higher priority than that of the interrupt currently processed is generated, the interrupt is acknowledged as multiple interrupt. If an interrupt request with a priority lower than that of the interrupt currently processed is generated, the interrupt is not acknowledged as multiple interrupt.

If an interrupt is disabled, or if a multiple interrupt is not acknowledged because it has a low priority, the interrupt is kept pending. After the processing of the current interrupt has been completed, and after one instruction of the main processing has been executed, the pending interrupt is acknowledged.

Multiple interrupts are not acknowledged while the non-maskable interrupt is processed.

Table 17-4 shows interrupt requests enabled for multiple interrupts. Figure 17-15 shows multiple interrupt examples.

Table 17-4. Interrupt Request Enabled for Multiple Interrupt during Interrupt Servicing

Multiple Interrupt Request		Non-Maskable Interrupt Request	Maskable Interrupt Request			
			xxPR = 0		xxPR = 1	
Servicing Interrupt			IE = 1	IE = 0	IE = 1	IE = 0
Non-maskable interrupt		D	D	D	D	D
Maskable interrupt	ISP = 0	E	E	D	D	D
	ISP = 1	E	E	D	E	D
Software interrupt		E	E	D	E	D

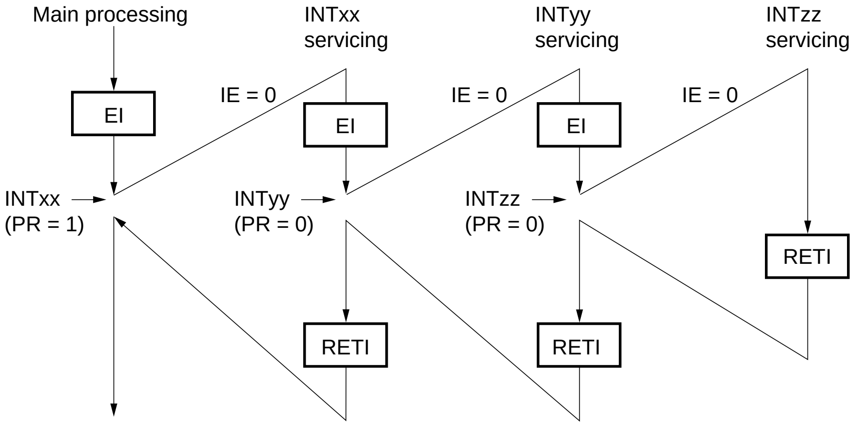
Remarks 1. E : Multiple interrupt enable
D : Multiple interrupt disable

2. ISP and IE are flags included in PSW.
ISP = 0 : High-priority interrupt servicing
ISP = 1 : Interrupt request is not acknowledged or low-priority interrupt servicing
IE = 0 : Interrupt request acknowledge disabled
IE = 1 : Interrupt request acknowledge enabled

3. xxPR is a flag included in PR0L, PR0H.
xxPR = 0: High-priority flag
xxPR = 1: Low-priority flag

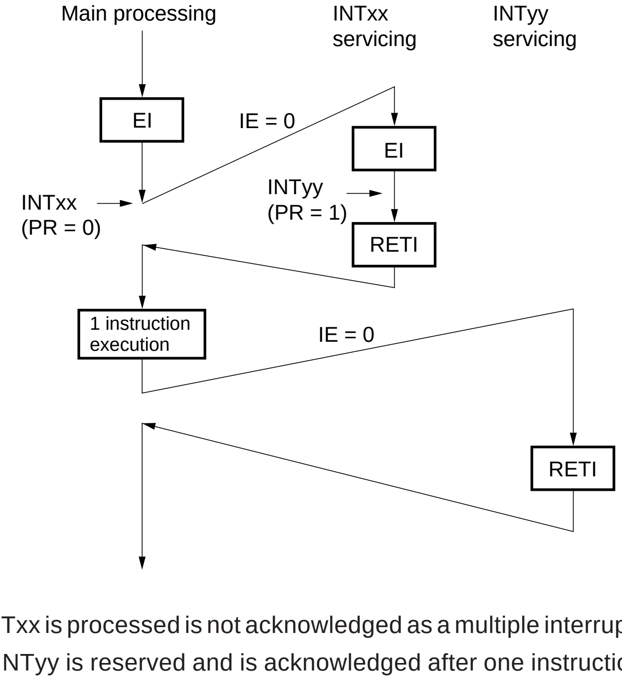
Figure 17-15. Multiple Interrupt Example (1/2)

Example 1. Two multiple interrupts are acknowledged.



Two multiple interrupt requests INTyy and INTzz are acknowledged while interrupt INTxx is processed. Before each interrupt request is acknowledged, the EI instruction is always issued and the interrupt request is enabled.

Example 2. Multiple interrupt is not acknowledged because of its priority.

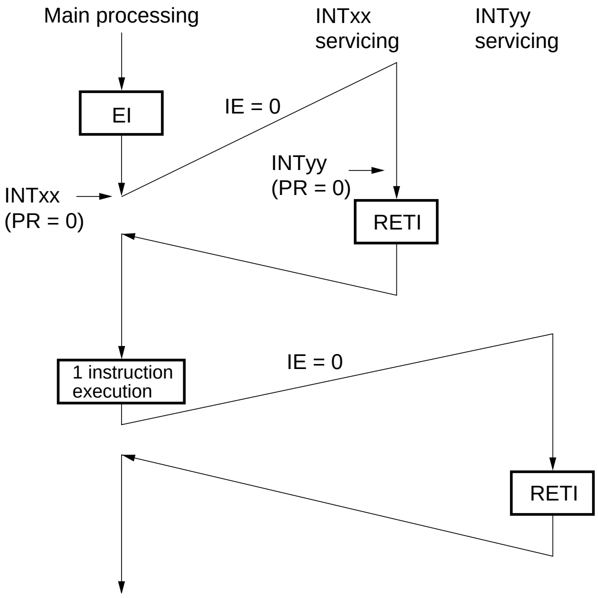


INTyy that occurs while INTxx is processed is not acknowledged as a multiple interrupt because the priority of INTyy is lower than that of INTxx. INTyy is reserved and is acknowledged after one instruction of the main processing has been executed.

PR = 0: High-priority interrupt
PR = 1: Low-priority interrupt
IE = 0: Interrupt acknowledge disabled

Figure 17-15. Multiple Interrupt Example (2/2)

Example 3. Multiple interrupt is not acknowledged because an interrupt is not enabled.



In the processing of INTxx, other interrupts are not enabled (the EI instruction is not executed). Therefore, INTyy is not acknowledged as a multiple interrupt. This interrupt is reserved and acknowledged after one instruction of the main processing has been executed.

PR = 0: High-priority interrupt
IE = 0: Interrupt acknowledge disabled

17.4.5 Interrupt request reserve

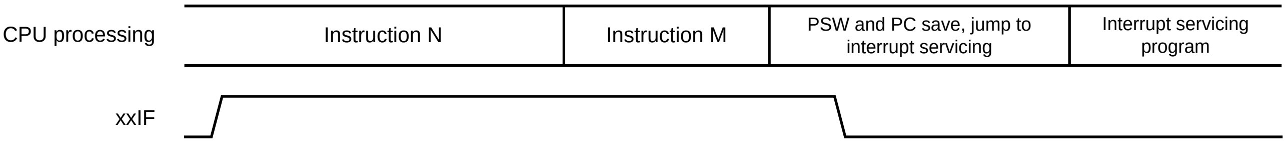
Even if an interrupt request is generated, some instructions reserve interrupt acknowledge while the current instruction is executed and until execution of the next instruction is completed. The instruction that reserve interrupt requests (interrupt request reserve) are shown below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulation instructions for IF0L, IF0H, MK0L, MK0H, PR0L, PR0H, and INTM0 registers

Caution The BRK instruction is not an interrupt request reserve instruction. However, the IE flag is cleared to 0 with a software interrupt that is started by BRK instruction execution. Thus, even if a maskable interrupt request is generated during BRK instruction, interrupt requests are not acknowledged. However, non-maskable interrupt requests are acknowledged.

Figure 17-16 shows the timing at which an interrupt request is reserved.

Figure 17-16. Interrupt Request Reserve


CPU processing
Instruction N
Instruction M
PSW and PC save, jump to interrupt servicing
Interrupt servicing program
xxIF

Remarks 1. Instruction N: Interrupt request reserve instruction

2. Instruction M: Instruction except interrupt request reserve instructions

3. Operation of xxIF (interrupt request) is not affected by xxPR (priority level) value.

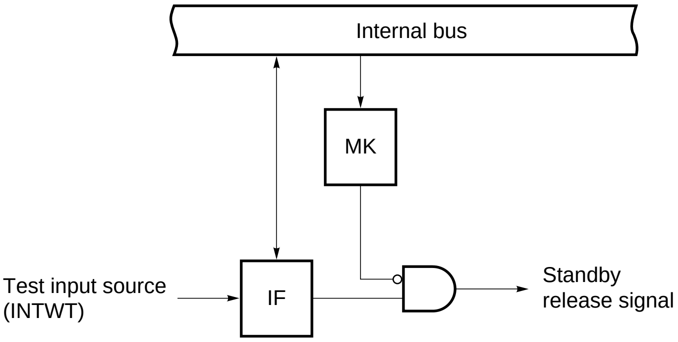
17.5 Test Function

When the watch timer overflows, an internal test input flag (INTWT) is set to 1, and the standby release signal is generated.

Unlike the interrupt function, vectored processing is not performed.

The basic configuration is shown in Figure 17-17.

Figure 17-17. Basic Configuration of Test Function



Remarks:
(1) IF : Test input flag
(2) MK : Test mask flag

17.5.1 Test function control registers

The following two types of registers are used to control the test functions.

- Interrupt request flag register 0H (IF0H)
- Interrupt mask flag register 0H (MK0H)

Names of test input flag and test mask flag corresponding to test input signal name are as follows.

Test Input Signal Name	Test Input Flag	Test Mask Flag
INTWT	WTIF	WTMK

(1) Interrupt request flag register (IF0H)

This register displays the watch timer overflow detection/non-detection.

IF0H is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IF0H at 00H.

Figure 17-18. Interrupt Request Flag Register 0H Format

Symbol	7	6	⑤	④	③	②	①	⓪	Address	After Reset	R/W
IF0H	0	0	WTIF	KSIF	ADIF	TMIF2	TMIF1	TMIF0	FFE1H	00H	R/W

WTIF	Watch Timer Overflow Detection Flag
0	Non detection
1	Detection

(2) Interrupt mask flag register (MK0H)

This register sets standby mode release enable/disable by watch timer.

MK0H is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0H to FFH.

Figure 17-19. Interrupt Mask Flag Register 0H Format

Symbol	7	6	⑤	④	③	②	①	⓪	Address	After Reset	R/W
MK0H	1	1	WTMK	KSMK	ADMK	TMMK2	TMMK1	TMMK0	FFE5H	FFH	R/W

WTMK	Standby Made Control by Watch Timer
0	Standby mode release enable
1	Standby mode release disable

17.5.2 Test input signal acknowledge operation

The internal test input signal (INTWT) is generated when the watch timer overflows, and sets the WTIF flag. Unless interrupts are masked by the interrupt mask flag (WTMK) at this time, the standby release signal is generated. The watch function is possible by checking the INTWT flag at a shorter cycle than the watch timer overflow cycle.

[MEMO]

CHAPTER 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 Standby function

The standby function is intended to decrease power consumption of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. The HALT mode is intended to stop the CPU operation clock. System clock oscillator continues oscillation. In this mode, current consumption cannot be decreased as in the STOP mode. The HALT mode is valid to restart immediately upon interrupt request and to carry out intermittent operations like clock operations.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the main system clock oscillator stops and the whole system stops. CPU current consumption can be considerably decreased.

Data memory low-voltage hold (down to VDD = 2 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption. Because this mode can be released by an interrupt request, it enables intermittent operations to be carried out.

However, because a wait time is necessary to secure an oscillation stabilization time after the STOP mode is released, select the HALT mode if it is necessary to start processing immediately by using an interrupt request.

In any modes, all the contents of the register, flag and data memory just before standby mode setting are held. The input/output port output latch and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the system operates with the main system clock (subsystem clock oscillation cannot be stopped). The HALT mode can be used with either the main system clock or the subsystem clock.

2. When proceeding to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

3. In consideration of power consumption reduction at the A/D converter, set bit 7 (CS) of the A/D converter mode register (ADM) register to 0 to stop the A/D converter's operation before executing the HALT or STOP instruction.

18.1.2 Standby function control register

A wait time after the STOP mode is released by an interrupt request till the oscillation stabilizes is controlled with the oscillation stabilization time select register (OSTS).

OSTS is set with an 8-bit memory manipulation instruction.

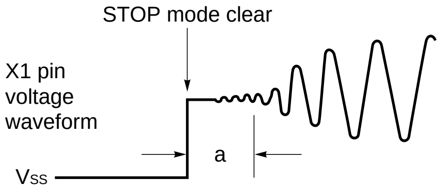
$\overline{\text{RESET}}$ input sets OSTS to 04H. Therefore, the STOP mode is released by $\overline{\text{RESET}}$ input, the time till it is released is 2^{17}/fx.

Figure 18-1. Oscillation Stabilization Time Select Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Selection of Oscillation Stabilization Time when STOP Mode is Released
0	0	0	2^{12}/fx (819 μs)
0	0	1	2^{14}/fx (3.28 ms)
0	1	0	2^{15}/fx (6.55 ms)
0	1	1	2^{16}/fx (13.1 ms)
1	0	0	2^{17}/fx (26.2 ms)
Other than above			Setting prohibited

Caution **The wait time after STOP mode is released does not include the time (see “a” below) from STOP mode clear to clock oscillation start, regardless of release by $\overline{\text{RESET}}$ input or by interrupt request generation.**



Remarks **1.** fX: Main system clock frequency

2. Values in parentheses apply to operating at fX = 5.0 MHz

18.2 Standby Function Operations

18.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction. It can be set with the main system clock or the subsystem clock.

The operating status in the HALT mode is described below.

Table 18-1. HALT Mode Operating Status

HALT mode Setting Item		When HALT Instruction is Executed During Main System Clock Oscillation		When HALT Instruction is Executed During Subsystem Clock Oscillation	
		Without Subsystem Clock Note 1	With Subsystem Clock Note 2	When Main System Clock Oscillation Continues	When Main System Clock Oscillation Stops
Clock generator		Both main system clock and subsystem clock can be oscillated. Clock supply to the CPU stops.			
CPU		Operation stop.			
Port (output latch)		Status before HALT instruction execution is held.			
16-bit timer/event counter		Operation enabled.			Operation stop.
8-bit timer/event counter					Operation enabled when TI1 and TI2 are selected for the count clock.
Watchdog timer					Operation stop.
A/D converter					Operation stop.
Watch timer		Operation enabled when $f_X/2^8$ is selected for the count clock.	Operation enabled.		Operation enabled when f_{XT} is selected for the count clock.
6-bit up/down counter		Operation enabled.			
FIP controller/driver		Operation disabled.			
Serial interface	Other than automatic transmit/receive function	Operation enabled.			Operation enabled when external SCK is selected.
	Automatic transmit/receive function	Operation stop.			
External interrupt	INTP0	Operation enabled when the clock ($f_X/2^6$ and $f_X/2^7$) for the peripheral hardware are selected as sampling clock.			Operation stop.
	INTP1 to INTP3	Operation enabled.			

Notes 1. Including the case where an external clock is not supplied as the subsystem clock

2. Including the case where an external clock is supplied as the subsystem clock

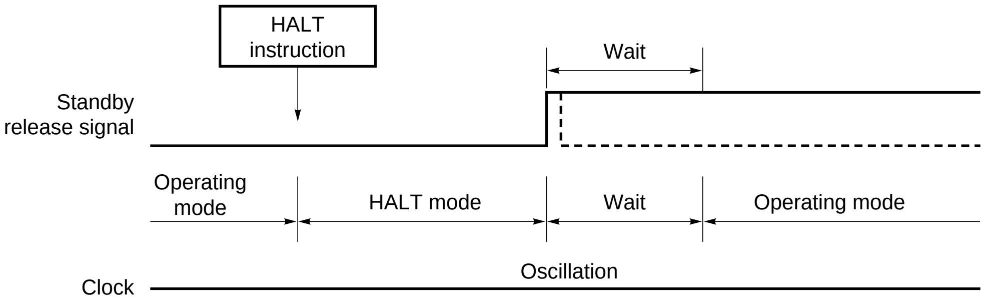
(2) Releasing HALT mode

The HALT mode can be released by the following four types of sources.

(a) Releasing by unmasked interrupt request

An unmasked interrupt request is used to release the HALT mode. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If disabled, the next address instruction is executed.

Figure 18-2. Releasing HALT Mode by Interrupt Request Generation



Remarks 1. The broken line indicates the case when the interrupt request which has released the standby status is acknowledged.

2. Wait time will be as follows:

- When vectored interrupt service is carried out : 8 to 9 clocks
- When vectored interrupt service is not carried out : 2 to 3 clocks

(b) Releasing by non-maskable interrupt request

The HALT mode is released and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

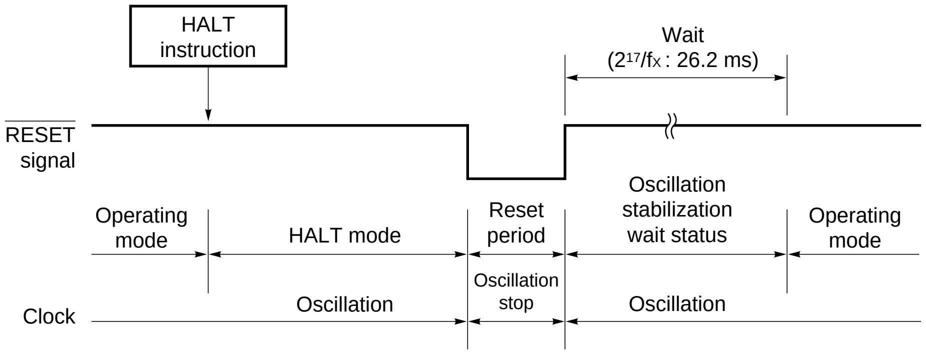
(c) Releasing by unmasked test input

The HALT mode is released by unmasked test input and the next address instruction of the HALT instruction is executed.

(d) Releasing by $\overline{\text{RESET}}$ input

As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 18-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remarks 1. fx : Main system clock oscillation frequency

2. The value in parentheses applies to operating at fx = 5.0 MHz.

Table 18-2. Operation after HALT Mode Release

Releasing Source	MKxx	PRxx	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	HALT mode hold
Non-maskable interrupt request	–	–	×	×	Interrupt service execution
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	HALT mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark × : don't care

18.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction. It can be set only with the main system clock.

★ **Cautions 1. When the STOP mode is set, the X2 pin is internally pulled up to V_{DD} to suppress the current leakage of the crystal oscillation circuit block.**

Thus, do not use the STOP mode in a system where an external clock is used for the main system clock.

2. Because the interrupt request signal is used to release the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately released if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction. After the wait set using the oscillation stabilization time select register (OSTS), the operating mode is set.

The operating status in the STOP mode is described below.

Table 18-3. STOP Mode Operating Status

Stop Made Setting / Item		With Subsystem Clock	Without Subsystem Clock
Clock generator		Only main system clock stops oscillation.	
CPU		Operation stop.	
Output port (output latch)		Status before STOP mode setting is held.	
16-bit timer/event counter		Operation stop.	
8-bit timer/event counter		Operation enabled only when TI1 and TI2 are selected for the count clock.	
Watchdog timer		Operation stop.	
A/D converter			
Watch timer		Operation enabled only when fXT is selected for the count clock.	Operation stop.
6-bit up/down counter		Operation enabled.	
FIP controller/driver		Operation disabled.	
Serial interface	Other than automatic transmit/receive function	Operation enabled only when external input clock is selected as serial clock.	
	Automatic transmit/ receive function	Operation stop.	
External interrupt	INTP0	Operation disabled.	
	INTP1 to INTP3	Operation enabled.	

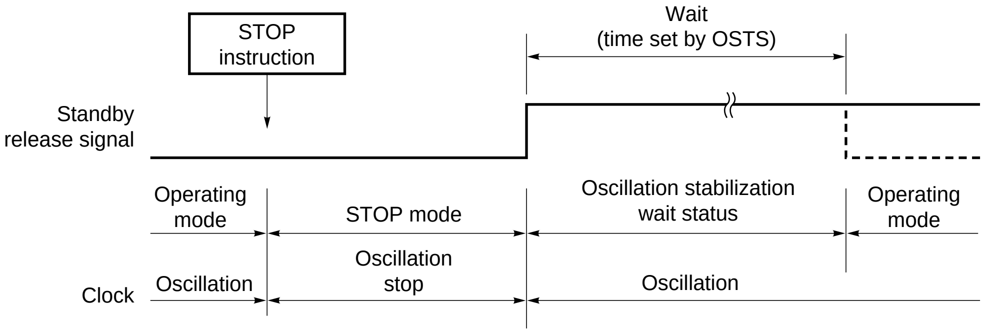
(2) Releasing STOP mode

The STOP mode can be released by the following three types of sources.

(a) Releasing by unmasked interrupt request

An unmasked interrupt request is used to release the STOP mode. If interrupt acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt acknowledge is disabled, the next address instruction is executed.

Figure 18-4. Releasing STOP Mode by Interrupt Request Generation



Remark The broken line indicates the case when the interrupt request which has released the standby status is acknowledged.

(b) Releasing by unmasked test input

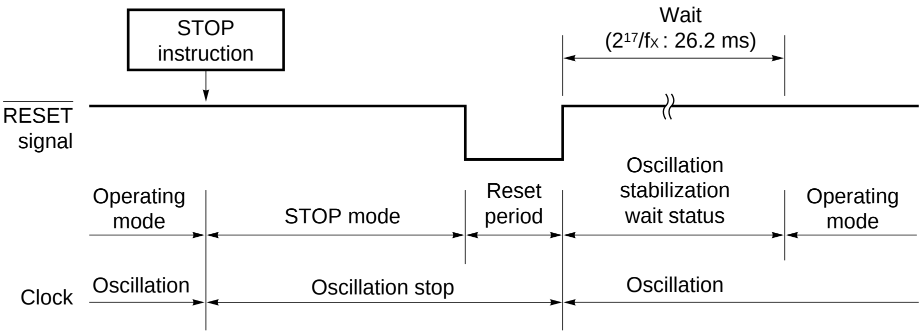
The STOP mode is released by unmasked test input.

After the lapse of oscillation stabilization time, the instruction at the next address of the STOP instruction is executed.

(c) Releasing by $\overline{\text{RESET}}$ input

The STOP mode is released and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 18-5. Release by STOP Mode $\overline{\text{RESET}}$ Input



Remarks 1. f_X : Main system clock oscillation frequency

2. The value in parentheses applies to operating at f_X = 5.0 MHz.

Table 18-4. Operation after STOP Mode Release

Releasing Source	MKxx	PRxx	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt service execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt service execution
	1	×	×	×	STOP mode hold
Test input	0	–	×	×	Next address instruction execution
	1	–	×	×	STOP mode hold
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

Remark × : don’t care

CHAPTER 19 RESET FUNCTION

19.1 Reset Function

To generate a reset signal, the following two operations are available.

(1) External reset input with $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer overrun time detection

External reset and internal reset have no functional differences. In both cases, program execution starts at addresses 0000H and 0001H by $\overline{\text{RESET}}$ input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status as shown in Table 19-1. Each pin has high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$). The reset applied by watchdog timer overflow is automatically cleared after a reset and program execution starts after the lapse of oscillation stabilization time ($2^{17}/f_X$) (see **Figures 19-2** to **19-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. During reset input, main system clock oscillation remains stopped but subsystem clock oscillation continues.

3. When the STOP mode is cleared by reset, the STOP mode contents are held during reset input. However, the port pin becomes high-impedance.

Figure 19-1. Block Diagram of Reset Function

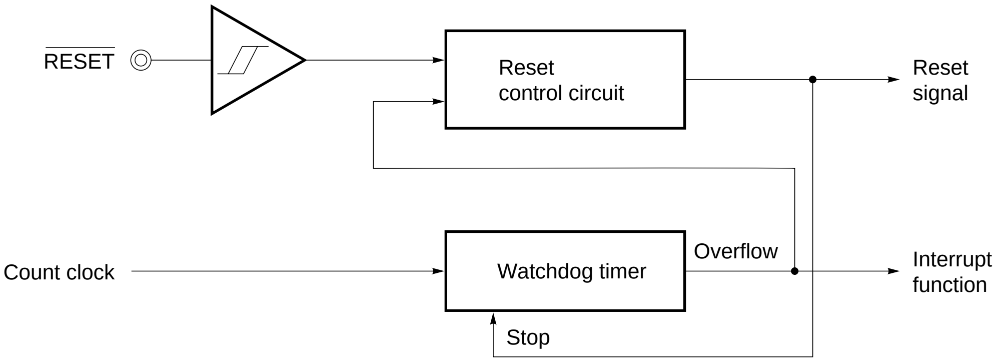


Figure 19-2. Timing of Reset Input by $\overline{\text{RESET}}$ Input

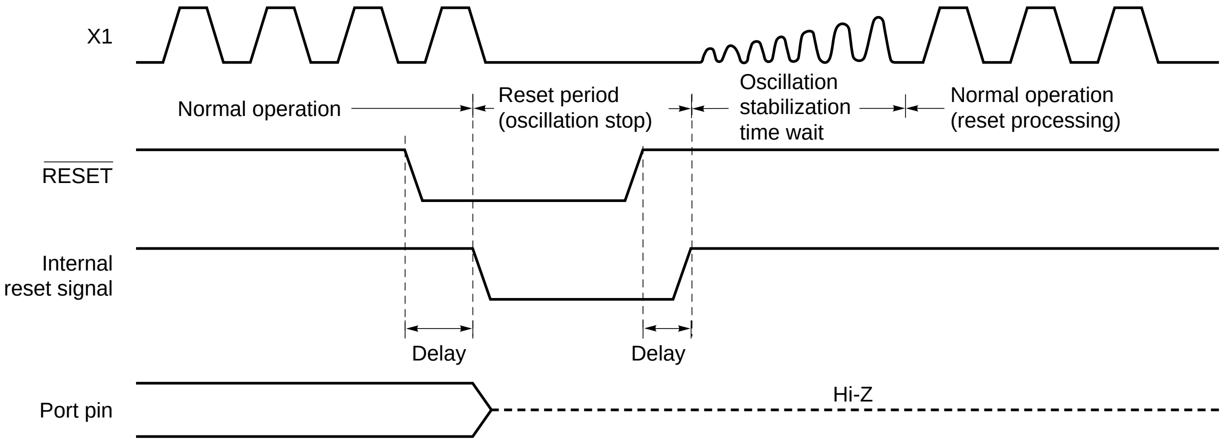


Figure 19-3. Timing of Reset due to Watchdog Timer Overflow

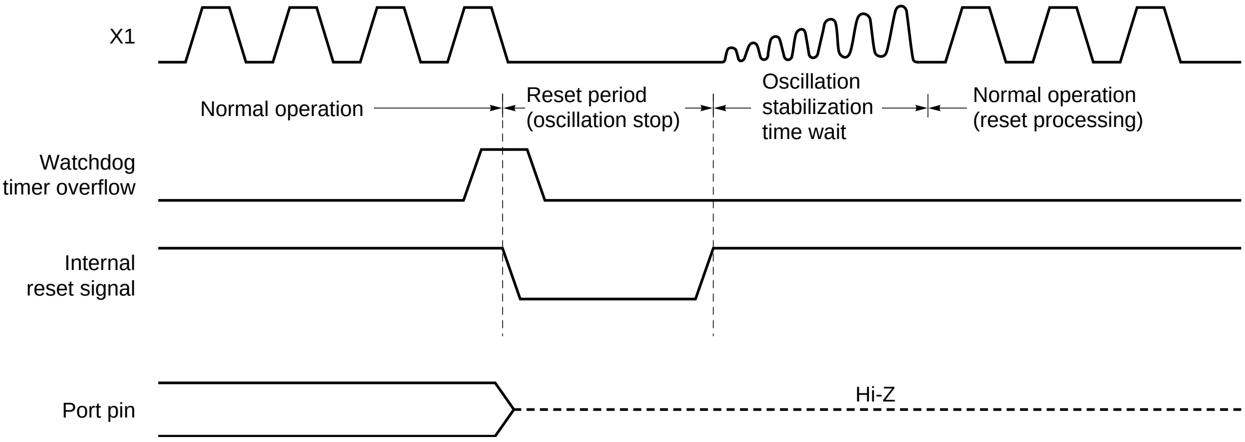


Figure 19-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input

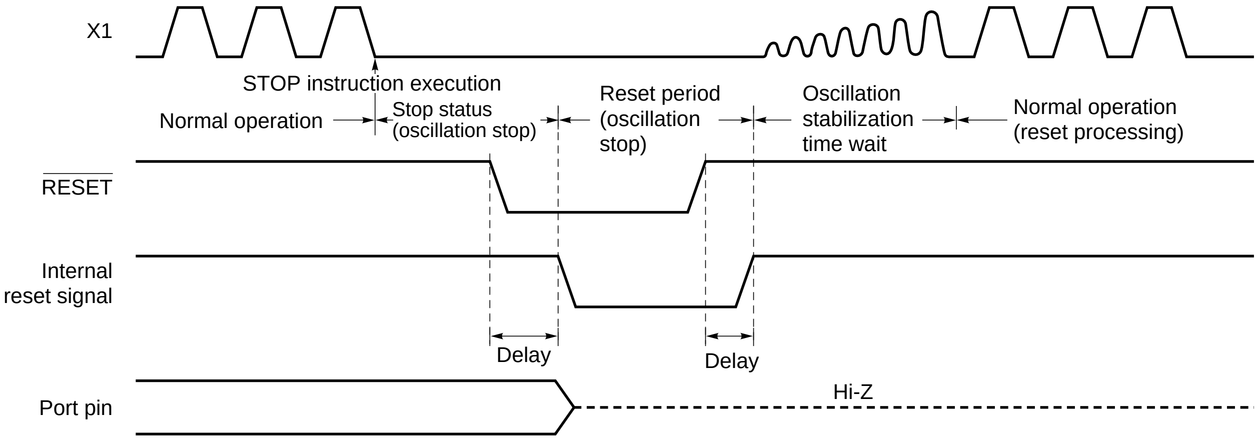


Table 19-1. Hardware Status after Reset (1/2)

Hardware		Status after Reset
Program counter (PC) Note 1		The contents of reset vector tables (0000H and 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined Note 2
	General register	Undefined Note 2
Port (Output latch)	Ports 0 to 3, Ports 7 to 12 (P0 to P3, P7 to P12)	00H
Port mode register	(PM0, PM7)	1FH
	(PM1, PM2, PM3, PM11, PM12)	FFH
Pull-up resistor option register (PUO)		00H
Processor clock control register (PCC)		04H
Memory size switching register (IMS)		**Note 3**
Internal expansion RAM size switching register (IXS)		**Note 3**
Oscillation stabilization time select register (OSTS)		04H
16-bit timer/event counter	Timer register (TM0)	00H
	Compare register (CR00)	Undefined
	Capture register (CR01)	Undefined
	Clock selection register (TCL0)	00H
	Mode control register (TMC0)	00H
	Output control register (TOC0)	00H
8-bit timer/event counter	Timer registers (TM1, TM2)	00H
	Compare registers (CR10, CR20)	Undefined
	Clock select register (TCL1)	00H
	Mode control registers (TMC1, TMC2)	00H
	Output control register (TOC1)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. The post-reset status is held in the standby mode.

3. The after-reset values of the memory size switching register (IMS) and internal expansion RAM size switching register (IXS) depend on products.

	μPD78042F	μPD78043F	μPD78044F	μPD78045F	μPD78P048A
IMS	44H	46H	C8H	CAH	CFH
IXS	None				0AH

Table 19-1. Hardware Status after Reset (2/2)

Hardware		Status after Reset
Watch timer	Clock select register (TCL2)	00H
Watchdog timer		
	Mode register (WDTM)	00H
6-bit up/down counter	Counter (UDC)	00H
	Compare register (UDCC)	00H
	Mode register (UDM)	00H
Serial interface	Clock select register (TCL3)	88H
	Shift registers (SIO0, SIO1)	Undefined
	Mode registers (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 address pointer (ADTP)	00H
	Automatic data transmit/receive interval specify register (ADTI)	00H
	Interrupt timing specify register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
FIP controller/driver	Display mode register 0 (DSPM0)	00H
	Display mode register 1 (DSPM1)	00H
Interrupt	Request flag register (IF0L, IF0H)	00H
	Mask flag register (MK0L, MK0H)	FFH
	Priority specify flag register (PR0L, PR0H)	FFH
	External interrupt mode register (INTM0)	00H
	Sampling clock select register (SCS)	00H

CHAPTER 20 μPD78P048A

The μPD78P048A is a product integrating a one-time programmable ROM or with an on-chip EPROM that enables program writing, erasure, and rewriting. The μPD78P048A is a μPD78044A Subseries device but is also compatible with μPD78044F subseries devices. Refer to Table 20-1 to check the differences between the μPD78P048A and the mask ROM versions (μPD78042F, 78043F, 78044F, and 78045F).

Table 20-1. Difference between μPD78P048A and Mask ROM Versions

Item	μPD78P048A	Mask ROM Versions
Supply voltage	V_{DD} = 2.7 to 6.0 V	V_{DD} = 2.7 to 5.5 V
Internal ROM configuration	One-time PROM/EPROM	Mask ROM
Internal ROM capacity	60 Kbytes	μPD78042F: 16 Kbytes μPD78043F: 24 Kbytes μPD78044F: 32 Kbytes μPD78045F: 40 Kbytes
Internal high-speed RAM capacity	1024 Kbytes	μPD78042F: 512 Kbytes μPD78043F: 512 Kbytes μPD78044F: 1024 Kbytes μPD78045F: 1024 Kbytes
Change in capacity of internal ROM and high-speed RAM by means of memory switching register (IMS)	Possible[Note1]	Impossible
Internal expansion RAM size switching register (IXS)	Available (internal expansion RAM capacity can be changed [Note 2] by means of IXS)	This register is not provided.
IC pin	None	Available
V_{PP} pin	Available	None
P80, P81, P90 to P97, P100 to P105	Pull-down resistors are not provided on chip.	On-chip mask option pull-down resistors are incorporated.
P30 to P37, P106, P107, P110 to P117, P120 to P127	Pull-down resistors are not provided on chip.	
P70 to P74	Pull-up resistors are not provided on chip.	On-chip mask option pull-up resistors are incorporated.
Electrical specifications	Read the separate data sheet.	

Notes 1. The internal PROM capacity is set to 60 Kbytes and the internal high-speed RAM capacity is set to 1024 bytes by $\overline{\text{RESET}}$ input.

2. The internal expansion RAM capacity is set to 1024 bytes by $\overline{\text{RESET}}$ input.

★ **Caution The PROM version and mask ROM version differ in noise immunity. When replacing a PROM version with a mask ROM version in the course of experimental production to mass production, make a thorough evaluation with a CS version (not ES version) of the mask ROM version.**

20.1 Internal Memory Size Switching Register

The μPD78P048A can select the internal memory capacity with the internal memory size switching register (IMS). The same memory map as that of the mask ROM version with a different internal memory capacity is possible by setting IMS.

In order to make the memory map of μPD78P048A identical to a mask ROM version, the value at the time the mask ROM version is reset must be set to IMS.

For the mask ROM version, IMS does not need to be set.

IMS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to the value shown in Table 20-2.

Figure 20-1. Memory Size Switching Register Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	FFF0H	**Note**	R/W

ROM3	ROM2	ROM1	ROM0	Internal ROM Capacity Selection
0	1	0	0	16 Kbytes
0	1	1	0	24 Kbytes
1	0	0	0	32 Kbytes
1	0	1	0	40 Kbytes
1	1	1	1	60 Kbytes
Other than above				Setting prohibited

RAM2	RAM1	RAM0	Internal High-Speed RAM Capacity Selection
0	1	0	512 bytes
1	1	0	1024 bytes
Other than above			Setting prohibited

Note The value of the memory size select register at reset differs depending on the model (refer to Table 20-2).

Table 20-2. Memory Size Switching Register Value at Reset

Part Number	Value at Reset
μPD78042F	44H
μPD78043F	46H
μPD78044F	C8H
μPD78045F	CAH
μPD78P048A	CFH

Caution To use a mask ROM version, do not set a value other than those shown in Table above to IMS.

20.2 Internal Expansion RAM Size Switching (IXS) Register

By setting the IXS register, the μPD78P048A can establish the same memory map as used in mask ROM version that have different internal expansion RAM capacity configurations.

For the mask ROM version, the IXS register does not need to be set.

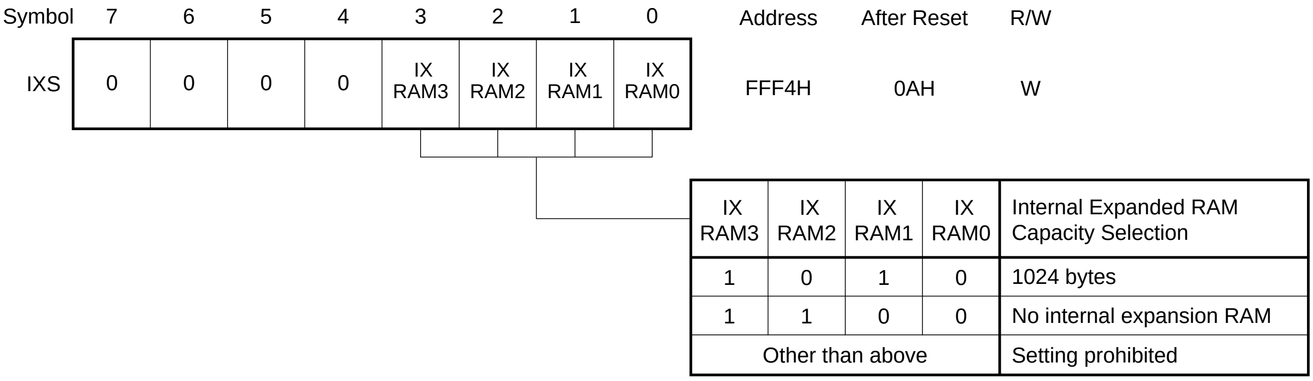
IXS is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets a value of 0AH.

Cautions 1. The IXS register is incorporated in only the μPD78P048A.

2. When using a mask ROM version, do not set a value other than those listed in Table 20-3 to the IXS.

Figure 20-2. Format of IXS Register



Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
IXS	0	0	0	0	IX RAM3	IX RAM2	IX RAM1	IX RAM0	FFF4H	0AH	W

IX RAM3	IX RAM2	IX RAM1	IX RAM0	Internal Expanded RAM Capacity Selection
1	0	1	0	1024 bytes
1	1	0	0	No internal expansion RAM
Other than above				Setting prohibited

Table 20-3 lists IXS register setting values for memory map equivalent to the mask ROM versions.

Table 20-3. IXS Register Settings

Target mask ROM version	IXS setting value
μPD78042F	0CH
μPD78043F	
μPD78044F	
μPD78045F	

The IXS register is not incorporated in the μPD78042F, μPD78043F, μPD78044F and μPD78045F.

However, if a write instruction to this register is executed in the μPD78042F, μPD78043F, μPD78044F and μPD78045F, the operations are not affected.

20.3 PROM Programming

The μPD78P048A incorporates a 60-Kbyte PROM as program memory. When programming the μPD78P048A, the PROM programming mode is set by means of the V_{PP} pin and the $\overline{\text{RESET}}$ pin. For the connection of unused pins, see section **1.5 "Pin Configuration, (2) PROM programming mode**."

Caution Programs must be written in addresses 0000H to EFFFH. (The last address EFFFH must be specified.) They cannot be written by a PROM programmer which cannot specify the write address.

20.3.1 Operating modes

When +5 V or +12.5 V is applied to the V_{PP} pin and a low-level signal is applied to the $\overline{\text{RESET}}$ pin, the μPD78P048A is set to the PROM programming mode. This is one of the operating modes shown in Table 20-4 below according to the setting of the $\overline{\text{CE}}$, $\overline{\text{OE}}$, and $\overline{\text{PGM}}$ pins.

The PROM contents can be read by setting the read mode.

Table 20-4. PROM Programming Operating Modes

Pin / Operating mode	$\overline{\text{RESET}}$	V_{PP}	V_{DD}	$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{PGM}}$	D0 to D7
Page data latch	L	+12.5 V	+6.5 V	H	L	H	Data input
Page write				H	H	L	High-impedance
Byte write				L	H	L	Data input
Program verify				L	L	H	Data output
Program inhibit				×	H	H	High-impedance
				×	L	L	
Read		+5 V	+5 V	L	L	H	Data output
Output disabled				L	H	×	High-impedance
Standby				H	×	×	High-impedance

× : L or H

(1) Read mode

Read mode is set by setting $\overline{\text{CE}}$ to L and $\overline{\text{OE}}$ to L.

(2) Output disable mode

If $\overline{\text{OE}}$ is set to H, data output becomes high impedance and the output disable mode is set.

Therefore, if multiple μPD78P048As are connected to the data bus, data can be read from any one device by controlling the $\overline{\text{OE}}$ pin.

(3) Standby mode

Setting $\overline{\text{CE}}$ to H sets the standby mode.

In this mode, data output becomes high impedance irrespective of the status of $\overline{\text{OE}}$.

(4) Page data latch mode

Setting $\overline{\text{CE}}$ to H, $\overline{\text{PGM}}$ to H, and $\overline{\text{OE}}$ to L at the start of the page write mode sets the page data latch mode. In this mode, 1-page 4-byte data is latched in the internal address/data latch circuit.

(5) Page write mode

After a 1-page 4-byte address and data are latched by the page data latch mode, a page write is executed by applying a 0.1-ms program pulse (active-low) to the $\overline{\text{PGM}}$ pin while $\overline{\text{CE}}$ = H and $\overline{\text{OE}}$ = H. After this, program verification can be performed by setting $\overline{\text{CE}}$ to L and $\overline{\text{OE}}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times ($X \leq 10$).

(6) Byte write mode

A byte write is executed by applying a 0.1-ms program pulse (active-low) to the $\overline{\text{PGM}}$ pin while $\overline{\text{CE}}$ = L and $\overline{\text{OE}}$ = H. After this, program verification can be performed by setting $\overline{\text{OE}}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times ($X \leq 10$).

(7) Program verify mode

Setting $\overline{\text{CE}}$ to L, $\overline{\text{PGM}}$ to H, and $\overline{\text{OE}}$ to L sets the program verify mode.

After writing is performed, this mode should be used to check whether the data was written correctly.

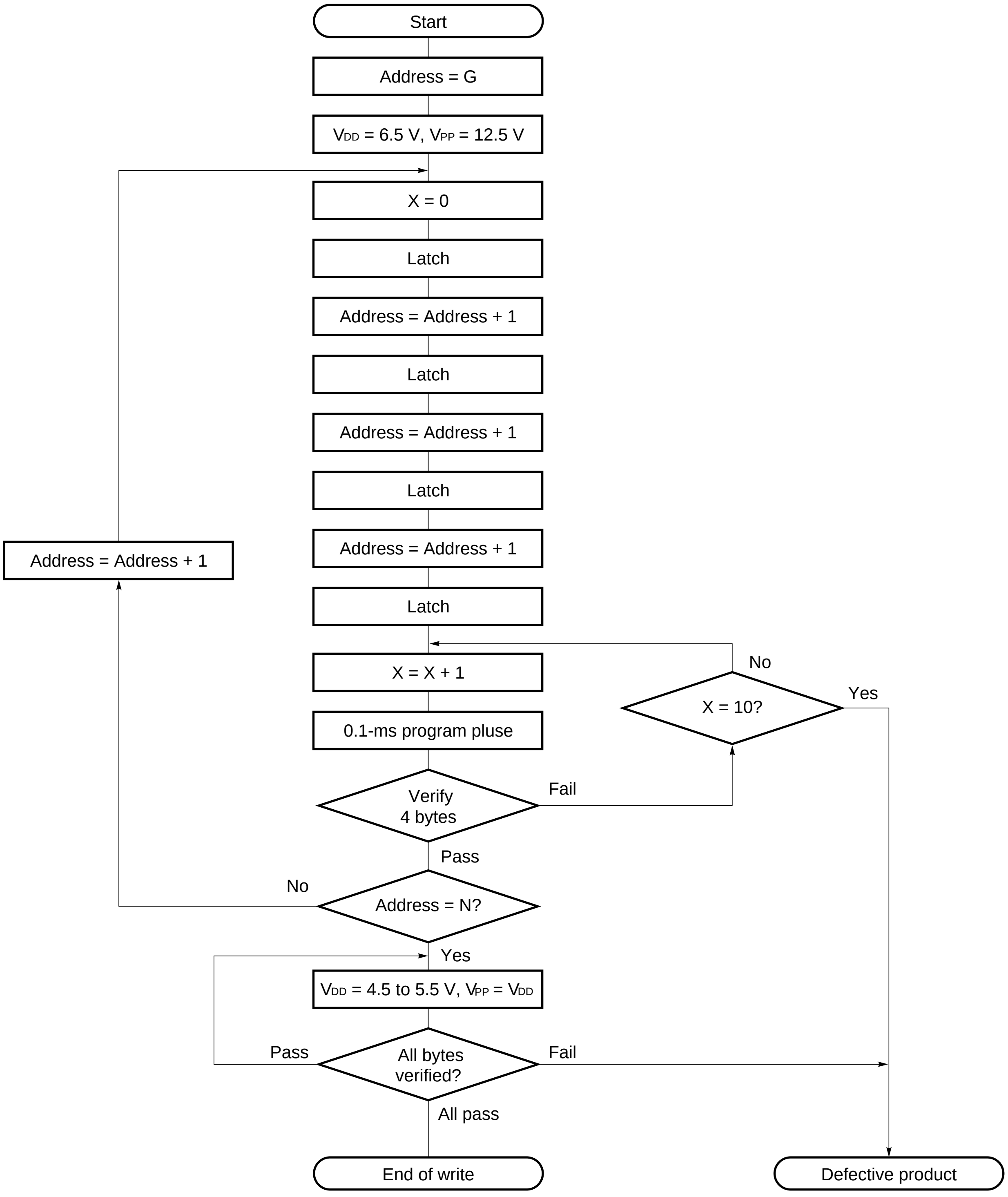
(8) Program inhibit mode

The program inhibit mode is used when the $\overline{\text{OE}}$ pins, V_{PP} pins and pins D0 to D7 of multiple μPD78P048As are connected in parallel, and when you wish to write to one of these devices.

The page write mode or byte write mode described above is used to perform a write. At this time, the write is not performed on the device which has the $\overline{\text{PGM}}$ pin driven high.

20.3.2 PROM write procedure

Figure 20-3. Page Program Mode Flowchart



Remarks 1. G = Start address

2. N = Last address of program

Figure 20-4. Page Program Mode Timing

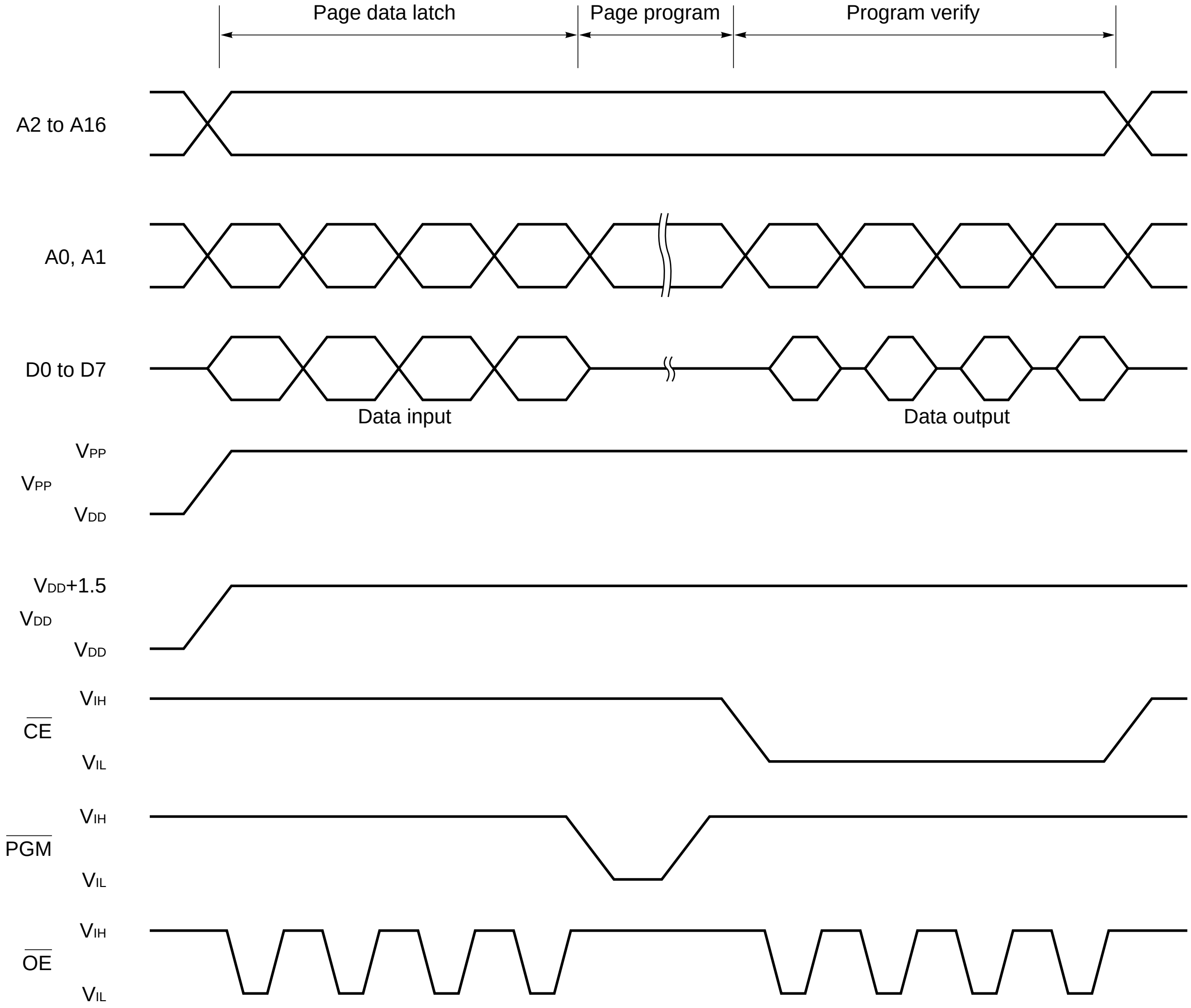
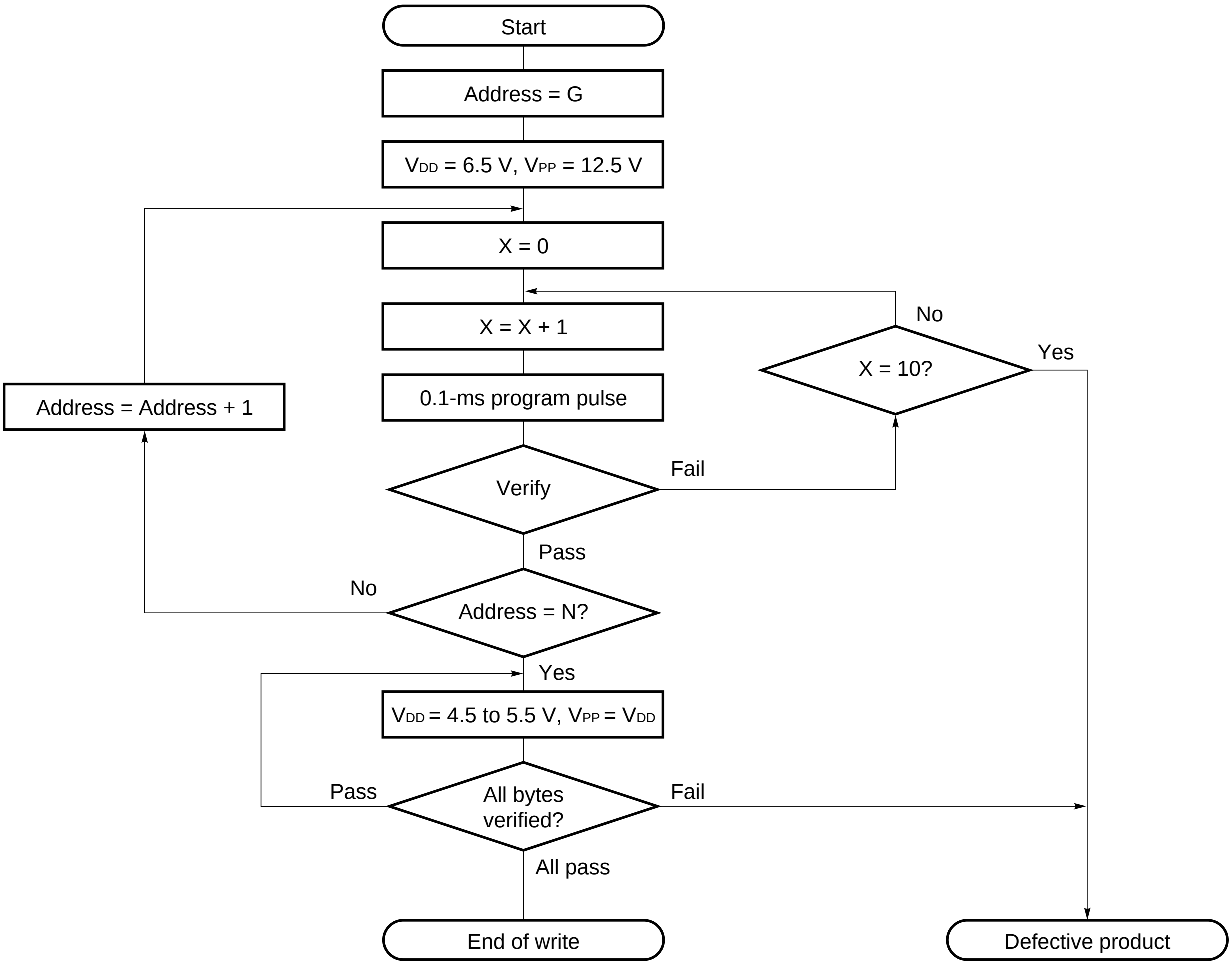


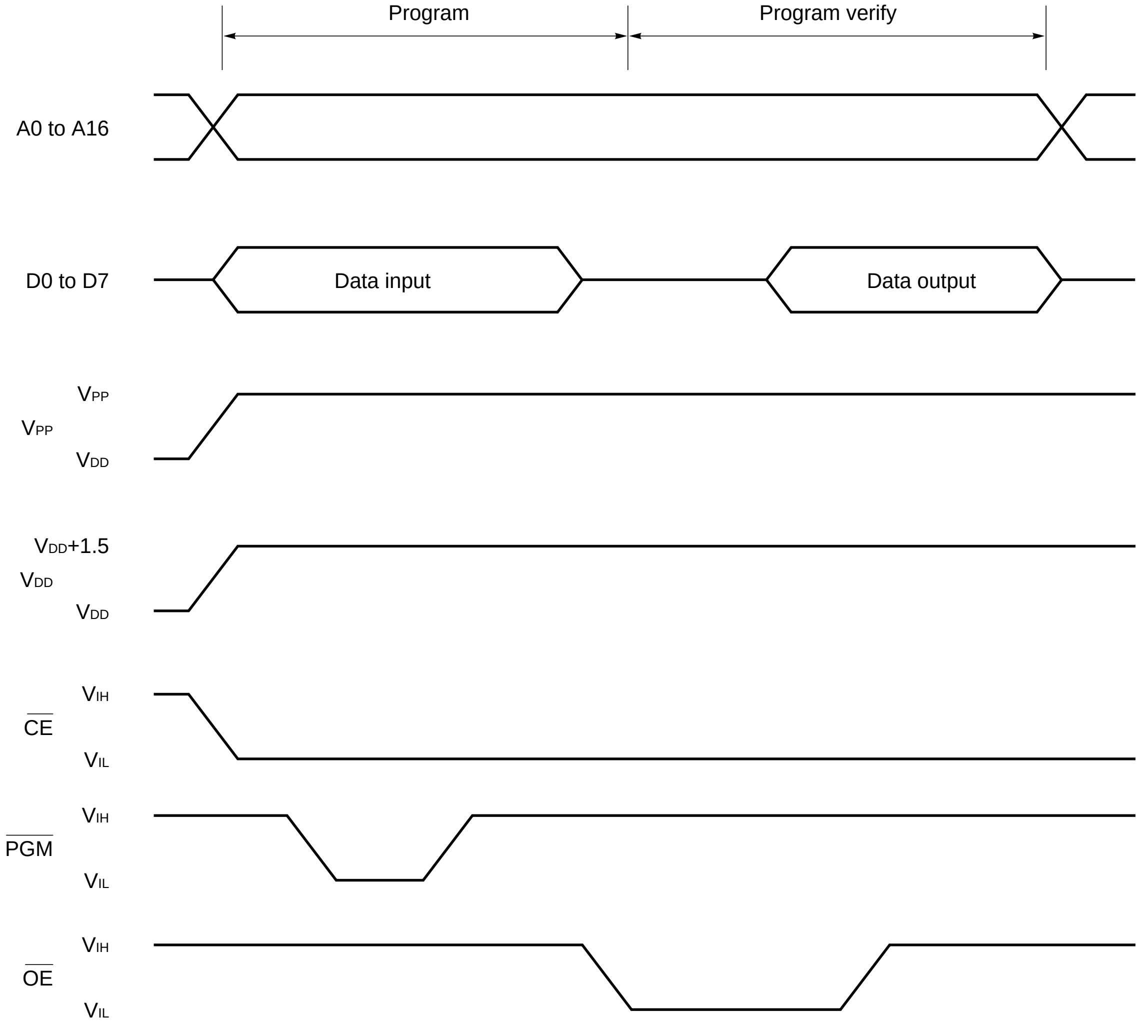
Figure 20-5. Byte Program Mode Flowchart



Remarks 1. G = Start address

2. N = Last address of program

Figure 20-6. Byte Program Mode Timing



Cautions 1. Ensure that V_{DD} is applied before V_{PP} and removed after V_{PP}.

2. Ensure that V_{PP} does not exceed +13.5 V including overshoot.

3. Removing the device while +12.5 V is being applied to V_{PP} may have an adverse affect on reliability.

20.3.3 PROM read procedure

PROM contents can be read onto the external data bus (D0 to D7) using the following procedure.

(1) Fix the $\overline{\text{RESET}}$ pin low, and supply +5 V to the V_{PP} pin. Unused pins are handled as shown in **1.5 "Pin Configuration, (2) PROM programming mode."**

(2) Supply +5 V to the V_{DD} and V_{PP} pins.

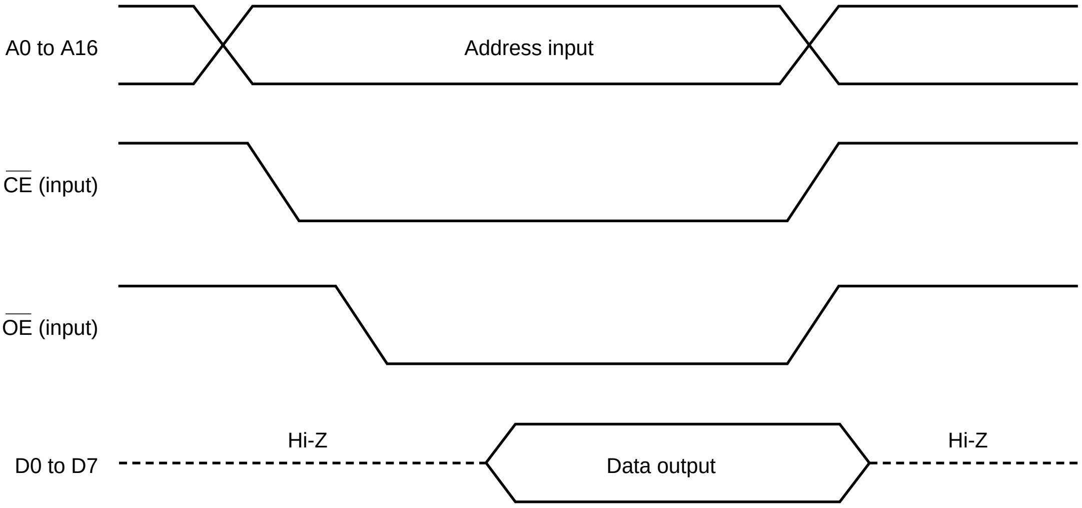
(3) Input address of data to be read to pins A0 to A16.

(4) Read mode.

(5) Output data to pins D0 to D7.

The timing for steps (2) through (5) above is shown in Figure 20-7.

Figure 20-7. PROM Read Timing



20.4 Erasure Characteristics (for μPD78P048AKL-S only)

The μPD78P048AKL-S's allows users to erase the data written in the program memory (FFH) and to rewrite data into it.

To erase the data, the erasure window on the top of the package should be exposed to light having a very short wavelength of less than 400 nm (typically 254 nm). The light intensity and exposure time required to completely erase all data are:

- Ultraviolet ray strength x Exposure time = 30 W•s/cm^2 or more
- Erasure time = 40 minutes or longer (with a 12 mW/cm^2 ultraviolet ray lamp)
  (When the ultraviolet ray lamp has been deteriorated or the erasure window is dirty, however, the time will be longer.)

Before starting an erasure process, place a μPD78P048AKL-S within 2.5 cm of the ultraviolet ray lamp and remove the filter on the ultraviolet ray lamp if it is attached.

20.5 Opaque Film on Erasure Window (for μPD78P048AKL-S only)

Unless erasing EPROM contents, be sure to cover the erasure window with a shading film to prevent an unintentional erasure of EPROM contents or malfunctions of the internal circuits other than EPROM due to light coming in through the window.

20.6 Screening of One-Time PROM Versions

One-time PROM version (μPD78P048AGF-3B9) devices cannot be fully tested by NEC before shipment due to the nature of PROM. After necessary data have been written, it is recommended to implement a screening process, that is, the written contents should be verified after the devices have been stored under the following high-temperature condition.

Storage Temperature	Storage Time
125°C	24 hours

[MEMO]

CHAPTER 21 INSTRUCTION SET

This chapter describes the instruction set for the μPD78044F Subseries. For the details of operations and machine languages (instruction codes) of each instruction, refer to **78K/0 Series User's Manual**: **Instructions** (U12326E).

21.1 Legend

21.1.1 Operand identifiers and description methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and are described as they are. Each symbol has the following meaning.

- # : Immediate data specification
- ! : Absolute address specification
- $: Relative address specification
- [ ] : Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 21-1. Operand Identifiers and Description Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol Note
sfrp	Special function register symbols (16-bit manipulatable register even addresses only) Note
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even addresses only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note FFD0H to FFDFH are not addressable.

Remark For special-function register symbols, refer to **Table 3-4 Special Function Register List**.

21.1.2 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 servicing flag
( )	:	Memory contents indicated by address or register contents in parentheses
X_H, X_L	:	Higher 8 bits and lower 8 bits of 16-bit register
∧	:	Logical product (AND)
∨	:	Logical sum (OR)
∀	:	Exclusive logical sum (exclusive OR)
——	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

21.1.3 Description of “flag operation” column

(Blank)	:	Unchanged
0	:	Cleared to 0
1	:	Set to 1
×	:	Set/cleared according to the result
R	:	Previously saved value is restored

21.2 Operation List

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-Bit Data Transfer	MOV	r, #byte	2	4	–	r ← byte			
		saddr, #byte	3	6	7	(saddr) ← byte			
		sfr, #byte	3	–	7	sfr ← byte			
		A, r Note 3	1	2	–	A ← r			
		r, A Note 3	1	2	–	r ← A			
		A, saddr	2	4	5	A ← (saddr)			
		saddr, A	2	4	5	(saddr) ← A			
		A, sfr	2	–	5	A ← sfr			
		sfr, A	2	–	5	sfr ← A			
		A, !addr16	3	8	9	A ← (addr16)			
		!addr16, A	3	8	9	(addr16) ← A			
		PSW, #byte	3	–	7	PSW ← byte	×	×	×
		A, PSW	2	–	5	A ← PSW			
		PSW, A	2	–	5	PSW ← A	×	×	×
		A, [DE]	1	4	5	A ← (DE)			
		[DE], A	1	4	5	(DE) ← A			
		A, [HL]	1	4	5	A ← (HL)			
		[HL], A	1	4	5	(HL) ← A			
		A, [HL+byte]	2	8	9	A ← (HL+byte)			
		[HL+byte], A	2	8	9	(HL+byte) ← A			
		A, [HL+B]	1	6	7	A ← (HL+B)			
		[HL+B], A	1	6	7	(HL+B) ← A			
		A, [HL+C]	1	6	7	A ← (HL+C)			
		[HL+C], A	1	6	7	(HL+C) ← A			
	XCH	A, r Note 3	1	2	–	A ↔ r			
		A, saddr	2	4	6	A ↔ (saddr)			
		A, sfr	2	–	6	A ↔ (sfr)			
		A, !addr16	3	8	10	A ↔ (addr16)			
		A, [DE]	1	4	6	A ↔ (DE)			
		A, [HL]	1	4	6	A ↔ (HL)			
		A, [HL+byte]	2	8	10	A ↔ (HL+byte)			
		A, [HL+B]	2	8	10	A ↔ (HL+B)			
		A, [HL+C]	2	8	10	A ↔ (HL+C)			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access
2. When an area except the internal high-speed RAM area is accessed
3. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-Bit Data Transfer	MOVW	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp **Note 3**	1	4	–	AX ← rp			
		rp, AX **Note 3**	1	4	–	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
		!addr16, AX	3	10	12	(addr16) ← AX			
	XCHW	AX, rp **Note 3**	1	4	–	AX ↔ rp			
8-Bit Operation	ADD	A, #byte	2	4	–	A, CY ← A+byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)+byte	×	×	×
		A, r **Note 4**	2	4	–	A, CY ← A+r	×	×	×
		r, A	2	4	–	r, CY ← r+A	×	×	×
		A, saddr	2	4	5	A, CY ← A+(saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A+(addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A+(HL)	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A+(HL+byte)	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A+(HL+B)	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A+(HL+C)	×	×	×
	ADDC	A, #byte	2	4	–	A, CY ← A+byte+CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)+byte+CY	×	×	×
		A, r **Note 4**	2	4	–	A, CY ← A+r+CY	×	×	×
		r, A	2	4	–	r, CY ← r+A+CY	×	×	×
		A, saddr	2	4	5	A, CY ← A+(saddr)+CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A+(addr16)+CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A+(HL)+CY	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A+(HL+byte)+CY	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A+(HL+B)+CY	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A+(HL+C)+CY	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-Bit Operation	SUB	A, #byte	2	4	–	A, CY ← A–byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)–byte	×	×	×
		A, r [Note 3]	2	4	–	A, CY ← A–r	×	×	×
		r, A	2	4	–	r, CY ← r–A	×	×	×
		A, saddr	2	4	5	A, CY ← A–(saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A–(addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A–(HL)	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A–(HL+byte)	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A–(HL+B)	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A–(HL+C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A–byte–CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr)–byte–CY	×	×	×
		A, r [Note 3]	2	4	–	A, CY ← A–r–CY	×	×	×
		r, A	2	4	–	r, CY ← r–A–CY	×	×	×
		A, saddr	2	4	5	A, CY ← A–(saddr)–CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A–(addr16)–CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A–(HL)–CY	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A–(HL+byte)–CY	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A–(HL+B)–CY	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A–(HL+C)–CY	×	×	×
	AND	A, #byte	2	4	–	A ← A∧byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∧byte	×		
		A, r [Note 3]	2	4	–	A ← A∧r	×		
		r, A	2	4	–	r ← r∧A	×		
		A, saddr	2	4	5	A ← A∧(saddr)	×		
		A, !addr16	3	8	9	A ← A∧(addr16)	×		
		A, [HL]	1	4	5	A ← A∧(HL)	×		
		A, [HL+byte]	2	8	9	A ← A∧(HL+byte)	×		
		A, [HL+B]	2	8	9	A ← A∧(HL+B)	×		
		A, [HL+C]	2	8	9	A ← A∧(HL+C)	×		

Notes **1.** When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-Bit Operation	OR	A, #byte	2	4	–	A ← A∨ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr)∨ byte	×		
		A, r Note 3	2	4	–	A ← A∨ r	×		
		r, A	2	4	–	r ← r∨ A	×		
		A, saddr	2	4	5	A ← A ∨(saddr)	×		
		A, !addr16	3	8	9	A ← A∨ (addr16)	×		
		A, [HL]	1	4	5	A ← A∨ (HL)	×		
		A, [HL+byte]	2	8	9	A ← A∨ (HL+byte)	×		
		A, [HL+B]	2	8	9	A ← A∨ (HL+B)	×		
		A, [HL+C]	2	8	9	A ← A ∨(HL+C)	×		
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×		
		A, r Note 3	2	4	–	A ← A⊻ r	×		
		r, A	2	4	–	r ← r ⊻ A	×		
		A, saddr	2	4	5	A ← A⊻ (saddr)	×		
		A, !addr16	3	8	9	A ← A⊻ (addr16)	×		
		A, [HL]	1	4	5	A ← A⊻ (HL)	×		
		A, [HL+byte]	2	8	9	A ← A⊻ (HL+byte)	×		
		A, [HL+B]	2	8	9	A ← A⊻ (HL+B)	×		
		A, [HL+C]	2	8	9	A ← A⊻ (HL+C)	×		
	CMP	A, #byte	2	4	–	A–byte	×	×	×
		saddr, #byte	3	6	8	(saddr)–byte	×	×	×
		A, r Note 3	2	4	–	A–r	×	×	×
		r, A	2	4	–	r–A	×	×	×
		A, saddr	2	4	5	A–(saddr)	×	×	×
		A, !addr16	3	8	9	A–(addr16)	×	×	×
		A, [HL]	1	4	5	A–(HL)	×	×	×
		A, [HL+byte]	2	8	9	A–(HL+byte)	×	×	×
		A, [HL+B]	2	8	9	A–(HL+B)	×	×	×
		A, [HL+C]	2	8	9	A–(HL+C)	×	×	×

Notes **1.** When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except r = A

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-Bit Operation	ADDW	AX, #word	3	6	–	AX, CY ← AX+word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX–word	×	×	×
	CMPW	AX, #word	3	6	–	AX–word	×	×	×
Multiple/ Divide	MULU	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increase/ Decrease	INC	r	1	2	–	r ← r+1	×	×	
		saddr	2	4	6	(saddr) ← (saddr)+1	×	×	
	DEC	r	1	2	–	r ← r–1	×	×	
		saddr	2	4	6	(saddr) ← (saddr)–1	×	×	
	INCW	rp	1	4	–	rp ← rp+1			
	DECW	rp	1	4	–	rp ← rp–1			
Rotation	ROR	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, ← $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, ← $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD Correction	ADJBA		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit Manipulation	MOV1	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Bit Manipulation	AND1	CY, saddr.bit	3	6	7	CY ← CY∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY∀ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY∀ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY∀ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY∀ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY∀ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{\text{CY}}$			×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Call Return	CALL	!addr16	3	7	–	(SP–1) ← $(PC+3)_H$, (SP–2) ← $(PC+3)_L$, PC ← addr16, SP ← SP–2			
	CALLF	!addr11	2	5	–	(SP–1) ← $(PC+2)_H$, (SP–2) ← $(PC+2)_L$, $PC_{15–11}$ ← 00001, $PC_{10–0}$ ← addr11, SP ← SP–2			
	CALLT	[addr5]	1	6	–	(SP–1) ← $(PC+1)_H$, (SP–2) ← $(PC+1)_L$, PC_H ← (00000000, addr5+1), PC_L ← (00000000, addr5) SP ← SP–2			
	BRK		1	6	–	(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	6	–	PC_H ← (SP+1), PC_L ← (SP), SP ← SP+2			
	RETI		1	6	–	PC_H ← (SP+1), PC_L ← (SP), PSW ← (SP+2), SP ← SP+3, NMIS ← 0	R	R	R
	RETB		1	6	–	PC_H ← (SP+1), PC_L ← (SP), PSW ← (SP+2), SP ← SP+3	R	R	R
Stack Manipulation	PUSH	PSW	1	2	–	(SP–1) ← PSW, SP ← SP–1			
		rp	1	4	–	(SP–1) ← rp_H, (SP–2) ← rp_L, SP ← SP–2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP+1	R	R	R
		rp	1	4	–	rp_H ← (SP+1), rp_L ← (SP), SP ← SP+2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional Branch	BR	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 jdisp8			
		AX	2	8	–	PC_H ← A, PC_L ← X			
Conditional Branch	BC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Conditional Branch	BT	saddr.bit, $addr16	3	8	9	PC ← PC+3+jdisp8 if(saddr.bit)=1			
		sfr.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if sfr.bit=1			
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit=1			
		PSW.bit, $addr16	3	–	9	PC ← PC+3+jdisp8 if PSW.bit=1			
		[HL].bit, $addr16	3	10	11	PC ← PC+3+jdisp8 if (HL).bit=1			
	BF	saddr.bit, $addr16	4	10	11	PC ← PC+4+jdisp8 if (saddr.bit)=0			
		sfr.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if sfr.bit=0			
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit=0			
		PSW.bit, $addr16	4	–	11	PC ← PC+4+jdisp8 if PSW.bit=0			
		[HL].bit, $addr16	3	10	11	PC ← PC+3+jdisp8 if (HL).bit=0			
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC+4+jdisp8 if(saddr.bit)=1 then reset (saddr.bit)			
		sfr.bit, $addr16	4	–	12	PC ← PC+4+jdisp8 if sfr.bit=1 then reset sfr.bit			
		A.bit, $addr16	3	8	–	PC ← PC+3+jdisp8 if A.bit=1 then reset A.bit			
		PSW.bit, $addr16	4	–	12	PC ← PC+4+jdisp8 if PSW.bit=1 then reset PSW.bit	×	×	×
		[HL].bit.$addr16	3	10	12	PC ← PC+3+jdisp8 if (HL).bit=1 then reset (HL).bit			
	DBNZ	B, $addr16	2	6	–	B ← B–1, then PC ← PC+2+jdisp8 if B≠0			
		C, $addr16	2	6	–	C ← C–1, then PC ← PC+2+jdisp8 if C≠0			
		saddr, $addr16	3	8	10	(saddr) ← (saddr)–1, then PC ← PC+3+jdisp8 if (saddr)≠0			
CPU Control	SEL	RBn	2	4	–	RBS1, 0 ← n			
	NOP		1	2	–	No Operation			
	EI		2	–	6	IE ← 1 (Enable Interrupt)			
	DI		2	–	6	IE ← 0 (Disable Interrupt)			
	HALT		2	6	–	Set HALT Mode			
	STOP		2	6	–	Set STOP Mode			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remark One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control (PCC) register.

21.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4,ROL4, PUSH, POP, DBNZ

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 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 instructions/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 instruciton					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

APPENDIX A DIFFERENCES AMONG μPD78044, μPD78044A, AND μPD78044F SUBSERIES

The differences among the μPD78044, μPD78044A, and μPD78044A Subseries are described below. Apart from these points, these three subseries have the same functions.

Table A-1. Differences among μPD78044, 78044A, and 78044F Subseries

Subseries Parameter	**μPD78044F Subseries**	μPD78044A Subseries	μPD78044 Subseries
Supply voltage	V_{DD} = 2.7 to 5.5 V	V_{DD} = 2.7 to 6.0 V	V_{DD} = 2.7 to 6.0 V
Internal high-speed RAM size (Mask ROM)	μPD78042F: 512 bytes μPD78043F: 512 bytes μPD78044F: 1024 bytes μPD78045F: 1024 bytes	μPD78042A: 512 bytes μPD78043A: 512 bytes μPD78044A: 1024 bytes μPD78045A: 1024 bytes	μPD78042: 512 bytes μPD78043: 1024 bytes μPD78044: 1024 bytes
Internal high-speed RAM size (PROM)	μPD78P048A: 1024 bytes	μPD78P048A:1024 bytes	μPD78P044: 1024 bytes
Internal expansion RAM size	Available only with μPD78P048A: 1024 bytes		None
Minimum instruction execution time	0.4 μs (@ 5.0 MHz)		0.48 μs (@ 4.19 MHz)
Main system clock oscillation frequency (maximum value)	5.0 MHz		4.19 MHz
A/D converter conversion time	19.1 μs (min.) to 200 μs (max.)		38.1 μs (min.) to 100 μs (max.)
Bit 5 (SIC) of interrupt timing specify register (SINT) of serial interface channel 0 (Selection of INTCSI0 interrupt causes)	If SIC is set to 1, CSIIF0 will be set when a bus release is detected or when a transfer is completed.		If SIC is set to 1, CSIIF0 will be set when a bus release is detected.
★ Display output current of FIP controller driver (V_{DD} = 4.5 to 6.0 V Note, V_{OD} = V_{DD} – 2 V)	–25 mA TYP. (FIP0 to FIP33)	–18 mA TYP. (FIP0 to FIP33)	–22 mA TYP. (FIP0 to FIP15) –5.5 mA TYP. (FIP16 to FIP33)
Programmer Adapter	PA-78P048GF PA-78P048KL-S		PA-78P044GF PA-78P044KL-S

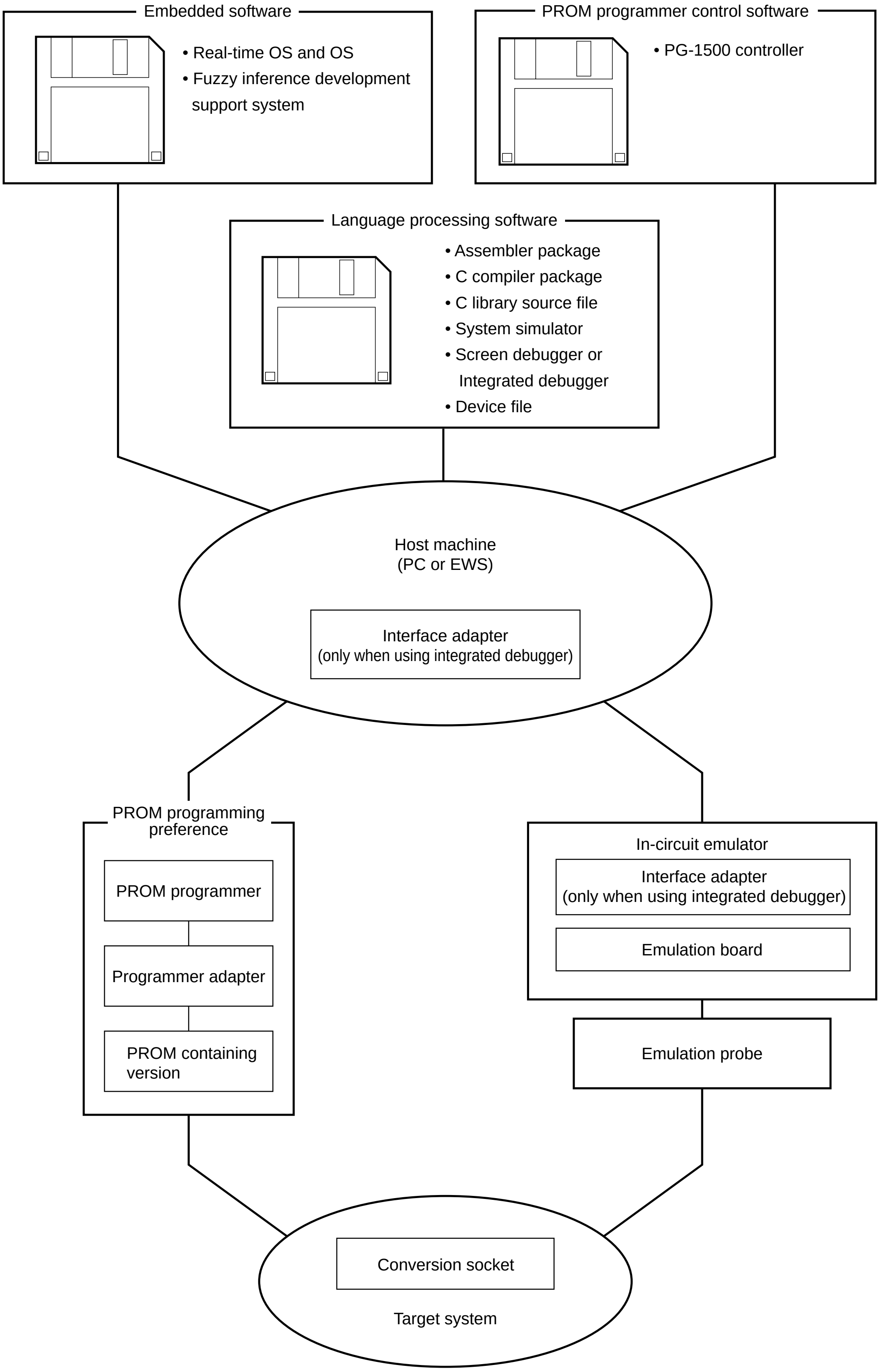
Note μPD78044F Subseries: V_{DD} = 4.5 to 5.5 V

[MEMO]

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD78044F Subseries. Figure B-1 shows the configuration of the development tools.

Figure B-1. Development Tools



B.1 Language Processing Software

RA78K/0 assembler package	This software is a program that translates programs described in mnemonics into object codes with which microcontrollers can operate. In addition, it can also generate symbol tables, and perform optimization of branch instructions automatically. This program should be used together with the separately available device file (DF78044).
	Part number: μSxxxxRA78K0
CC78K/0 C compiler package	This software is a program that translates programs described in C language into object codes with which microcontrollers can operate. This program should be used together with the separately available assembler package (RA78K/0)and device file (DF78044).
	Part number: μSxxxxCC78K0
DF78044 device file [Note]	This file stores device-proprietary information and can be used for all devices in the μPD78044A subseries. This file should be used together with the separately available programs: RA78K/0, CC78K/0, SM78K0, ID78K0, and SD78K/0.
	Part number: μSxxxxDF78044
CC78K/0-L C library source file	This file contains source programs of the functions registered in the object libraries for the C compiler package (CC78K/0). This file is required if users have to modify some CC78K/0 object libraries according to user's application specifications.
	Part number: μSxxxxCC78K0-L

Note DF78044 can be used with any of RA78K/0, CC78K/0, SM78K0, ID78K0, and SD78K/0.

Remark The xxxx in the part number differs depending on the host machine and operating system to be used.

μSxxxxRA78K0
μSxxxxCC78K0
μSxxxxDF78044
μSxxxxCC78K0-L

xxxx	Host Machine	Operating System	Supply Media
5A13 5A10	PC-9800 Series	MS-DOS (Ver. 3.30 to 6.2 [Note])	3.5-inch 2HD 5-inch 2HD
7B13 7B10	IBM PC/AT or compatibles	See section **B.4**.	3.5-inch 2HC 5-inch 2HC
3H15	HP9000 Series 300™	HP-UX™ (rel. 7.05B)	Cartridge tape (QIC-24)
3P16	HP9000 Series 700™	HP-UX (rel. 9.01)	Digital audio tape (DAT)
3K15	SPARCstation™	SunOS™ (rel. 4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 Series (RISC)	EWS-UX/V (rel. 4.0)	

Note Although the task swap function is incorporated in MS-DOS Ver. 5.0 and later versions, this function cannot be used with the above software.

B.2 PROM Writing Tools

B.2.1 Hardware

PG-1500 PROM programmer	This is a PROM programmer capable of programming the single-chip microcontroller incorporated in the PROM by manipulating from the stand-alone or host machine through connection of a program adapter separately purchasable and attached board. It can also program representative PROMs ranging from 256K bits to 4M bits.
PA-78P048GF PA-78P048KL-S PROM programmer adapter	PROM programmer adapter common to the μPD78P048A and is connected to the PG-1500. PA-78P048GF : 80-pin plastic QFP (GF-3B9 type) PA-78P048KL-S : 80-pin ceramic WQFN (KL-S type)

B.2.2 Software

PG-1500 controller	The PG-1500 is controlled in the host machine through connection with the host machine and PG-1500 via serial and parallel interfaces.
	Part Number : μSxxxxPG1500

Remark The xxxx in the part number differs depending on the host machine and operating system to be used.

μSxxxxPG1500

xxxx	Host Machine	Operating System	Supply Medium
5A13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2$^{\text{Note}}$)	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT and compatible machines	Refer to **B.4**	3.5-inch 2HD
7B10			5-inch 2HC

Note Although the task swap function is incorporated in MS-DOS Ver.5.0 and later versions, this function cannot be used with the above software.

B.3 Debugging Tools

B.3.1 Hardware

★	IE-78000-R-A in-circuit emulator (supports integrated debugger)	The IE-78000-R-A is an in-circuit emulator that serves to debug hardware and software when developing application systems using the 78K/0 Series. The IE-78000-R-A supports the integrated debugger (ID78K0) and used in combination with an emulation probe and an interface adapter for connection with a host machine.
★	IE-70000-98-IF-B interface adapter	The IE-70000-98-IF-B is an adapter necessary when using the PC-9800 Series (except notebook personal computers) as a host machine for the IE-78000-R-A.
★	IE-70000-98N-IF interface adapter	The IE-70000-98N-IF is an adapter/cable necessary when using PC-9800 Series notebook computer as a host machine for the IE-78000-R-A.
★	IE-70000-PC-IF-B interface adapter	The IE-70000-PC-IF-B is an adapter necessary when using IBM PC/AT as a host machine for the IE-78000-R-A.
★	IE-78000-R-SV3 interface adapter	The IE-78000-R-SV3 is an adapter/cable necessary when using EWS as a host machine for the IE-78000-R-A. Use the IE-78000-R-SV3 connected to the board in the IE-78000-R-A. The IE-78000-R-SV3 supports 10 Base-5 for Ethernet™. A commercially available adapter is required for other communication systems.
	IE-78000-R in-circuit emulator (supports screen debugger)	The IE-78000-R is an in-circuit emulator that serves to debug hardware and software when developing application systems using the 78K/0 Series. The IE-78000-R supports the screen debugger (SD78K/0). Debugging can be executed efficiently through connection with the host machine and the PROM programmer.
	IE-78044-R-EM emulation board	The IE-78044-R-EM is a board used to perform emulation of the hardware inherent to the device. Use the IE-78044-R-EM in combination with an in-circuit emulator.
	EP-78130GF-R emulation probe	The EP-78130GF-R is a probe to connect an in-circuit emulator to the target system. The EP-78130GF-R is for 80-pin plastic QFP (GF-3B9 type). An 80-pin conversion socket EV-9200G-80 is provided to facilitate user system development.
	EV-9200G-80 conversion socket	Conversion socket to connect the EP-78130GF-R and target system board created to mount the 80-pin plastic QFP (GF-3B9 type). The μPD78P048AKL-S (ceramic WQFN) can be mounted instead of connecting the EP-78130GF-R.
	EV-9900	A jig used to remove the μPD78P048AKL-S from the EV-9200G-80.

Remark EV-9200G-80s are sold in sets of five units.

B.3.2 Software (1/3)

SM78K0 system simulator	Debugs program at C source level or assembler level while simulating operation of target system on host machine. SM78K0 runs on Windows. By using the SM78K0, the logic and performance of an application can be verified independently of hardware development even when the In-circuit emulator is not used. This enhances development efficiency and improves software quality. Used in combination with optional device file (DF78044).
	Part Number : μSxxxxSM78K0

Remark The xxxx in the part number differs depending on the host machine and operating system to be used.

★ μSxxxxSM78K0

xxxx	Host Machine	Operating System	Supply Medium
AA13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2 Note) + Windows (Ver.3.0 to Ver.3.1)	3.5-inch 2HD
AB13	IBM PC/AT and compatible machines (Windows Japanese version)	Refer to **B.4**	3.5-inch 2HC
BB13	IBM PC/AT and compatible machines (Windows English version)		

Note Although the task swap function is incorporated in MS-DOS Ver.5.0 or later versions, this function cannot be used with the above software.

B.3.2 Software (2/3)

★

ID78K0 integrated debugger	The ID78K0 is a control program to debug the 78K/0 Series. The ID78K0 uses Windows on personal computers and OSF/Motif™ on EWS as graphical interface, and presents the appearance and operatability conforming to these platforms. The ID78K0 has enhanced debugging function supporting C language, and the trace result can be displayed by using the window integration function that interlocks source program, disassemble display, and memory display to the trace result. In addition, debugging efficiency for programs using real time OS can be enhanced by incorporating extension modules such as task debugger and system performance analyzer. Used in combination with optional device file (DF78044).
	Part Number : μSxxxxID78K0

Remark The xxxx in the part number differs depending on the host machine and operating system to be used.

μSxxxxID78K0

xxxx	Host Machine	Operating System	Supply Medium
AA13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2 Note) + Windows (Ver.3.1)	3.5-inch 2HD
AB13	IBM PC/AT and their compatible machines (Windows Japanese version)	Refer to **B.4**	3.5-inch 2HC
BB13	IBM PC/AT and their compatible machines (Windows English version)		
3P16	HP9000 Series 700	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3K13			3.5-inch 2HC
3R16	NEWS™ (RISC)	NEWS-OS™ (6.1x)	1/4-inch CGMT
3R13			3.5-inch 2HC
3M15	EWS4800 Series (RISC)	EWS-UX/V (rel.4.0)	Cartridge tape (QIC-24)

Note Although the task swap function is incorporated in MS-DOS Ver.5.0 or later versions, this function cannot be used with the above software.

B.3.2 Software (3/3)

SD78K/0 screen debugger	Connects IE-78000-R to host machine with serial interface (RS-232-C) to control IE-78000-R on host machine. Used in combination with optional device file (DF78044).
	Part Number : µSxxxxSD78K0
DF78044 Note device file	This is a file containing the information inherent to the device. Used in combination with optional RA78K/0, CC78K/0, SM78K0, ID78K0, or SD78K/0.
	Part Number : µSxxxxDF78044

Note The DF78044 can be used with any of the RA78K/0, CC78K/0, SM78K0, ID78K0, and SD78K/0 products.

Remark The xxxx in the part number differs depending on the host machine and operating system to be used.

µSxxxxSD78K0
µSxxxxDF78044

xxxx	Host Machine	Operating System	Supply Medium
5A13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2 Note)	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT and their compatible machines (Windows Japanese version)	Refer to **B.4**	3.5-inch 2HC
7B10			5-inch 2HC

Note Although the task swap function is incorporated in MS-DOS Ver.5.0 or later versions, this function cannot be used with the above software.

B.4 Operating Systems for IBM PC

The following software products are supported as the operating systems for IBM PC. When SM78K0, ID78K0, or FE9200 (See Section **C.2 Fuzzy Inference Development Support System.**) will be operated, Windows (Ver. 3.0 or 3.1) is required.

Operating System	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 the English mode is supported.

Caution Although the task swap function is incorporated in MS-DOS Ver. 5.0 or later versions, this function cannot be used with the above software.

B.5 System-Upgrade Method from Other In-Circuit Emulator to 78K/0 Series In-Circuit Emulator

If you already own an in-circuit emulator for the 78K Series or 75X/XL Series, you can use it for the 78K/0 Series with the same functionality as the IE-78000-R or IE-78000-R-A by replacing the break board in the main unit with the IE-78000-R-BK.

Table B-1. System-Up Method from Other In-Circuit Emulator to IE-78000-R

Series Name	Owned In-Circuit Emulator	Board to be Purchased
75X/XL Series	IE-75000-R [Note], IE-75001-R	IE-78000-R-BK
78K/I Series	IE-78130-R, IE-78140-R	
78K/II Series	IE-78230-R [Note], IE-78230-R-A, IE-78240-R [Note], IE-78240-R-A	
78K/III Series	IE-78320-R [Note], IE-78327-R, IE-78330-R, IE-78350-R	

Note Maintenance parts

★ **Table B-2. System-Up Method from Other In-Circuit Emulator to IE-78000-R-A**

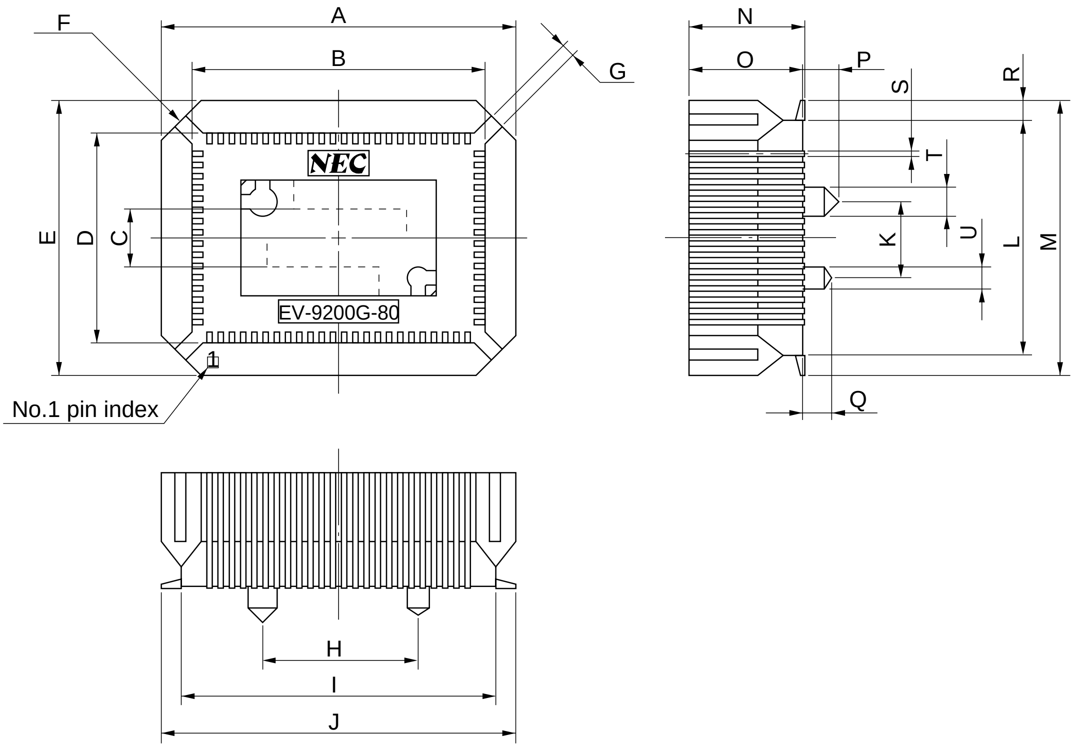
Series Name	Owned In-Circuit Emulator	Board to be Purchased
75X/XL Series	IE-75000-R [Note 1], IE-75001-R	IE-78000-R-BK [Note 2]
78K/I Series	IE-78130-R, IE-78140-R,	
78K/II Series	IE-78230-R [Note 1], IE-78230-R-A, IE-78240-R [Note 1], IE-78240-R-A	
78K/III Series	IE-78320-R [Note 1], IE-78327-R, IE-78330-R, IE-78350-R	
78K/0 Series	IE-78000-R	— [Note 2]

Notes 1. Maintenance parts

2. Partial remodelling of the frame of the in-circuit emulator and replacement of control/trace board with a supervisor board must be done by NEC.

Package Drawing and Recommended Footprint of Conversion Socket (EV-9200G-80)

Figure B-2. EV-9200G-80 Package Drawing (for Reference Only)

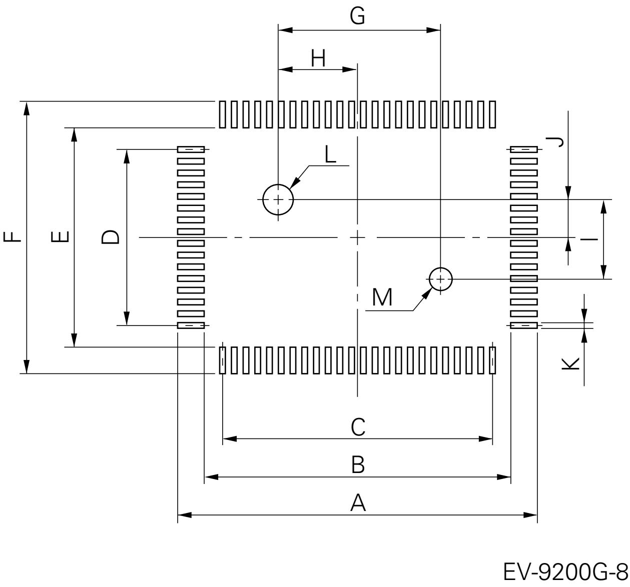


EV-9200G-80-G0

ITEM	MILLIMETERS	INCHES
A	25.0	0.984
B	20.30	0.799
C	4.0	0.157
D	14.45	0.569
E	19.0	0.748
F	4-C 2.8	4-C 0.11
G	0.8	0.031
H	11.0	0.433
I	22.0	0.866
J	24.7	0.972
K	5.0	0.197
L	16.2	0.638
M	18.9	0.744
N	8.0	0.315
O	7.8	0.307
P	2.5	0.098
Q	2.0	0.079
R	1.35	0.053
S	0.35±0.1	$0.014^{+0.004}_{-0.005}$
T	ϕ2.3	ϕ0.091
U	ϕ1.5	ϕ0.059

Figure B-3. Recommended EV-9200G-80 Footprint (for Reference Only)

Based on EV-9200G-80
(2) Pad drawing (in mm)



ITEM	MILLIMETERS	INCHES
A	25.7	1.012
B	21.0	0.827
C	0.8±0.02 × 23=18.4±0.05	$0.031^{+0.002}_{-0.001} \times 0.906=0.724^{+0.003}_{-0.002}$
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	15.2	0.598
F	19.9	0.783
G	11.00±0.08	$0.433^{+0.004}_{-0.003}$
H	5.50±0.03	$0.217^{+0.001}_{-0.002}$
I	5.00±0.08	$0.197^{+0.003}_{-0.004}$
J	2.50±0.03	$0.098^{+0.002}_{-0.001}$
K	0.5±0.02	$0.02^{+0.001}_{-0.002}$
L	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
M	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).

[MEMO]

APPENDIX C EMBEDDED SOFTWARE

This section describes the embedded software av-ailable for users to develop and maintain programs for the μPD78044F Subseries.

C.1 Real-time OS (1/2)

RX78K/0 real-time OS	RX78K/0 is a real-time OS which is based on the μITRON specification. Supplied with the RX78K/0 nucleus and a tool to prepare multiple information tables (configurator). This software should be used together with the separately available assembler package (RA78K/0) and device file (DF78044)
	Part Number: μSxxxxRX78013-ΔΔΔΔ

Caution When purchasing the RX78K/0, fill in the purchase application form in advance and sign the license agreement.

Remark The xxxx and ΔΔΔΔ in the part number differ depending on the host machine, operating system, and other factors as shown below.

μSxxxxRX78013-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	100,000
001M		1,000,000
010M		10,000,000
S01	Source program	Source program of mass-production object

xxxx	Host Machine	Operating System	Supply Media
5A13	PC-9800 Series	MS-DOS (Ver.3.30 to Ver.6.2 Note)	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatibles	See section **B.4**.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 Series 300	HP-UX (rel.7.05B)(QIC-24)	Cartridge tape
3P16	HP9000 Series 700	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS-4800 Series (RISC)	EWS-UX/V (rel.4.0)	

Note Although the task swap function is incorporated in MS-DOS Ver. 5.0 or later versions, this function cannot be used with the above software.

C.1 Real-Time OS (2/2)

MX78K0 operating system	This operating system is for μITRON specification subsets. A nucleus for MX78K0 is also included as a companion product. This software manages tasks, events, and time. In the task management, determining the task execution order and switching from a task to the next task are performed.
	Part number: μSxxxxMX78K0-ΔΔΔ

Remark The xxxx and ΔΔΔ in the part number differ depending on the host machine, operating system, and other factors as shown below.

μSxxxxMX78K0-ΔΔΔ

ΔΔΔ	Product Outline	Caution
001	Evaluation object	Use it in preproduction stages.
XX	Mass-production object	Use it in the mass production stage.
S01	Source program	Only the users who purchased mass-production objects are allowed to purchase this program.

xxxx	Host Machine	Operating System	Supply Media
5A13	PC-9800 Series	MS-DOS (Ver. 3.30 to Ver 6.2 Note)	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT and compatibles	See section **B.4**.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 Series 300	HP-UX (rel. 7.05B)	Cartridge tape (QIC-24)
3P16	HP9000 Series 700	HP-UX (rel. 9.01)	Digital audio tape (DAT)
3K15	SPARCstation	SunOS (rel. 4.1.1)	Cartridge tape
3M15	EWS4800 Series (RISC)	EWS-UX/V (rel. 4.0)	(QIC-24)

Note Although the task swap function is incorporated in MS-DOS Ver. 5.0 or later versions, this function cannot be used with the above software.

C.2 Fuzzy Inference Development Support System

FE9000/FE9200 fuzzy knowledge data preparation tool	This program supports input of fuzzy knowledge data (fuzzy rule and membership function), editing (edit), and evaluation (simulation). It runs on Windows.
	Part Number: μSxxxxFE9000 (PC-9800 Series) μSxxxxFE9200 (IBM PC/AT and compatibles)
FT9080/FT9085 translator	This program converts fuzzy knowledge data obtained by using fuzzy knowledge data preparation tool into the assembler source program for the RA78K/0
	Part Number: μSxxxxFT9080 (PC-9800 Series) μSxxxxFT9085 (IBM PC/AT and compatibles)
FI78K0 fuzzy inference module	This program executes fuzzy inference by being linked with fuzzy knowledge data converted by the translator.
	Part Number: μSxxxxFI78K0 (PC-9800 Series, IBM PC/AT and compatibles)
FD78K0 fuzzy inference debugger	This software supports developers who perform evaluation and adjustment of fuzzy knowledge data at hardware levels by using the in-circuit emulator.
	Part Number : μSxxxxFD78K0 (PC-9800 Series, IBM PC/AT and compatibles)

Remark The xxxx in the part number differs depending on the host machine, operating system, and other factors as shown below.

μSxxxxFE9000
μSxxxxFT9080
μSxxxxFI78K0
μSxxxxFD78K0

xxxx	Host Machine	Operating System	Supply Media
5A13	PC-9800 Series	MS-DOS (Ver. 3.30 to Ver 6.2 Note)	3.5-inch 2HD
5A10			5-inch 2HD

Note Although the task swap function is incorporated in MS-DOS Ver. 5.0 or later versions, this function cannot be used with the above software.

μSxxxxFE9200
μSxxxxFT9085
μSxxxxFI78K0
μSxxxxFD78K0

xxxx	Host Machine	Operating System	Supply Media
7B13	IBM PC/AT or compatibles	See section **B.4**.	3.5-inch 2HC
7B10			5-inch 2HC

APPENDIX D   REGISTER INDEX

D.1 Register Index (by Register Name)

D.2 Register Index (by Register Symbol)

★

APPENDIX E   REVISION HISTORY

The revision history of this document is listed below. “Location” indicates the chapter or page of this edition where the revision has been made.

Edition	Major Revision from Preceding Edition	Location
2nd edition	Change of all target devices from “under development” to “developed”	Throughout
	Addition of “Quality Grade”	**CHAPTER 1 OUTLINE**
	Change of Caution on inputting external clock	**CHAPTER 5 CLOCK GENERATOR**
	Correction of description on current consumption in STOP mode in “Clock Generator Options”	
	Addition of Caution on operation of the OVF0 flag	**CHAPTER 6 16-BIT TIMER/ EVENT COUNTER**
	Change of watchdog timer mode register format and addition of Note and Caution	**CHAPTER 9 WATCHDOG TIMER**
	Addition of Caution on selecting serial interface channel 0 operating mode	**CHAPTER 14 SERIAL INTERFACE CHANNEL 0**
	Addition of Caution on busy control when controlling interval time by using automatic data transmit/receive interval specify register (ADTI)	**CHAPTER 15 SERIAL INTERFACE CHANNEL 1**
	Correction of Caution on setting STOP mode	**CHAPTER 18 STANDBY FUNCTION**
	Addition of Caution on replacement from PROM version to mask ROM version	**CHAPTER 20 μPD78P048A**
	Change of display output current of FIP controller/driver for μPD78044F Subseries	**APPENDIX A DIFFERENCE AMONG μPD78044, 78044A, AND 78044F SUBSERIES**
	Addition of following products: IE-78000-R-A, IE-70000-98-IF-B, IE-70000-98N-IF, IE-70000-PC-IF-B, IE-78000-R-SV3, ID78K0	**APPENDIX B DEVELOPMENT TOOLS**

[MEMO]

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: +55-11-889-1689

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	❏	❏	❏	❏