To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

RENESAS

User's Manual

μPD178018A Subseries

8-bit Single-chip Microcontroller

μPD178004A
μPD178006A
μPD178016A
μPD178018A
μPD178P018A

Document No. U12742EJ1V0UM00 (1st edition)
Date Published October 1997 N

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

MS-DOS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
IBM DOS, PC/AT and PC DOS are trademarks of IBM Corporation.
HP9000 Series 300, HP9000 Series 700, and HP-UX are trademarks of Hewlett-Packard Company.
SPARCstation is a trademark of SPARC International, Inc.
SunOS is a trademark of Sun Microsystems, Inc.
Ethernet is a trademark of Xerox Corporation.
NEWS and NEWS-OS are trademarks of Sony Corporation.
OSF/Motif is a trademark of Open Software Foundation, Inc.
TRON is an abbreviation of The Realtime Operating system Nucleus.
ITRON is an abbreviation of Industrial TRON.

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

License not needed	:	μPD178P018AKK-T
The customer must judge the need for license	:	μPD178004AGC-xxx-3B9, 178006AGC-xxx-3B9, μPD178016AGC-xxx-3B9, 178018AGC-xxx-3B9, μPD178P018AGC-3B9

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

Purchase of NEC I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

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

PREFACE

Readers

This manual has been prepared for user engineers who want to understand the functions of the μPD178018A subseries and design and develop its application systems and programs.

Of the above members, the μPD178P018AKK-T should be used only for experiment or function evaluation, because it is not intended for use in equipment that will be mass-produced and require high reliability.

Purpose

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

Organization

The μPD178018A subseries manual is separated into two parts: this manual and the instruction edition (common to the 78K/0 series).

μPD178018A subseries User's Manual (This manual)	78K/0 series User's Manual Instruction
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupt	• Explanation of each instruction
• Other on-chip peripheral functions	

How to Read This Manual

Before reading this manual, you should have general knowledge of electric and logic circuits and microcomputers.

- ○ When you want to understand the functions in general:
  → Read this manual in the order of the contents.
- ○ To know the μPD178018A subseries instruction function in detail:
  → Refer to the **78K/0 series User's Manual -Instructions (U12326E)**
- ○ How to interpret the register format:
  → For the circled bit number, the bit name is defined as a reserved word in DF178018 and RA78K/0, and in CC78K/0, already defined in the header file named sfrbit.h.
- ○ To know the electrical specifications of the μPD178018A subseries:
  → Refer to separately available Data Sheet.
  μPD178004A, 178006A, 178016A, 178018A Data Sheet (U12641E)
  μPD178P018A Data Sheet (U12642E)

Legend

Data representation weight	:	High digits on the left and low digits on the right
Active low representations	:	$\overline{\times\times\times}$ (line over the pin and signal names)
Note	:	Description of **Note** in the text.
Caution	:	Information requiring particular attention
Remark	:	Additional explanatory material
Numeral representations	:	Binary ... ×××× or ××××B Decimal ... ×××× Hexadecimal ... ××××H

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

- **Related documents for μPD178018A subseries**

Document Name		Document No.	
		Japanese	English
μPD178004A, 178006A, 178016A, 178018A Data Sheet		U12641J	U12641E
μPD178P018A Data Sheet		U12642J	U12642E
μPD178018A Subseries User's Manual		U12742J	This manual
78K/0 Series User's Manual—Instruction		U12326J	U12326E
78K/0 Series Instruction Table		U10903J	—
78K/0 Series Instruction Set		U10904J	—
μPD178018A Subseries Special Function Register Table		Planned	—
78K/0 Series Application Note	Basics II	U10121J	U10121E

- **Development Tool Documents (User's Manuals)**

Document Name		Document Number	
		Japanese	English
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	U11789E	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
CC78K Series Library Source File		U12322J	–
PG-1500 PROM Programmer		U11940J	EEU-1335
PG-1500 Controller PC-9800 Series (MS-DOS™) Base		EEU-704	EEU-1291
PG-1500 Controller IBM PC Series (PC DOS™) Base		EEU-5008	U10540E
IE-78000-R		U11376J	U11376E
IE-78000-R-A		U10057J	U10057E
IE-78000-R-BK		EEU-867	EEU-1427
IE-178018-R-EM		U10668J	U10668E
EP-78230		EEU-985	EEU-1515
SM78K0 System Simulator Windows Base	Reference	U10181J	U10181E
SM78K Series System Simulator	External Components User Open Interface Specifications	U10092J	U10092E
ID78K/0 Integrated Debugger EWS Base	Reference	U11151J	–
ID78K0 Integrated Debugger PC Base	Reference	U11539J	U11539E
ID78K0 Integrated Debugger Windows Base	Guide	U11649J	U11649E
SD78K/0 Screen Debugger	Introduction	EEU-852	–
PC-9800 Series (MS-DOS) Base	Reference	U10952J	–
SD78K/0 Screen Debugger	Introduction	EEU-5024	U10539E
IBM PC/AT™ (PC DOS) Base	Reference	U11279J	U11279E

Caution The above documents are subject to change without prior notice. Be sure to use the latest version document when starting design.

- **Documents for Embedded Software (User's Manual)**

Document Name		Document No.	
		Japanese	English
78K/0 Series Real-Time OS	Basics	EEU-912	—
	Installation	EEU-911	—
78K/0 Series OS MX78K0	Basics	U12257J	—
Fuzzy Knowledge Data Creation Tool		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

- **Other Documents**

Document Name	Document No.	
	Japanese	English
IC Package Manual	C10943X	
Semiconductor Device Mounting Technology Manual	C10535J	C10535E
Quality Grade on NEC Semiconductor Devices	C11531J	C11531E
Reliability Quality Control on NEC Semiconductor Devices	C10983J	C10983E
Electric Static Discharge (ESD) Test	MEM-539	—
Semiconductor Devices Quality Assurance Guide	C11893J	MEI-1202
Microcomputer Related Product Guide—Third Party Manufacturers	U11416J	—

Caution The above documents are subject to change without prior notice. Be sure to use the latest version document when starting design.

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 OUTLINE

1.1 Features

- On-chip high-capacity ROM and RAM

Type / Part Number	Program Memory (ROM)	Data Memory		
		Internal High-Speed RAM	Buffer RAM	Internal Expansion RAM
μPD178004A	32 Kbytes	1024 bytes	32 bytes	None
μPD178006A	48 Kbytes			
μPD178016A				2048 bytes
μPD178018A	60 Kbytes			
μPD178P018A				

- Instruction set suitable for system control
  - Bit processing across entire address space
  - Multiplication/division instructions
- General-purpose I/O ports: 62 pins
- Hardware for PLL frequency synthesizer
  - Dual modulus prescaler (130 MHz MAX.)
  - Programmable divider
  - Phase comparator
  - Charge pump
- Frequency counter
- 8-bit resolution A/D converter: 6 channels
- Serial interface: clocked 2-channel
  - 3-wire serial I/O/SBI/2-wire serial I/O/I^2C bus**Note** mode : 1 channel
  - 3-wire serial I/O mode (with automatic transfer/receive function): 1 channel
- Timer: 5 channels
  - Basic timer (timer carry FF) : 1 channel
  - 8-bit timer/event counter : 2 channels
  - 8-bit timer (D/A converter: PWM output) : 1 channel
  - Watchdog timer : 1 channel
- Vectored interrupt source : 17 sources
  - Maskable interrupt source : 15 (internal: 8, external: 7)
  - Non-maskable interrupt source : 1
  - Software interrupt source : 1
- Test input : 1 pin
- Instruction cycle : 0.44 μs (with 4.5-MHz crystal resonator)
- Supply voltage : V_{DD} = 4.5 to 5.5 V (with PLL operating)
  V_{DD} = 3.5 to 5.5 V (with CPU operating, at CPU clock: f_X/2 or less)
  V_{DD} = 4.5 to 5.5 V (with CPU operating, at CPU clock: f_X)
- Power-ON clear circuit

Note When using the I^2C bus mode (including when this mode is implemented by software without using the internal hardware), consult NEC when you place an order for mask.

1.2 Applications

Car stereo, home stereo systems.

1.3 Ordering Information

Part Number	Package	Internal ROM
μPD178004AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Mask ROM
μPD178006AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Mask ROM
μPD178016AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Mask ROM
μPD178018AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Mask ROM
μPD178P018AGC-3B9**Note**	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	One-time PROM
μPD7178P018AKK-T**Note**	80-pin ceramic WQFN (14 × 14 mm, 0.65-mm pitch)	EPROM

Note Under development

Remark ××× indicates a ROM code suffix. When using I^2C bus mode, E×× is the ROM code suffix.

1.4 Quality Grade

Part Number	Package	Quality grade
μPD178004AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Standard
μPD178006AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Standard
μPD178016AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Standard
μPD178018AGC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Standard
μPD178P018AGC-3B9**Note**	80-pin plastic QFP (14 × 14 mm, 0.65-mm pitch)	Standard
μPD178P018AKK-T**Note**	80-pin ceramic WQFN (14 × 14 mm, 0.65-mm pitch)	Not assured (for function evaluation only)

Note Under development

Remark ××× indicates a ROM code suffix. When using I^2C bus mode, E×× is the ROM code suffix.

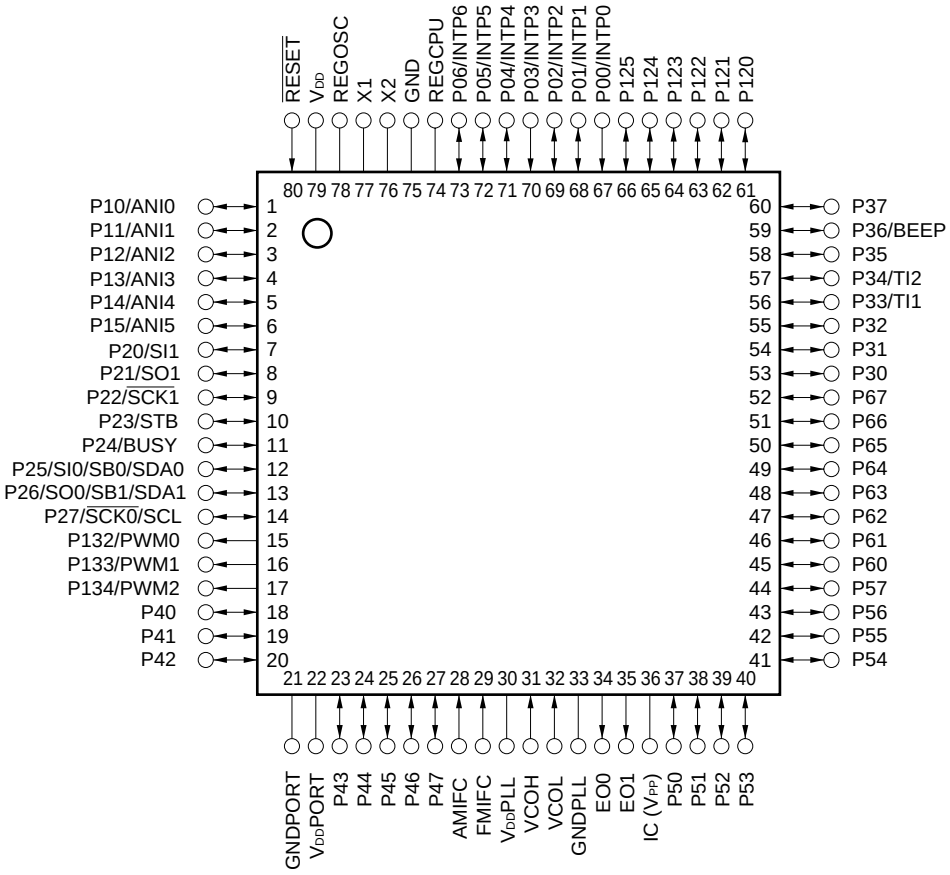
Of the above members, the μPD178P018AKK-T should be used only for experiment or function evaluation, because it is not intended for use in equipment that will be mass-produced and require high reliability.

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 × 14 mm)**
  μPD178004AGC-×××-3B9, 178006AGC-×××-3B9
  μPD178016AGC-×××-3B9, 178018AGC-×××-3B9, 178P018AGC-3B9**Note**
- **80-pin ceramic WQFN**
  μPD178P018AKK-T**Note**



Note Under development

Cautions 1. Be sure to connect IC (Internally Connected) pin to GND directly.
2. Set the potential of VDDPORT and VDDPLL pins to the same as that of VDD.
3. Set the potential of the GNDPORT and GNDPLL pins to the same as that of GND.
4. Connect each of the REGOSC and REGCPU pins to GND via a 0.1-μF capacitor.

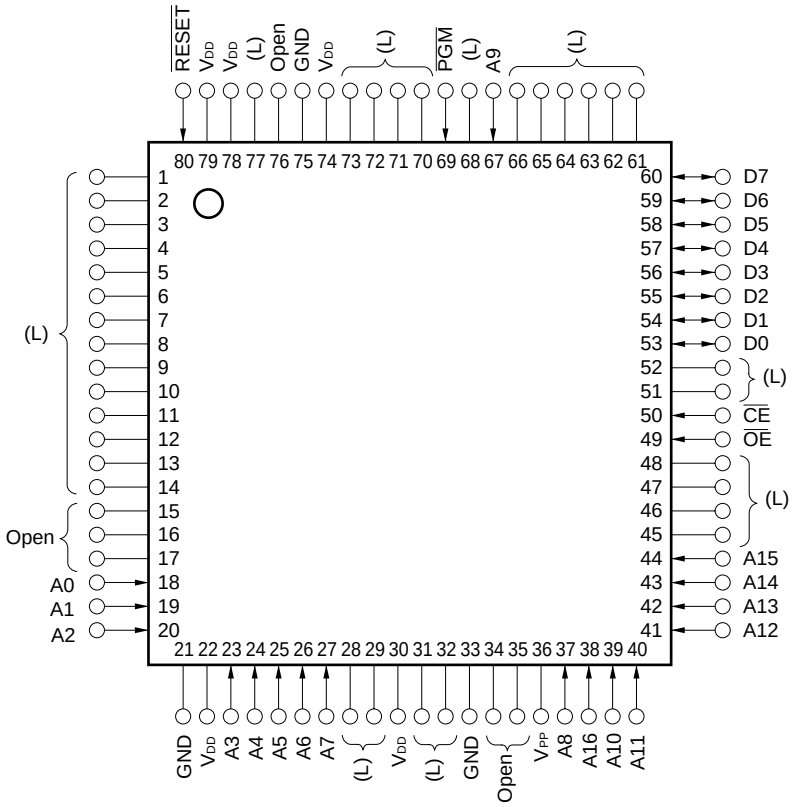
Remark Pin connection in parentheses is intended for the μPD178P018A.

Pin Name

AMIFC : AM intermediate frequency counter input
ANI0-ANI5 : A/D converter input
BEEP : Buzzer output
BUSY : Busy output
EO0, EO1 : Error out output
FMIFC : FM intermediate frequency counter input
GND : Ground
GNDPLL : PLL ground
GNDPORT : Port ground
IC : Internally connected
INTP0-INTP6 : Interrupt input
P00-P06 : Port 0
P10-P15 : Port 1
P20-P27 : Port 2
P30-P37 : Port 3
P40-P47 : Port 4
P50-P57 : Port 5
P60-P67 : Port 6
P120-P125 : Port 12
P132-P134 : Port 13
PWM0-PWM2 : PWM output
REGCPU : CPU power supply regulator
REGOSC : Oscillation circuit regulator
$\overline{\text{RESET}}$: Reset input
SB0, SB1 : Serial data bus I/O
$\overline{\text{SCK0}}$, $\overline{\text{SCK1}}$: Serial clock I/O
SCL : Serial clock I/O
SDA0, SDA1 : Serial data I/O
SI0, SI1 : Serial data input
SO0, SO1 : Serial data output
STB : Strobe output
TI1, TI2 : Timer clock input
VCOL, VCOH : Local oscillation input
V_{DD} : Power supply
V_{DD}PLL : PLL power supply
V_{DD}PORT : Port power supply
V_{PP} : Programming power supply
X1, X2 : Crystal resonator connection

(2) PROM programming mode

- **80-pin plastic QFP (14 × 14 mm)**
  μPD178P018AGC-3B9**Note**
- **80-pin ceramic WQFN**
  μPD178P018AKK-T**Note**

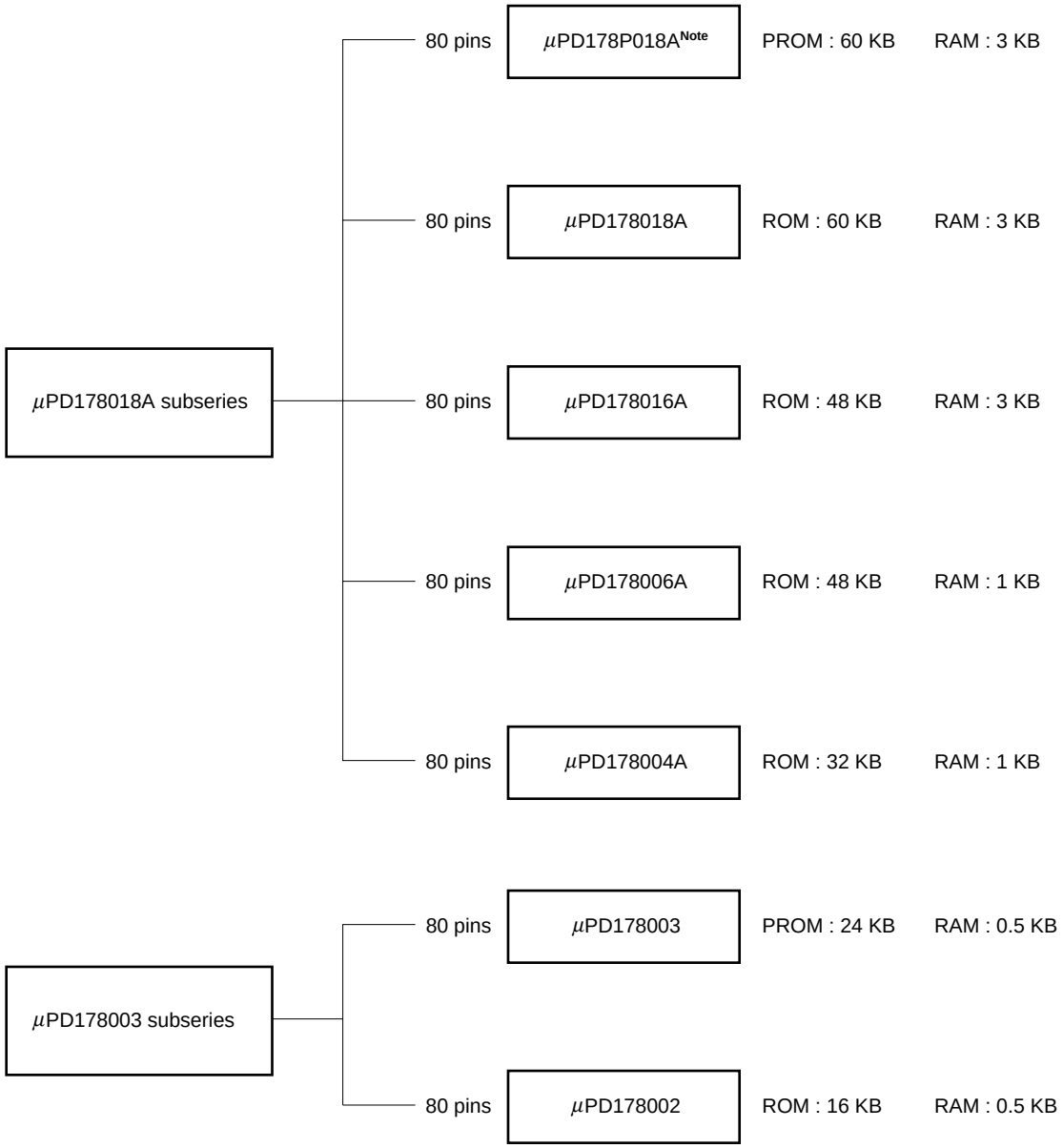


Note Under development

Cautions 1. (L) : Connect individually to GND via a pull-down resistor.
2. GND : Connect to the ground.
3. RESET : Set to the low level.
4. Open : Do not connect anything.

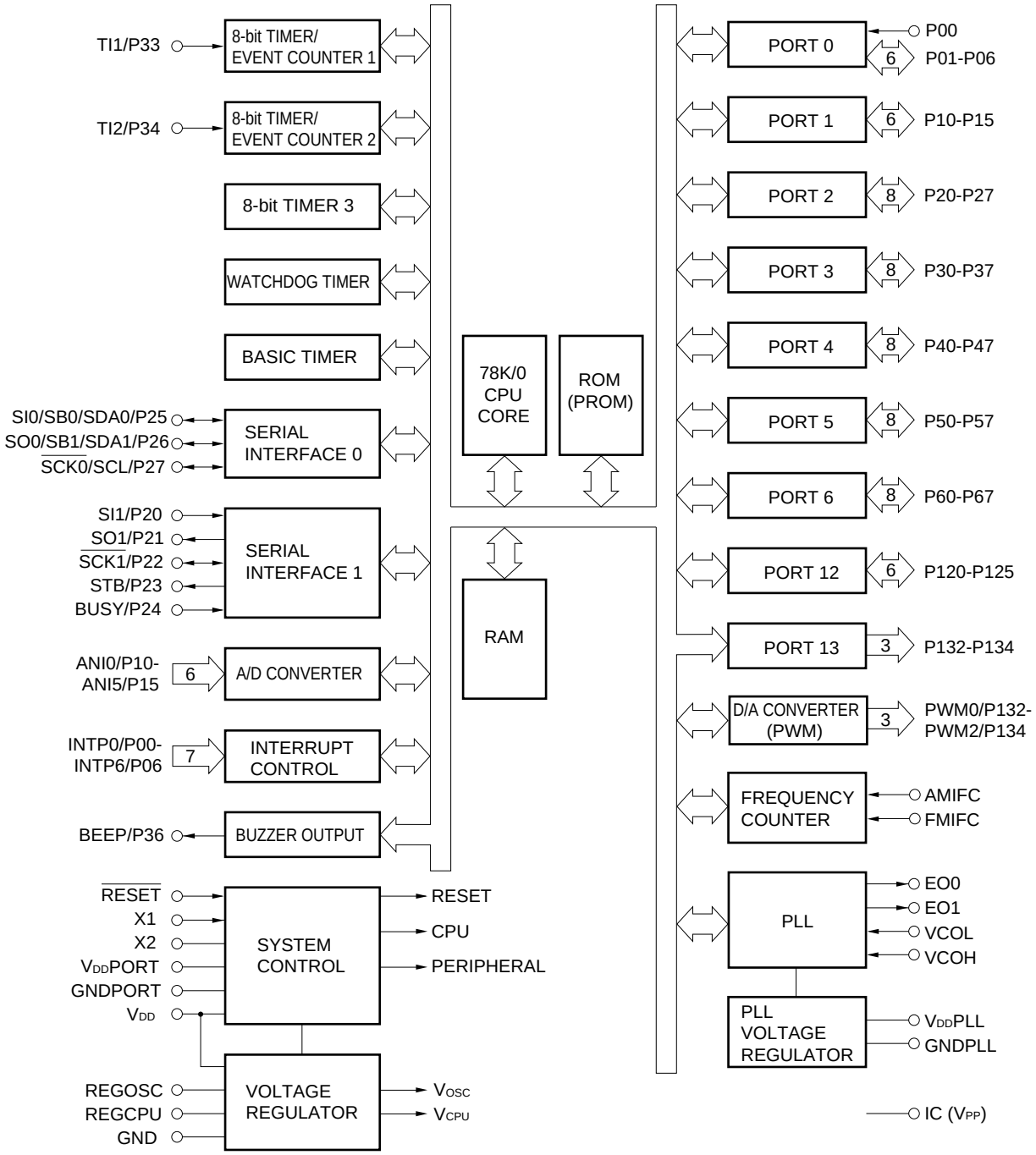
A0 to A16	: Address Bus	PGM	: Program
CE	: Chip Enable	RESET	: Reset
D0 to D7	: Data Bus	VDD	: Power Supply
GND	: Ground	VPP	: Programming Power Supply
OE	: Output Enable		

1.6 Development of μPD178018A Subseries and μPD178003 Subseries



Note Under development

1.7 Block Diagram



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

2. Pin connection in parentheses is intended for the μPD178P018A.

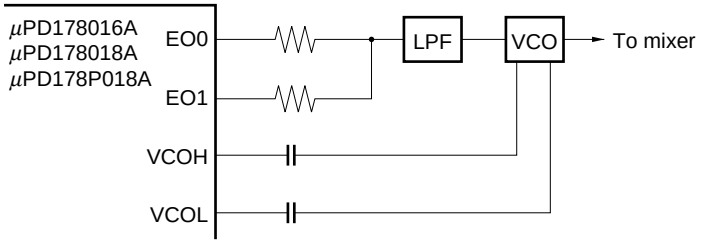
1.8 Outline of Function

Item \ Part Number		μPD178004A	μPD178006A	μPD178016A	μPD178018A	μPD178P018A
Internal memory	ROM (ROM structure)	32K bytes (mask ROM)	48K bytes (mask ROM)		60K bytes (mask ROM)	60K bytes (PROM)
	High-speed RAM	1024 bytes				
	Buffer RAM	32 bytes				
	Expansion RAM	None		2048 bytes		
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.44 μs/0.88 μs/1.78 μs/3.56 μs/7.11 μs/14.22 μs (with 4.5-MHz crystal resonator)				
Instruction set		• 16-bit operation • Multiplication/division (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulation (set, reset, test, Boolean operation) • BCD adjustment, etc.				
I/O port		Total : 62 pins CMOS input : 1 pin CMOS I/O : 54 pins N-ch open-drain I/O : 4 pins N-ch open-drain output : 3 pins				
A/D converter		8-bit resolution × 6 channels				
Serial interface		• 3-wire serial I/O/SBI/2-wire serial I/O/I^2C bus[Note] mode selectable: 1 channel • 3-wire serial I/O mode (with automatic transfer/receive function of up to 32 byte) : 1 channel				
Timer		• Basic timer (timer carry FF (10 Hz)) : 1 channel • 8-bit timer/event counter : 2 channels • 8-bit timer (D/A converter: PWM output): 1 channel • Watchdog timer : 1 channel				
Buzzer (BEEP) output		1.5 kHz, 3 kHz, 6 kHz				
Vectored interrupt source	Maskable	Internal: 8, external: 7				
	Non-maskable	Internal: 1				
	Software	Internal: 1				
Test input		Internal: 1				

Note When using the I^2C bus mode (including when this mode is implemented by software without using the internal hardware), consult NEC when you place an order for mask.

Item \ Part Number		μPD178004A	μPD178006A	μPD178016A	μPD178018A	μPD178P018A
PLL frequency synthesizer	Division mode	Two types • Direct division mode (VCOL pin) • Pulse swallow mode (VCOH and VCOL pins)				
	Reference frequency	7 types selectable by program (1, 3, 5, 9, 10, 25, 50 kHz)				
	Charge pump	Error out output: 2 (EO0, EO1 pins[Note 1])				
	Phase comparator	Unlock detectable by program				
Frequency counter		• Frequency measurement • AMIFC pin: for 450 kHz count • FMIFC pin: for 450 kHz/10.7 MHz counter				
D/A converter (PWM output)		8-/9-bit resolution × 3 channels (shared by 8-bit timer)				
Standby function		• HALT mode • STOP mode				
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Reset by power-ON clear circuit (3-value detection) • Detection of less than 4.5 V[Note 2] (CPU clock: f_X) • Detection of less than 3.5 V[Note 2] (CPU clock: f_X/2 or less and on power application) • Detection of less than 2.5 V[Note 2] (in STOP mode)				
Supply voltage		• V_{DD} = 4.5 to 5.5 V (with PLL operating) • V_{DD} = 3.5 to 5.5 V (with CPU operating, CPU clock: f_X/2 or less) • V_{DD} = 4.5 to 5.5 V (with CPU operating, CPU clock: f_X)				
Package		• 80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch) • 80-pin ceramic WQFN (14 × 14 mm, 0.65 mm pitch, μPD178P018A only)				

Notes 1. With the μPD178016A, 178018A and 178P018A the EO1 pin can be set to high-impedance. An application example is as follows:



LPF: Low pass filter VCO: Voltage control generator

- **To lock target frequency at high-speed**
  By setting EO0 and EO1 pins to error out output, the output current capability and voltage control capability of LPF are improved.
- **Steady state**
  By setting only EO0 pin to error out output, the stability of LPF is maintained.

2. For these values, refer to **CHAPTER 18 RESET FUNCTION**.

[MEMO]

CHAPTER 2 PIN FUNCTION

2.1 Pin Function List

2.1.1 Normal operating mode pins

(1) Port pins

Pin Name	I/O	Function		After Reset	Alternative Function
P00	Input	Port 0.	Input only	Input	INTP0
P01-P06	I/O	7-bit input/output port.	Input/output mode can be specified bit-wise.	Input	INTP1-INTP6
P10-P15	I/O	Port 1. 6-bit input/output port. Input/output mode can be specified bit-wise.		Input	ANI0-ANI5
P20	I/O	Port 2.		Input	SI1
P21		8-bit input/output port.			SO1
P22		Input/output mode can be specified bit-wise.			$\overline{\text{SCK1}}$
P23					STB
P24					BUSY
P25					SI0/SB0/SDA0
P26					SO0/SB1/SDA1
P27					$\overline{\text{SCK0}}$/SCL
P30-P32	I/O	Port 3.		Input	—
P33		8-bit input/output port.			TI1
P34		Input/output mode can be specified bit-wise.			TI2
P35					—
P36					BEEP
P37					—
P40-P47	I/O	Port 4. 8-bit input/output port. Input/output mode can be specified in 8-bit units. Test input flag (KRIF) is set to 1 by falling edge detection.		Input	—
P50-P57	I/O	Port 5. 8-bit input/output port. Input/output mode can be specified bit-wise.		Input	—
P60-P63	I/O	Port 6. 8-bit input/output port.	Middle voltage N-ch open drain input/output port.	Input	—
P64-P67		Input/output mode can be specified bit-wise.	LEDs can be driven directly.		
P120-P125	I/O	Port 12. 6-bit input/output port. Input/output mode can be specified bit-wise.		Input	—
P132-P134	Output	Port 13. 3-bit output port. N-ch open-drain output port.		—	PWM0-PWM2

(2) Pins other than port pins

Pin Name	I/O	Function	After Reset	Alternative Function
INTP0-INTP6	Input	External maskable interrupt request inputs with specifiable valid edges (rising edge, falling edge, both rising and falling edges).	Input	P00-P06
SI0	Input	Serial interface serial data input	Input	P25/SB0/SDA0
SI1				P20
SO0	Output	Serial interface serial data output	Input	P26/SB1/SDA1
SO1				P21
SB0	I/O	Serial interface serial data input/output	Input	P25/SI0/SDA0
SB1				P26/SO0/SDA1
SDA0				P25/SI0/SB0
SDA1				P26/SO0/SB1
$\overline{SCK0}$	I/O	Serial interface serial clock input/output	Input	P27/SCL
$\overline{SCK1}$				P22
SCL				P27/$\overline{SCK0}$
STB	Output	Serial interface automatic transmit/receive strobe output	Input	P23
BUSY	Input	Serial interface automatic transmit/receive busy input	Input	P24
TI1	Input	External count clock input to 8-bit timer (TM1)	Input	P33
TI2		External count clock input to 8-bit timer (TM2)		P34
BEEP	Output	Buzzer output	Input	P36
ANI0-ANI5	Input	A/D converter analog input	Input	P10-P15
PWM0-PWM2	Output	PWM output	—	P132-P134
EO0, EO1	Output	Error out output from charge pump of the PLL frequency synthesizer	—	—
VCOL	Input	Inputs PLL local band frequency (In HF, MF mode)	—	—
VCOH	Input	Inputs PLL local band frequency (In VHF mode)	—	—
AMIFC	Input	Inputs AM intermediate frequency counter	—	—
FMIFC	Input	Inputs FM intermediate frequency counter	—	—
$\overline{RESET}$	Input	System reset input	—	—
X1	Input	System clock oscillation resonator connection	—	—
X2	—		—	—
REGOSC	—	Oscillation regulator. Connect to GND via a 0.1 μF capacitor.	—	—
REGCPU	—	CPU power supply regulator. Connect to GND via a 0.1 μF capacitor.	—	—
V_{DD}	—	Positive power supply	—	—
GND	—	Ground	—	—
V_{DD}PORT	—	Positive power supply for port block	—	—
GNDPORT	—	Ground for port block	—	—
V_{DD}PLL[Note]	—	Positive power supply for PLL	—	—
GNDPLL[Note]	—	Ground for PLL	—	—
IC	—	Internally connected. Connect to GND or GNDPORT directly.	—	—

Note Connect 1000 pF capacitor between V_{DD}PLL pin and GNDPLL pin.

2.1.2 PROM programming mode pins (PROM versions 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-A16	Input	Address bus
D0-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
GND	—	Ground potential

2.2 Description of Pin Functions

2.2.1 P00 to P06 (Port 0)

These are 7-bit input/output ports. Besides serving as input/output ports, they function as an external interrupt request input. The following operating modes can be specified bit-wise.

(1) Port mode

P00 functions as input-only port and P01 to P06 function as input/output ports.

P01 to P06 can be specified for input or output ports bit-wise with a port mode register 0 (PM0).

(2) Control mode

In this mode, these ports function as an external interrupt request input pin (INTP0-INTP6).

INTP0 to INTP6 are external interrupt request input pins which can specify valid edges (rising edge, falling edge, and both rising and falling edges).

2.2.2 P10 to P15 (Port 1)

These are 6-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 6-bit input/output ports.

They can be specified bit-wise as input or output ports with a port mode register 1 (PM1).

(2) Control mode

These ports function as A/D converter analog input pins (ANI0-ANI5).

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 port mode register 2 (PM2).

(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, SDA0[Note], SDA1[Note]

Serial interface serial data input/output pins

(b) $\overline{\text{SCK0}}$ and $\overline{\text{SCK1}}$, SCL[Note]

Serial interface serial clock input/output pins

Note For I^2C bus mode

(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 When this port is used as a serial interface, the I/O and output latches must be set according to the function the user requires. For the setting, refer to Figure 12-5 Serial Operating Mode Register 0 Format and Figure 13-3 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 and buzzer output.

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).

(2) Control mode

These ports function as timer input and buzzer output (BEEP).

(a) TI1 and TI2

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

(b) BEEP

Buzzer (BEEP) output pin.

2.2.5 P40 to P47 (Port 4)

These are 8-bit input/output ports.

The test input flag (KRIF) can be set to 1 by detecting a falling edge.

They can be specified in 8-bit units for input or output ports by using the port mode register 4 (MM).

2.2.6 P50 to P57 (Port 5)

These are 8-bit input/output ports.

Port 5 can drive LEDs directly.

They can be specified bit-wise as input/output ports with port mode register 5 (PM5).

2.2.7 P60 to P67 (Port 6)

These are 8-bit input/output ports. P60 to P63 can drive LEDs directly.

They can be specified bit-wise as input or output ports with port mode register 6 (PM6).

P60 to P63 are N-ch open drain outputs.

2.2.8 P120 to P125 (Port 12)

These are 6-bit input/output ports.

They can be specified bit-wise as input or output ports with port mode register 12 (PM12).

2.2.9 P132 to P134 (Port 13)

These are 3-bit output port. Besides serving as output port, they are used for PWM output.

The following operating modes can be specified bit-wise.

(1) Port mode

These ports function as 3-bit output port.

(2) Control mode

These ports function as PWM output (PWM0-PWM2).

2.2.10 EO0, EO1

These are the output pins of the charge pump of the PLL frequency synthesizer.

They output the result of phase comparison between the frequency divided by the programmable divider of the local oscillation input (VCOL and VCOH pins) and the reference frequency.

With the μPD178016A, 178018A, or 178P018A, the EO1 pin can be set to high-impedance.

2.2.11 VCOL, VCOH

These pins input the local oscillation frequency (VCO) of the PLL.

Because signals are input to these pins via an AC amplifier, cut the DC component of the input signals by using a capacitor.

- VCOL
  - HF, MF input
  - Becomes active when the HF or AM mode is selected by program; otherwise, goes into a high-impedance state.
- VCOH
  - VHF input
  - Becomes active when the FM mode is selected by program; otherwise, goes into a high-impedance state.

2.2.12 AMIFC

This is the input pin of the AM intermediate frequency counter.

2.2.13 FMIFC

This is the input pin of the FM intermediate frequency counter.

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

This is a low-level active system reset input pin.

2.2.15 X1 and X2

Crystal resonator connect pins for system clock oscillation.

2.2.16 REGOSC

Oscillation circuit regulator pin. Connect to GND via a 0.1-μF capacitor.

2.2.17 REGCPU

CPU power supply regulator pin. Connect to GND via a 0.1-μF capacitor.

2.2.18 V_{DD}

Positive power supply pin.

2.2.19 GND

Ground potential pin

2.2.20 V_{DD}PORT

Positive power supply pin for port. Set the same potential as that of V_{DD} pin.

2.2.21 GNDPORT

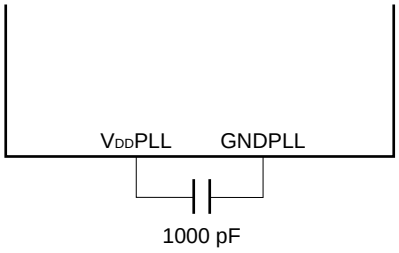
Ground potential pin for port. Set the same potential as that of GND pin.

2.2.22 V_{DD}PLL

Positive power supply pin for PLL. Connect 1000 pF capacitor between V_{DD}PLL pin and GNDPLL pin.

2.2.23 GNDPLL

Ground potential pin for PLL. Connect 1000 pF capacitor between GNDPLL pin and V_{DD}PLL pin.



2.2.24 V_{PP} (PROM versions only)

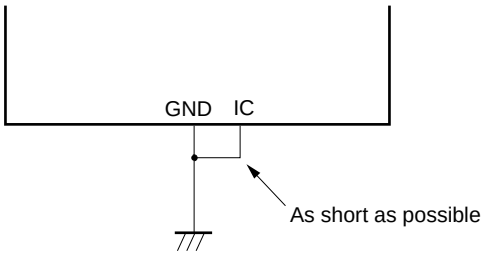
High-voltage apply pin for PROM programming mode setting and program write/verify. Connect directly to GND in normal operating mode.

2.2.25 IC (Mask ROM version only)

The IC (Internally Connected) pin is provided to set the test mode to check the μPD178018A subseries at delivery. Connect it directly to the GND pin with the shortest possible wire in the normal operating mode.

When a potential difference is produced between the IC pin and GND pin because the wiring between those two pins is too long or an external noise is input to the IC pin, the user's program may not run normally.

○ **Connect IC pin to GND pin directly.**



2.3 Input/output Circuits and Recommended Connection of Unused Pins

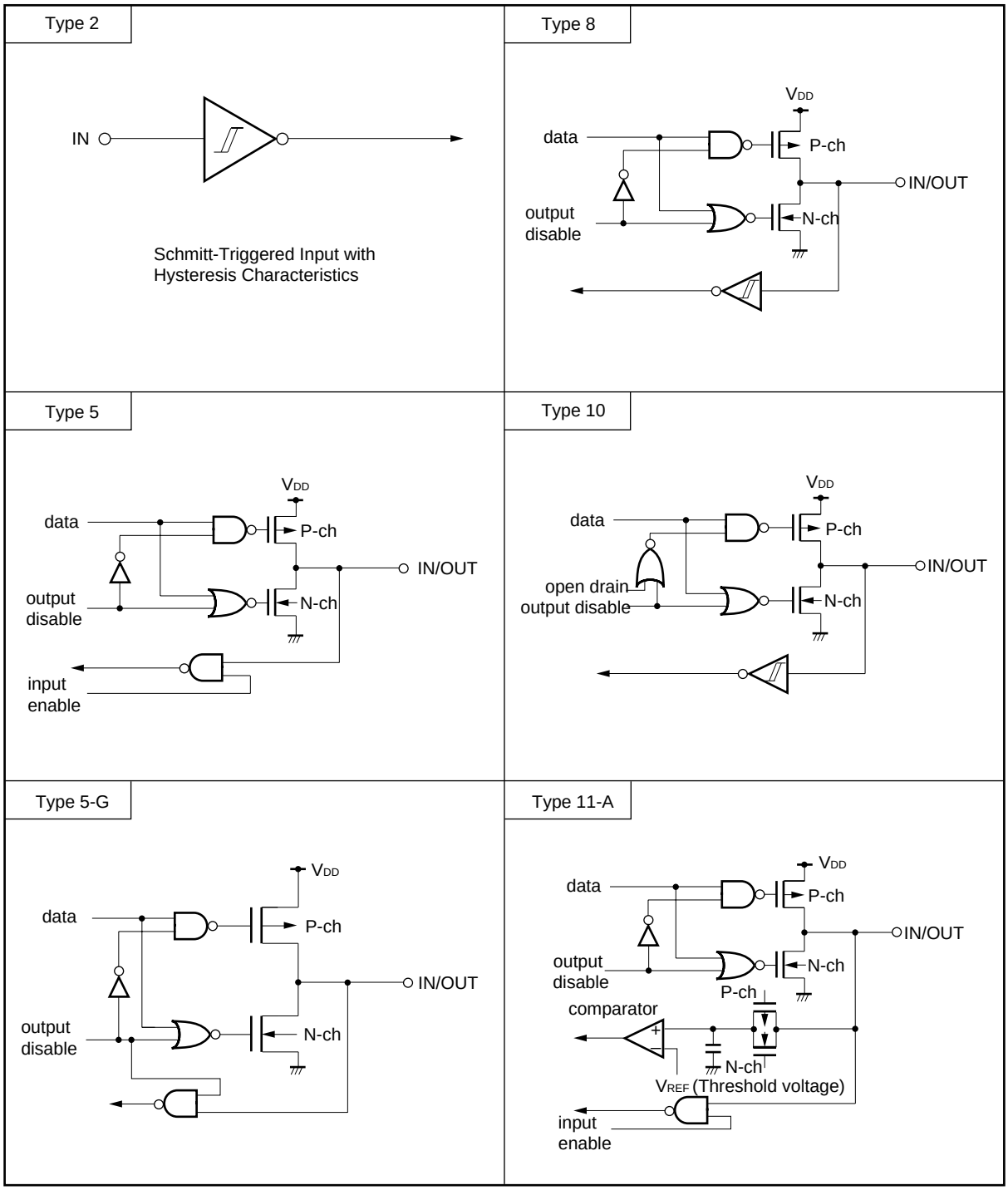
Table 2-1 shows the input/output circuit types of pins and the recommended conditions for unused pins. Refer to Figure 2-1 for the configuration of the input/output circuit of each type.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connections of Unused Pins
P00/INTP0	2	Input	Connected to GND or GNDPORT
P01/INTP1-P06/INTP6	8	I/O	Set in general-purpose input port mode by software and individually connected to V_{DD}, V_{DD}PORT, GND, or GNDPORT via resistor.
P10/ANI0-P15/ANI5	11-A		
P20/SI1	8		
P21/SO1	5		
P22/$\overline{\text{SCK1}}$	8		
P23/STB	5		
P24/BUSY	8		
P25/SI0/SB0/SDA0 P26/SO0/SB1/SDA1 P27/$\overline{\text{SCK0}}$/SCL	10		
P30-P32	5		
P33/TI1, P34/TI2	8		
P35 P36/BEEP P37	5		
P40-P47	5-G		
P50-P57	5		
P60-P63	13-D		
P64-P67	5		
P120-P125			
P132/PWM0-P134/PWM2	19	Output	Set to low-level output by software and open
EO0	DTS-EO1		Open
EO1	DTS-EO3[Note]		
VCOL, VCOH	DTS-AMP	Input	Set to disabled status by software and open
AMIFC, FMIFC			
IC (mask ROM model)	–	–	Connect to GND or GNDPORT directly
V_{PP} (PROM model)			

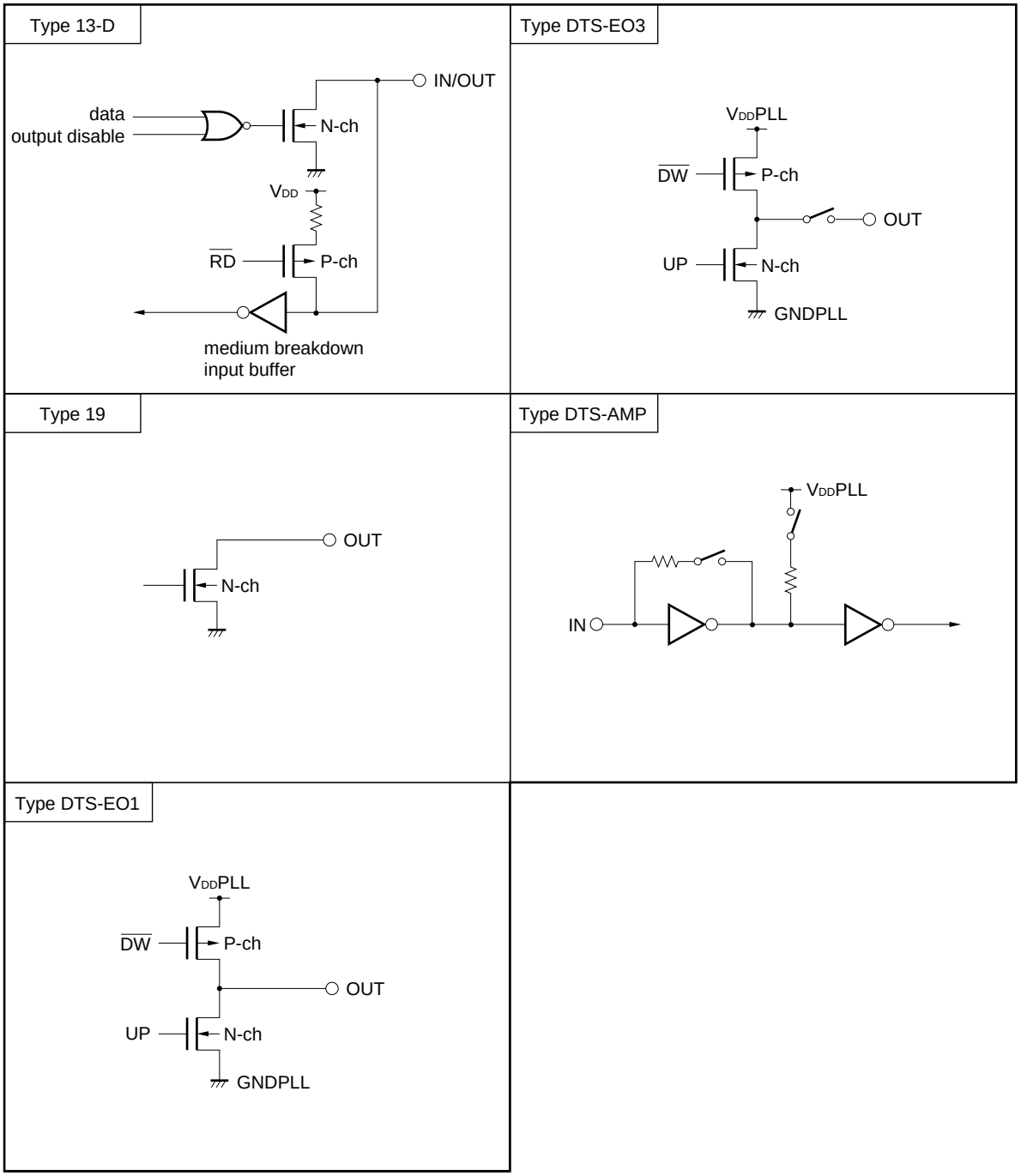
Note DTS-EO1 type for the μPD178004A and 178006A.

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



Remark All V_{DD} and GND in the above figures are the positive power supply and ground potential of the ports, and should be taken as V_{DD}PORT and GNDPORT, respectively.

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



Remark All V_{DD} and GND in the above figures are the positive power supply and ground potential of the ports, and should be taken as V_{DD}PORT and GNDPORT, respectively.

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Spaces

Figures 3-1 to 3-5 shows memory maps.

Figure 3-1. Memory Map (μPD178004A)

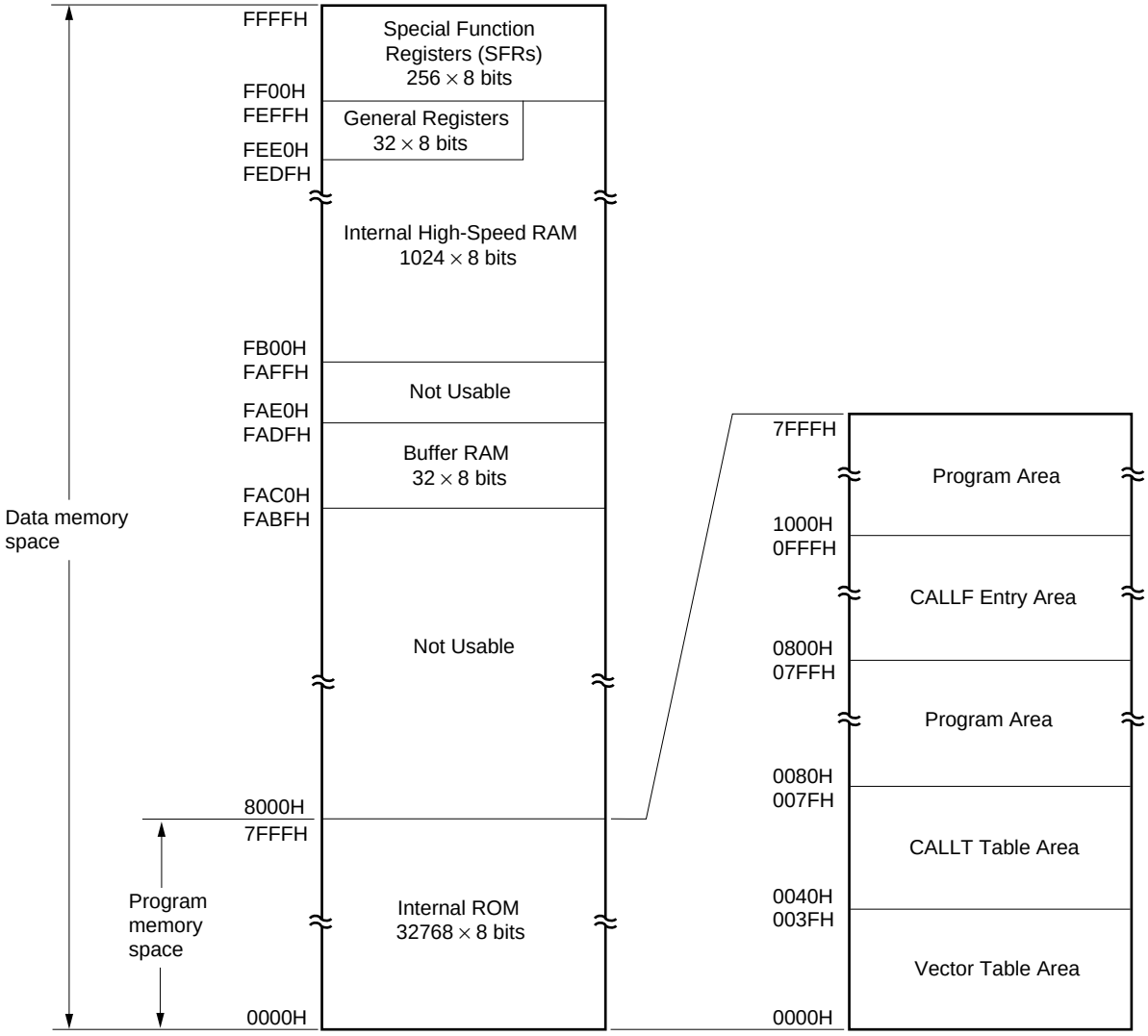


Figure 3-2. Memory Map (μPD178006A)

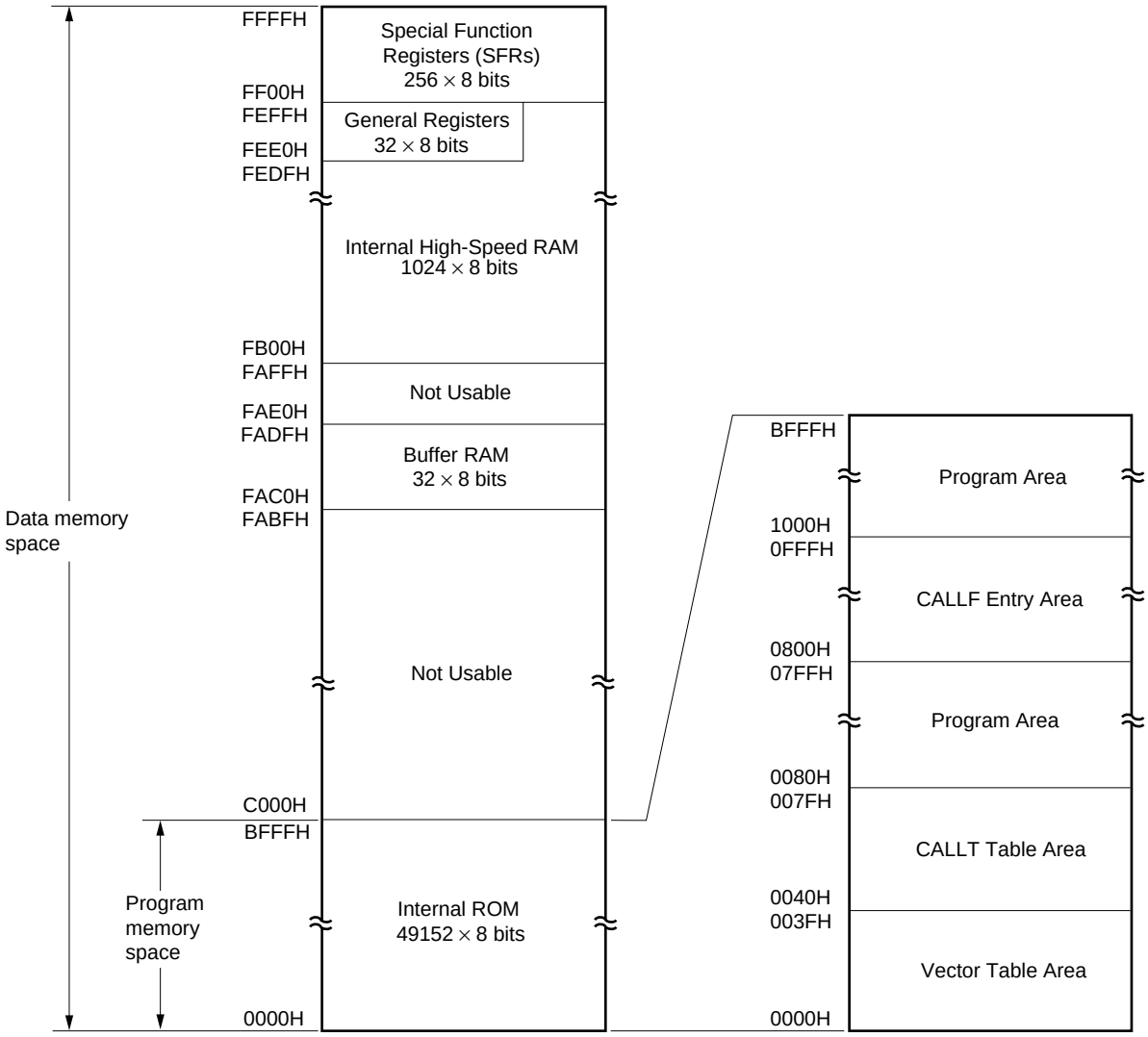


Figure 3-3. Memory Map (μPD178016A)

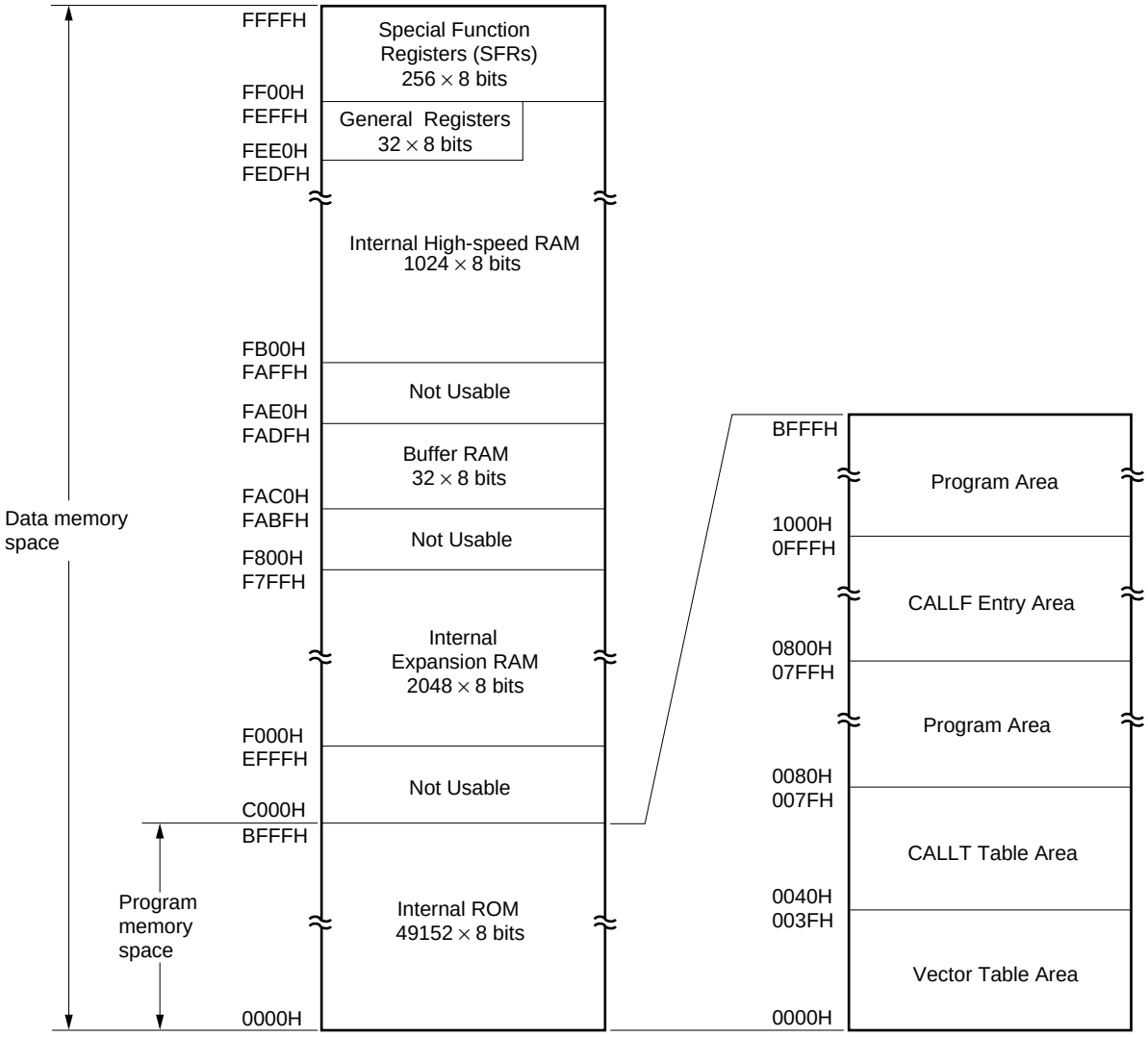


Figure 3-4. Memory Map (μPD178018A)

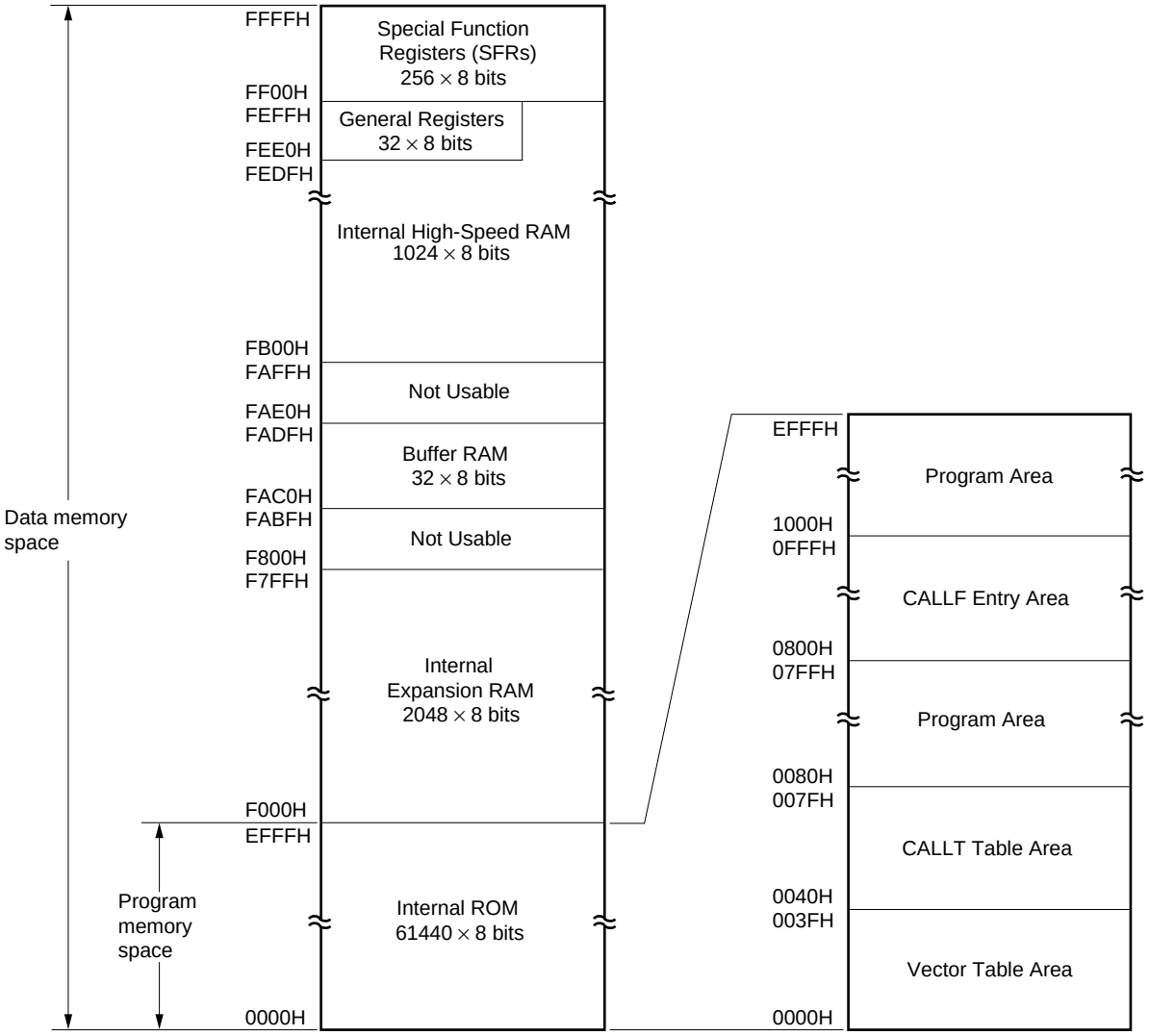
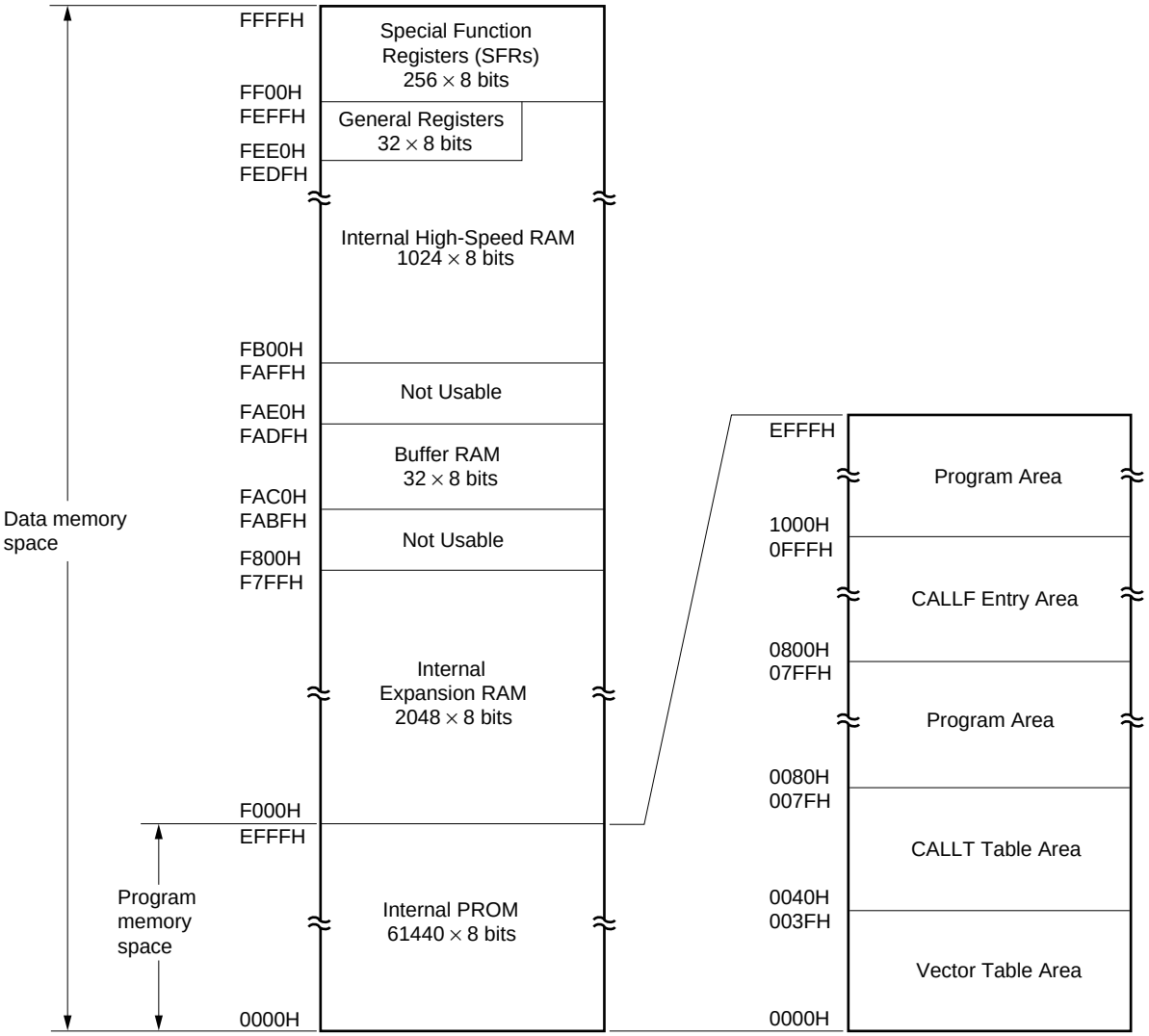


Figure 3-5. Memory Map (μPD178P018A)



3.1.1 Internal program memory space

The μPD178018A subseries has various sizes of internal ROM or PROM as shown below.

The internal program memory space stores programs and table data. Normally, they are addressed with a program counter (PC).

Part Number	Internal ROM	
	Type	Capacity
μPD178004A	Mask ROM	32768 × 8 bits
μPD178006A		49152 × 8 bits
μPD178016A		49152 × 8 bits
μPD178018A		61440 × 8 bits
μPD178P018A	PROM	61440 × 8 bits

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a 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, low-order 8 bits are stored at even addresses and high-order 8 bits are stored at odd addresses.

Table 3-1. Vector Table

Vector Table Address	Interrupt Source
0000H	Reset input
0004H	INTWDT
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
000EH	INTP4
0010H	INTP5
0012H	INTP6
0014H	INTCSI0
0016H	INTCSI1
0018H	INTTMC
001AH	INTPWM
001CH	INTTM1
001EH	INTTM2
0020H	INTAD
003EH	BRK

(2) CALLT instruction table area

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

(3) 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 μPD178018A subseries units incorporate the following RAMs.

(1) Internal high-speed RAM

High-speed memory of the 1024 × 8 bits is incorporated.

In this area, four 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 area.

(2) Buffer RAM

Buffer RAM is allocated to the 32-byte area from FAC0H to FADFH. Buffer RAM is used to store transmit/receive data for serial interface channel 1 (3-wire serial I/O mode with automatic transmit/receive function). When not used in this mode, buffer RAM can also be used as normal RAM.

(3) Internal expansion RAM (μPD178016A, 178018A, 178P018A only)

Internal expansion RAM is allocated to the 2048-byte area from F000H to F7FFH.

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 **3.2.3 Special-Function Register (SFR) Table 3-2 Special-Funtion Register List**).

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

3.1.4 Data memory addressing

The method of specifying the address of the instruction to be executed next or the address of the register or memory location 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 the memory location to be manipulated when an instruction is executed, the μPD178018A subseries offers variety of addressing modes to provide good operability. In particular, at addresses corresponding to data memory area (FB00H to FFFFH), particular addressing modes are possible to meet the functions of the special function registers (SFRs) and general registers. Figures 3-6 to 3-10 show the data memory addressing modes. For details of the addressing modes, refer to **3.4 Operand Address Addressing.**

Figure 3-6. Data Memory Addressing (μPD178004A)

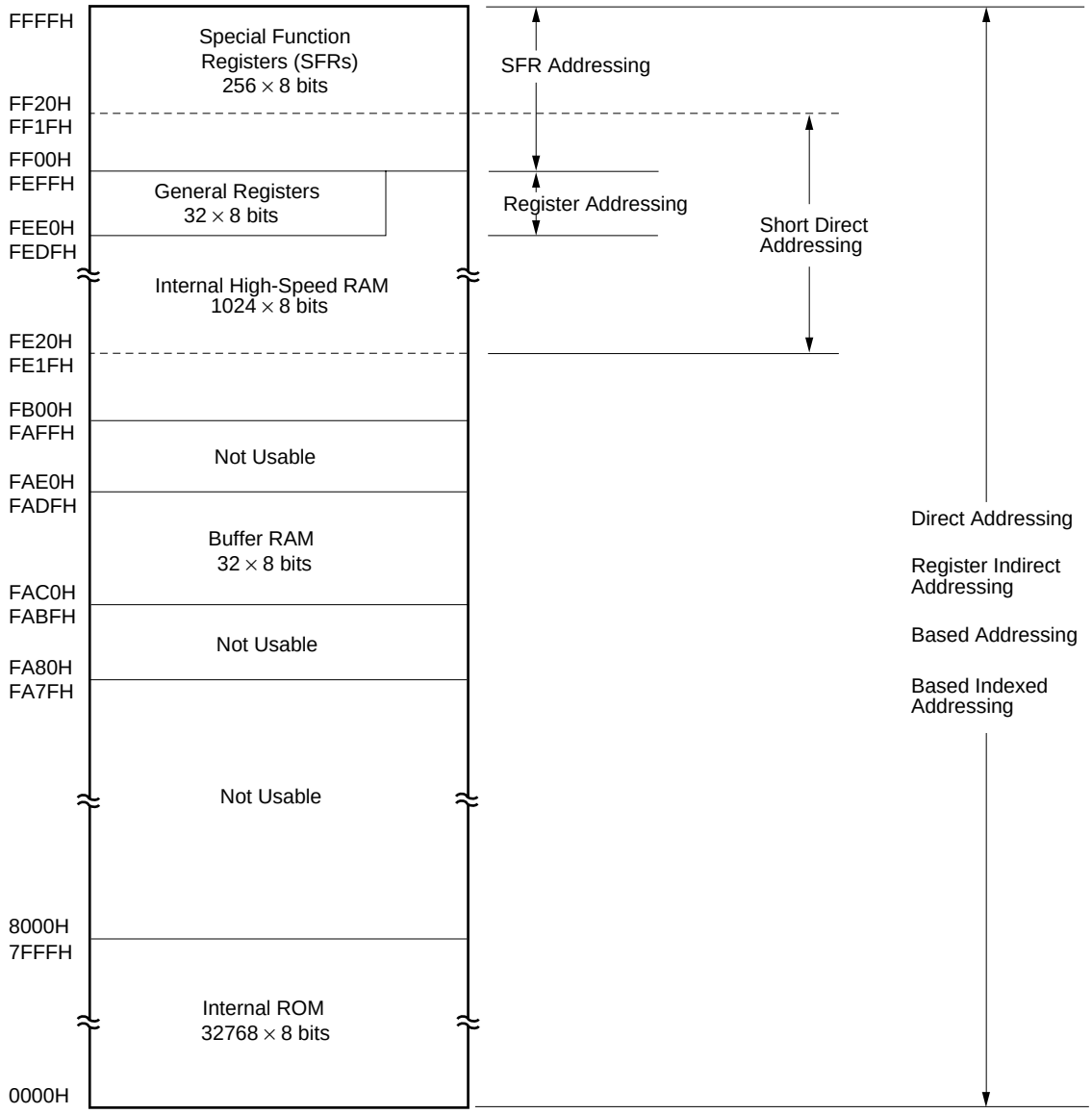


Figure 3-7. Data Memory Addressing (μPD178006A)

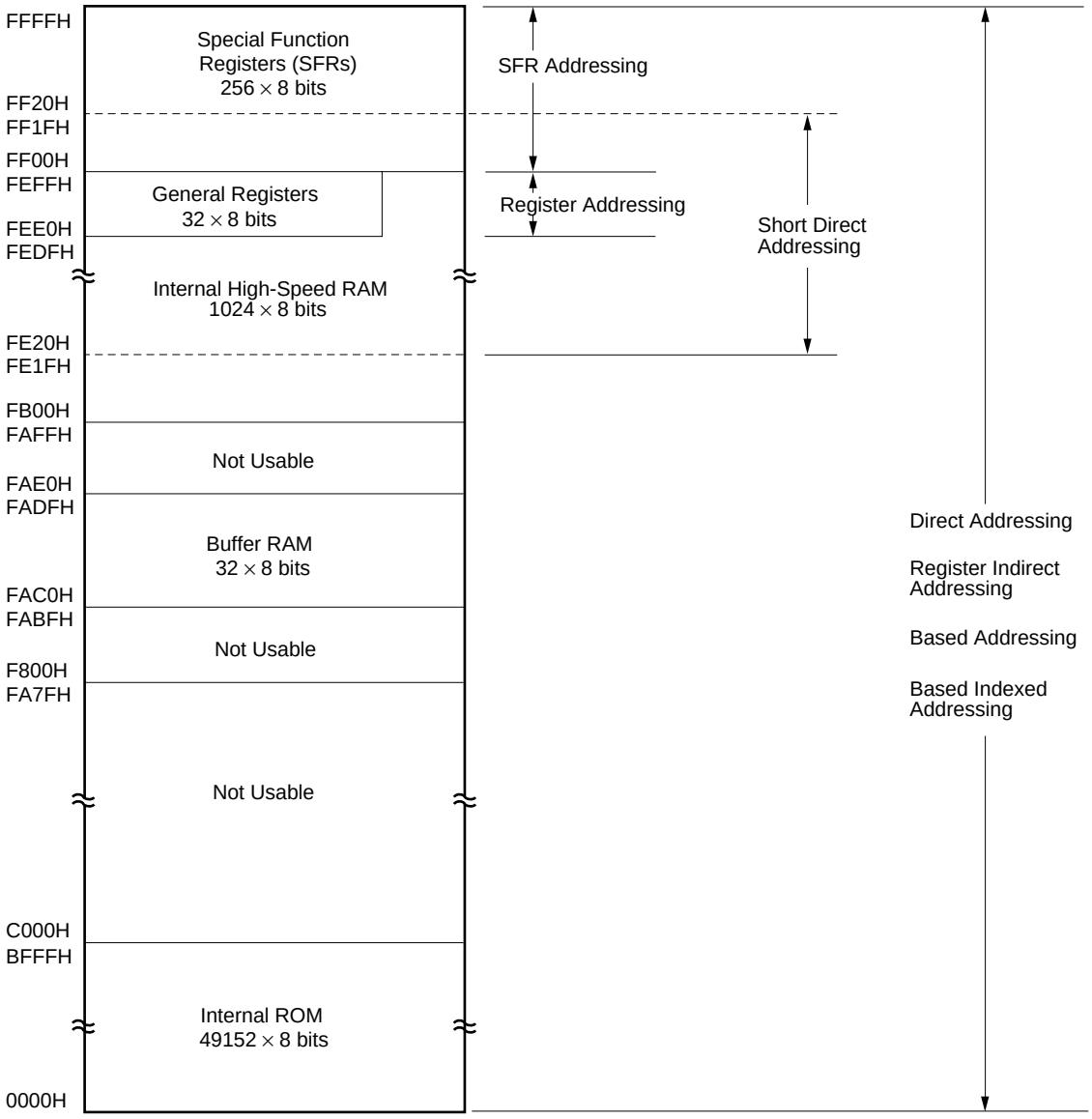


Figure 3-8. Data Memory Addressing (μPD178016A)

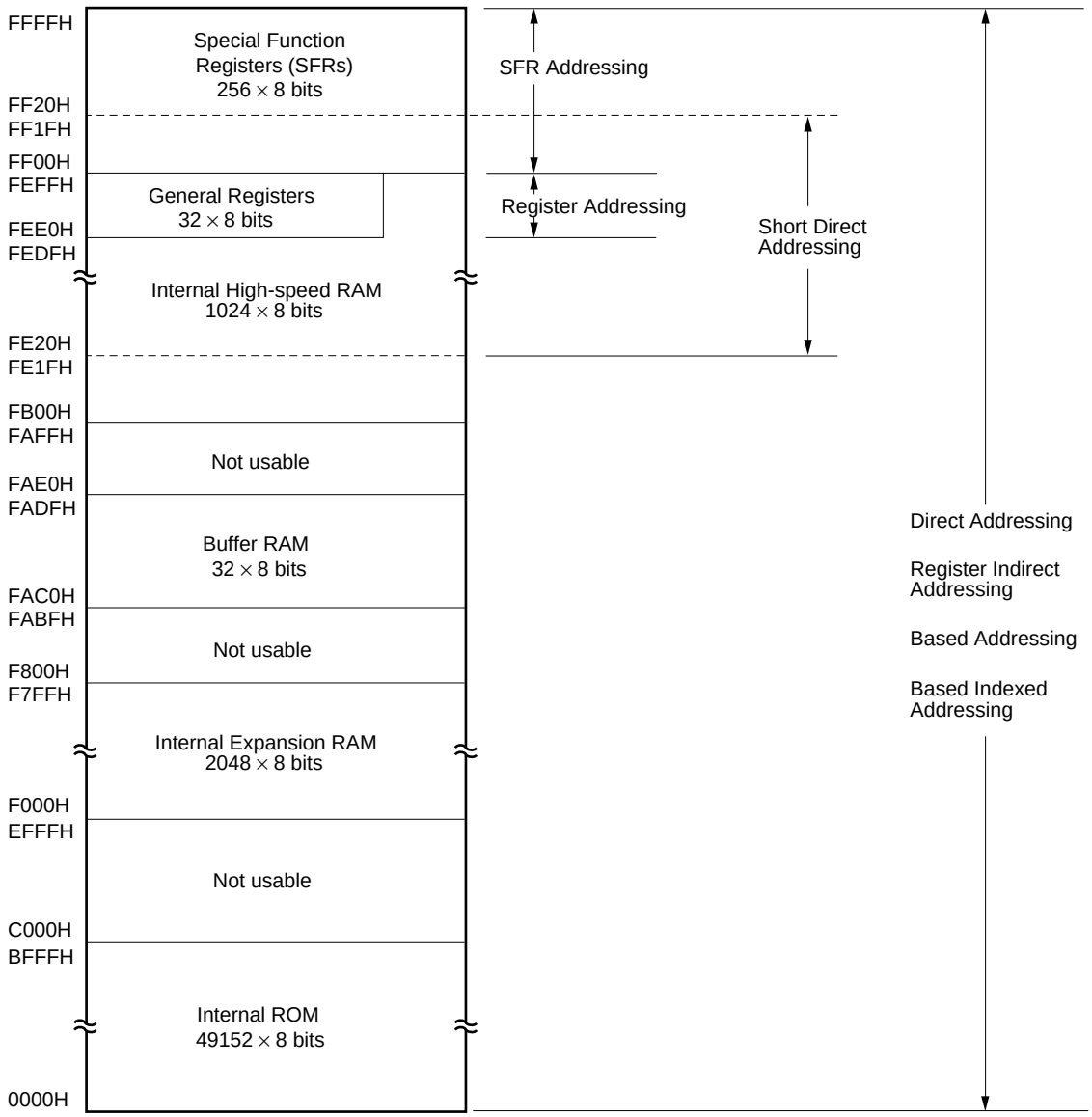


Figure 3-9. Data Memory Addressing (μPD178018A)

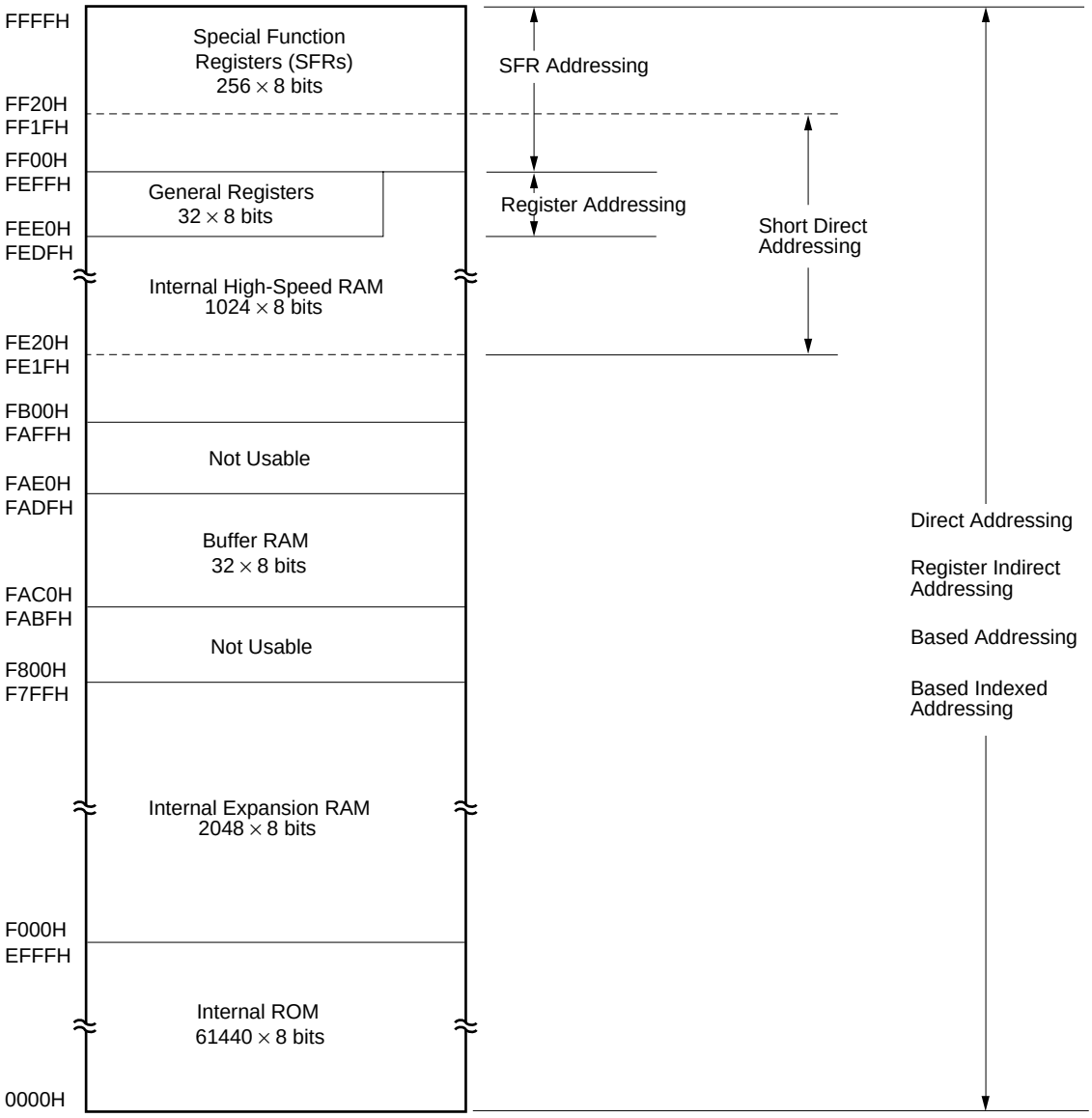
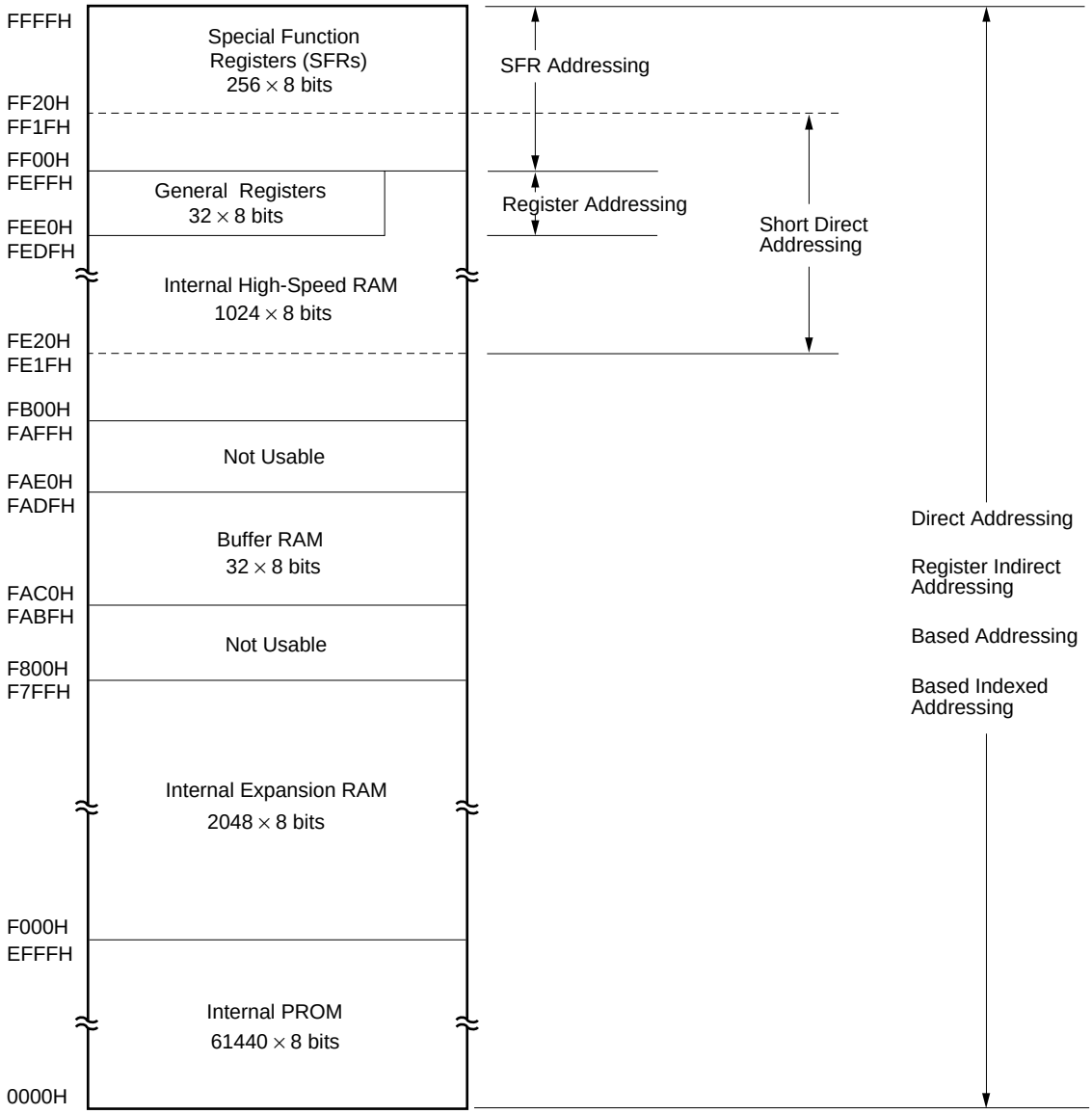


Figure 3-10. Data Memory Addressing (μPD178P018A)



3.2 Processor Registers

The μPD178018A subseries units incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 16-bit register 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.

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 restored upon execution of the RETB, RETI and POP PSW instructions.

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 the interrupt request acknowledge operations of the CPU.

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

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

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

(b) Zero flag (Z)

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

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

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

In these flags, the 2-bit information 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 (1). It is reset (0) in all other cases.

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When ISP = 0, acknowledging the vectored interrupt requests to which a low priority is assigned by the priority specification flag registers (PR0L, PR0H) (refer to **14.3 (3) Priority specification flag registers (PR0L, PR0H)**) is disabled. Whether an interrupt request is actually accepted depends on the status of 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 (FB00H-FEFFH) can be set as the stack area.

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 (restored) from the stack memory.

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

Caution Since reset input makes SP contents undefined, be sure to initialize the SP before instruction execution.

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

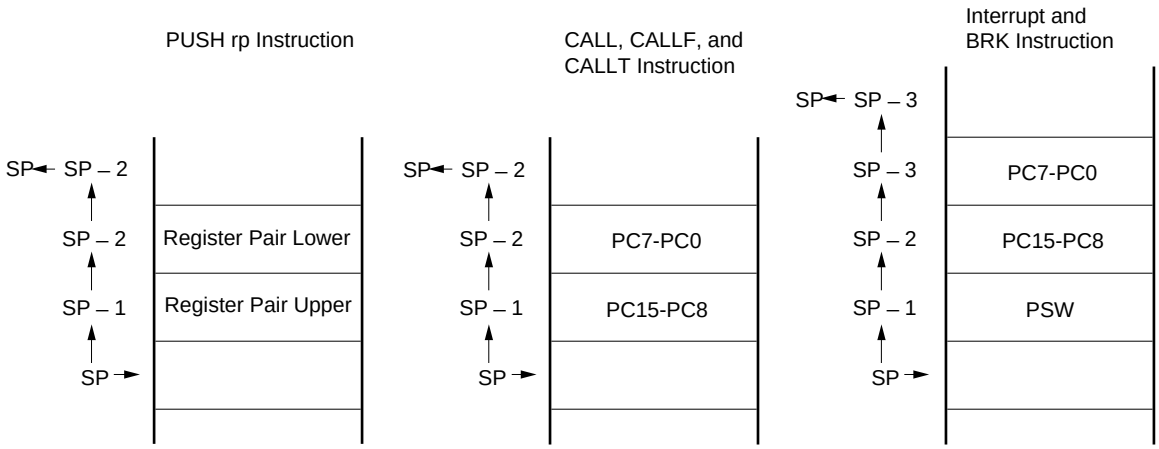
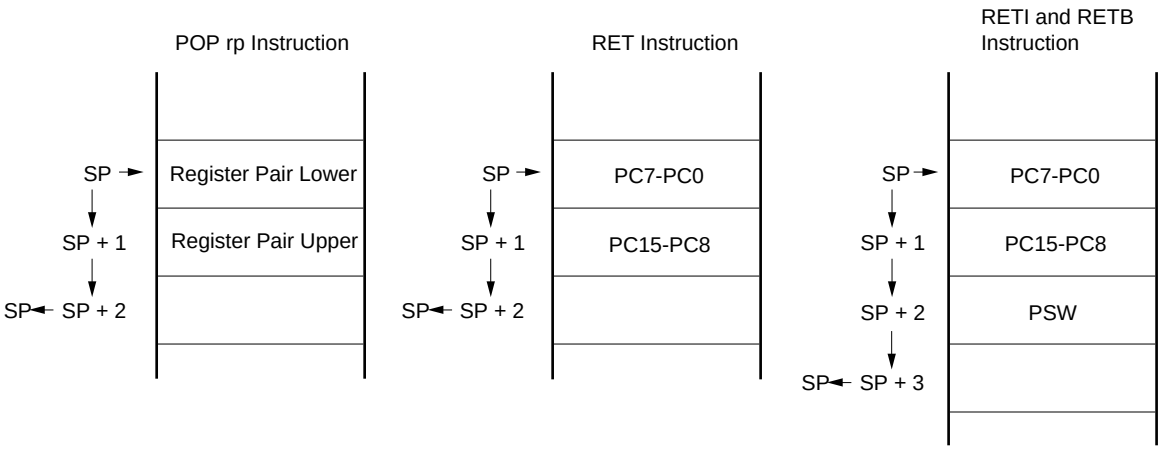


Figure 3-15. Data to be Restored from Stack Memory



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).

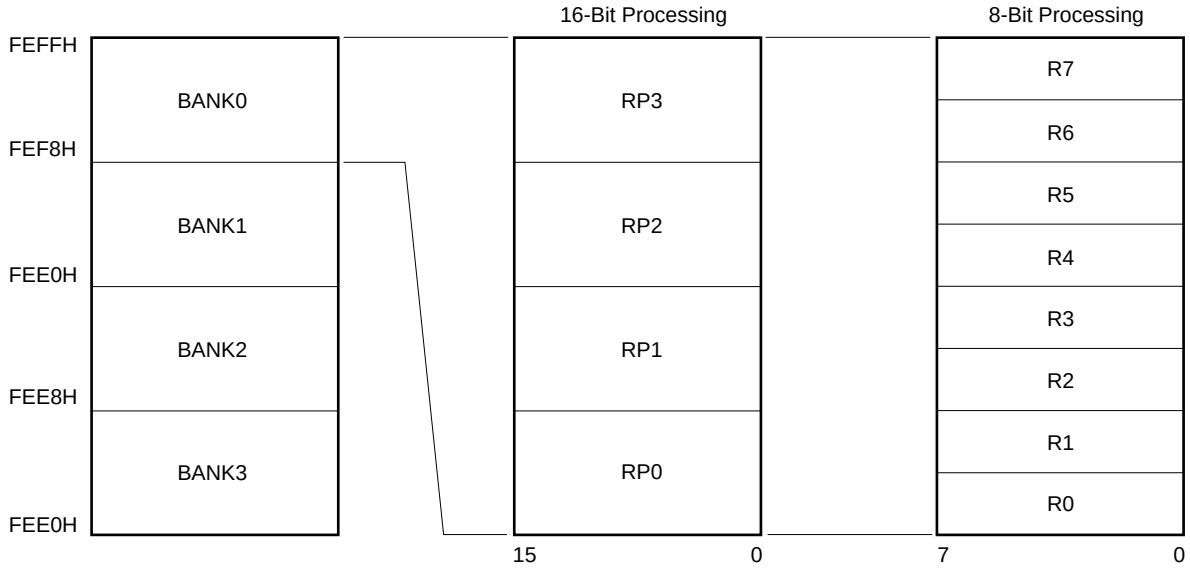
Each register can also be used as an 8-bit register. Two 8-bit registers can be used in pairs as a 16-bit register (AX, BC, DE and HL).

They can be written 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).

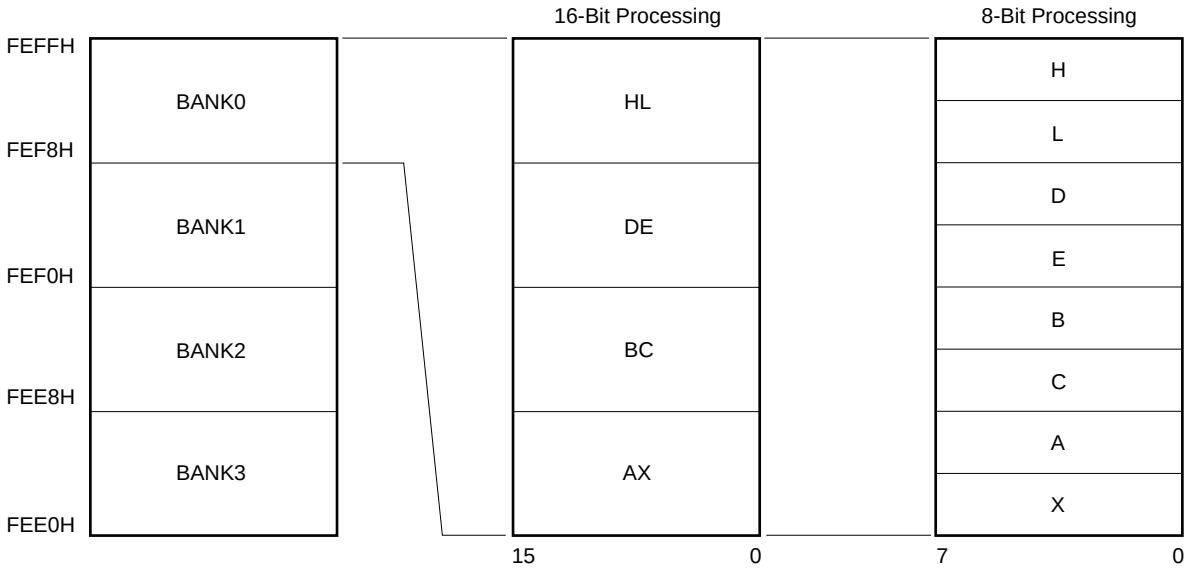
Register banks to be used for instruction execution are set with the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt request for each bank.

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**
  Use the symbol reserved in the assembler for the 1-bit manipulation instruction operand (sfr.bit).
  This manipulation can also be specified with an address.
- **8-bit manipulation**
  Use the symbol reserved in the assembler for the 8-bit manipulation instruction operand (sfr).
  This manipulation can also be specified with an address.
- **16-bit manipulation**
  Use the symbol reserved in the assembler for the 16-bit manipulation instruction operand (sfrp).
  When addressing an address, use an even address.

Table 3-2 gives a list of special-function registers. The meaning of items in the table is as follows.

- **Symbol**
  This is a symbol to indicate an address of the special-function register.
  These symbols are reserved for the DF178018 and RA78K/0, and defined by header file sfrbit.h for the CC78K/0.
  They can be written as instruction operands when the RA78K/0, ID78K0, or SD78K/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
  R&Reset : Read only (reset to 0 when read)
  W : Write only
- **Manipulatable bit units**
  ○ indicates manipulatable bit units 1, 8, and 16. – indicates the bit units that cannot be manipulated.
- **After reset**
  Indicates each register status upon reset. The values of special function registers whose addresses are not shown in the table are undefined at reset.

Table 3-2. 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		○	○	—	
FF04H	Port 4	P4		○	○	—	Undefined
FF05H	Port 5	P5		○	○	—	
FF06H	Port 6	P6		○	○	—	
FF0CH	Port 12	P12		○	○	—	00H
FF0DH	Port 13	P13		○	○	—	
FF16H	Compare register 10	CR10		—	○	—	
FF17H	Compare register 20	CR20		—	○	—	
FF18H	8-bit timer register 1	TMS TM1	R	—	○	○	00H
FF19H	8-bit timer register 2	TMS 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	○	○	—	FFH
FF21H	Port mode register 1	PM1		○	○	—	
FF22H	Port mode register 2	PM2		○	○	—	
FF23H	Port mode register 3	PM3		○	○	—	
FF25H	Port mode register 5	PM5		○	○	—	
FF26H	Port mode register 6	PM6		○	○	—	
FF2CH	Port mode register 12	PM12		○	○	—	
FF2DH	**Note**	PM13**Note**	R	○	○	—	E3H
FF41H	Timer clock select register 1	TCL1	R/W	—	○	—	00H
FF42H	Timer clock select register 2	TCL2		—	○	—	
FF43H	Timer clock select register 3	TCL3		—	○	—	88H
FF47H	Sampling clock select register	SCS		—	○	—	00H
FF49H	8-bit timer mode control register 1	TMC1		○	○	—	
FF4FH	**Note**	TOC1**Note**	R	○	○	—	
FF60H	Serial operating mode register 0	CSIM0	R/W	○	○	—	
FF61H	Serial bus interface control register	SBIC		○	○	—	
FF62H	Slave address register	SVA		—	○	—	Undefined
FF63H	Interrupt timing specification register	SINT		○	○	—	00H
FF68H	Serial operating mode register 1	CSIM1		○	○	—	01H
FF69H	Automatic data transmit/receive control register	ADTC		○	○	—	00H
FF6AH	Automatic data transmit/receive address pointer	ADTP		—	○	—	
FF6BH	Automatic data transmit/receive interval specification register	ADTI		○	○	—	

Note Do not use the symbol of this register in software as a user-defined symbol.

Table 3-2. 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	
FF80H	A/D converter mode register		ADM		R/W	○	○	—	01H
FF84H	A/D converter input select register		ADIS			—	○	—	00H
FFA0H	PLL mode select register		PLLMD			○	○	—	
FFA1H	PLL reference mode register		PLLRF			○	○	—	0FH
FFA2H	PLL unlock FF judge register		PLLUL		R&Reset	○	○	—	Held[Note 2]
FFA3H	PLL data transfer register		PLLNS		W	○	○	—	00H
FFA4H	EO select register		EOCON		R/W	○	○	—	
FFA5H	**Note 1**		PLLS[Note 1]			○	○	—	
FFA6H	PLL data register	PLL data register L	PLLR	PLLRL		○	○	○	Undefined
FFA7H		PLL data register H		PLLRH		○	○		
FFA8H	PLL data register 0		PLLR0			○	○	—	
FFA9H	IF counter mode select register		IFCMD			○	○	—	00H
FFABH	IF counter gate judge register		IFCJG		R	○	○	—	
FFACH	IF counter control register		IFCR		W	○	○	—	
FFB0H	PWM mode select register		PWMSEL		R/W	○	○	—	
FFB1H	PWM control register		PWMCR			○	○	—	
FFB4H	PWM data	PWM data register 0L	PWMR0	PWMR0L		○	○	○	01FFH
FFB5H	register 0	PWM data register 0H		PWMR0H					
FFB6H	PWM data	PWM data register 1L	PWMR1	PWMR1L		○	○	○	
FFB7H	register 1	PWM data register 1H		PWMR1H					
FFB8H	PWM data	PWM data register 2L	PWMR2	PWMTMR		○	○	○	
FFB9H	register 2	PWM data register 2H		PWMR2H					
FFBBH	**Note 1**		CHECK[Note 1]			—	—	—	00H
FFBFH	POC status register		POCS		R&Reset	○	○	○	Held[Note 2]
FFD0H to FFDFH	External access area[Note 3]				R/W	○	○	—	Undefined
FFE0H	Interrupt request flag register 0L		IF0	IF0L		○	○	○	00H
FFE1H	Interrupt request flag register 0H			IF0H		○	○		
FFE2H	**Note 4**		IF1L[Note 4]			○	○	—	
FFE4H	Interrupt mask flag register 0L		MK0	MK0L		○	○	○	FFH
FFE5H	Interrupt mask flag register 0H			MK0H		○	○		
FFE6H	**Note 4**		MK1L[Note 4]			○	○	—	
FFE8H	Priority order specification flag register 0L		PR0	PR0L		○	○	○	
FFE9H	Priority order specification flag register 0H			PR0H		○	○		
FFEAH	**Note 4**		PR1L[Note 4]			○	○	—	

Notes 1. Do not use the symbol of this register in software as a user-defined symbol.
Do not use the symbols CHECK0, CHECK2, CHECK3, CPSTP, and PPSTP as user-defined symbols.

2. The value of this register is set to 01H only when reset is effected through power-ON clearing.

3. The external access area cannot be used in the μPD178018A subseries.

4. Do not use the symbol of this register in software as a user-defined symbol.

Table 3-2. 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	
FFECH	External interrupt mode register 0	INTM0	R/W	—	○	—	00H
FFEDH	External interrupt mode register 1	INTM1		—	○	—	
FFF2H	Oscillation mode selection register	OSMS	W	—	○	—	
FFF6H	Key return mode register	KRM	R/W	○	○	—	02H
FFF8H	Port mode register 4	MM		○	○	—	10H
FFF9H	Watchdog timer mode register	WDTM		○	○	—	00H
FFFAH	Oscillation stabilization time select register	OSTS		—	○	—	04H
FFFBH	Processor clock control register	PCC		○	○	—	

3.3 Instruction Address Addressing

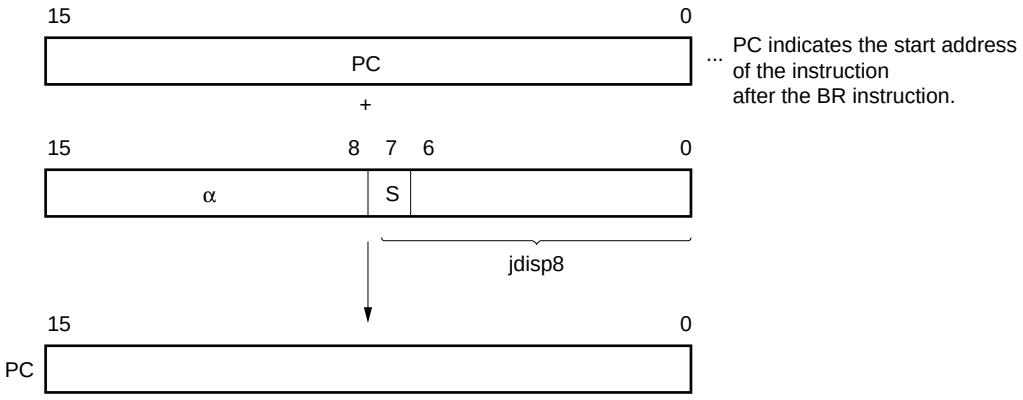
An instruction address is determined by program counter (PC) contents, and the contents is normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing. (For details of instructions, refer to **78K/0 User's Manual -Instruction (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. That is, using relative addressing, the program branches in the range –128 to +127 relative to the first address of the next instruction. This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]



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

3.3.2 Immediate addressing

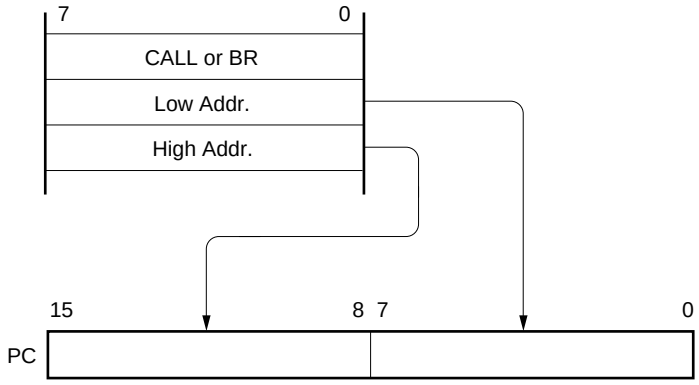
[Function]

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

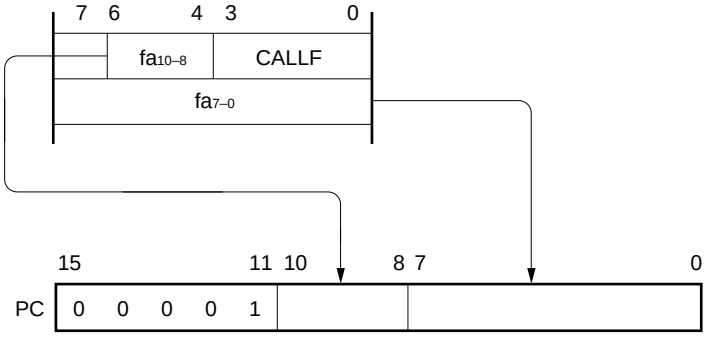
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed. The CALL !addr16 and BR !add16 instructions can be used to branch to any location in the memory. The CALLF !addr11 instruction is used to branch to the area between 0800H through 0FFFH.

[Illustration]

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



In the case of CALLF !addr11 instruction



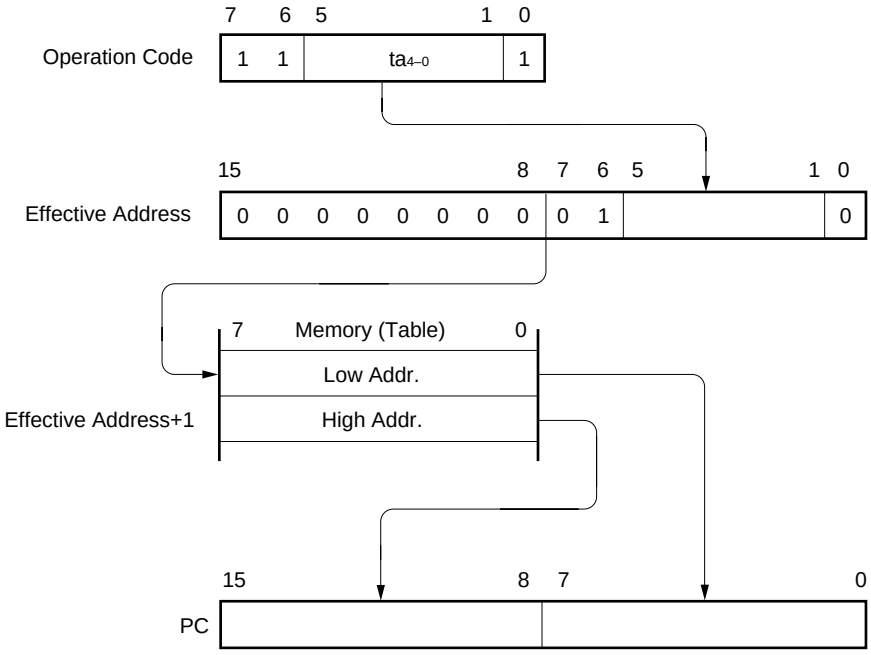
3.3.3 Table indirect addressing

[Function]

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

This addressing is used when the CALLT [addr5] instruction is executed. This instruction references an address stored in the memory table between 40H through 7FH, and can be used to branch to any location in the memory.

[Illustration]



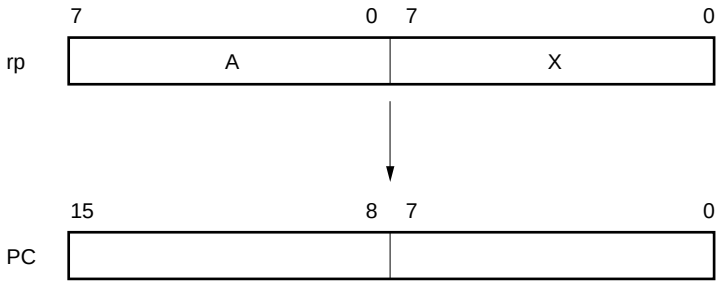
3.3.4 Register addressing

[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 various 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 addressed (implied). Of the μPD178018A 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 which become decimal correction targets
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.

[Example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

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

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

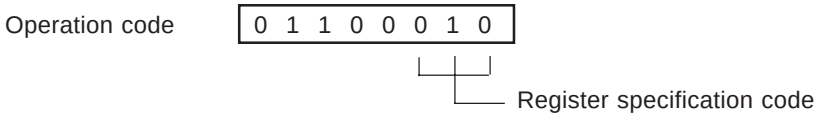
[Operand format]

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

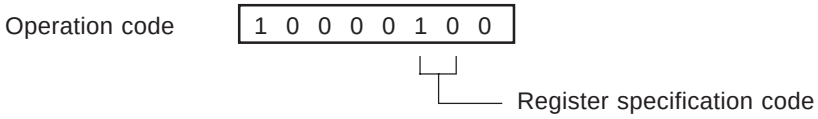
'r' and 'rp' can be written with function names (X, A, C, B, E, D, L, H, AX, BC, DE and HL) as well as absolute names (R0 to R7 and RP0 to RP3).

[Example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

The memory with immediate data in an instruction word is directly addressed.

[Operand format]

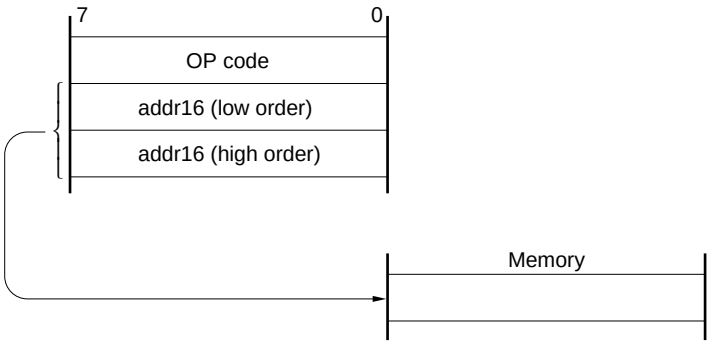
Symbol	Description
addr16	Label or 16-bit immediate data

[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 fixed 256-byte space FE20H to FF1FH. An internal 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 one part of all the SFR areas. In this area, ports which are frequently accessed in a program and a compare register of the timer/event counter is 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 [**Illustration**] on the next page.

[Operand format]

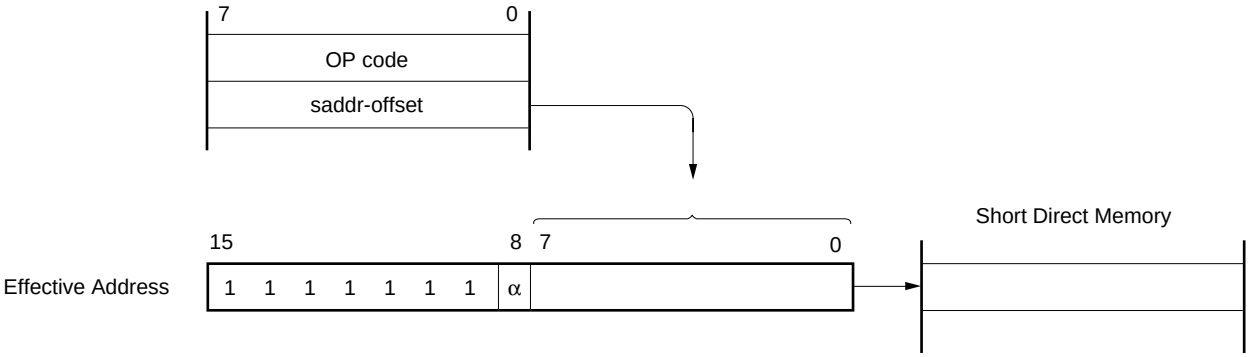
Symbol	Description
saddr	Label of FE20H to FF1FH immediate data
saddrp	Label of FE20H to FF1FH immediate data (even address only)

[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, $\alpha = 0$
When 8-bit immediate data is 00H to 1FH, $\alpha = 1$

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]

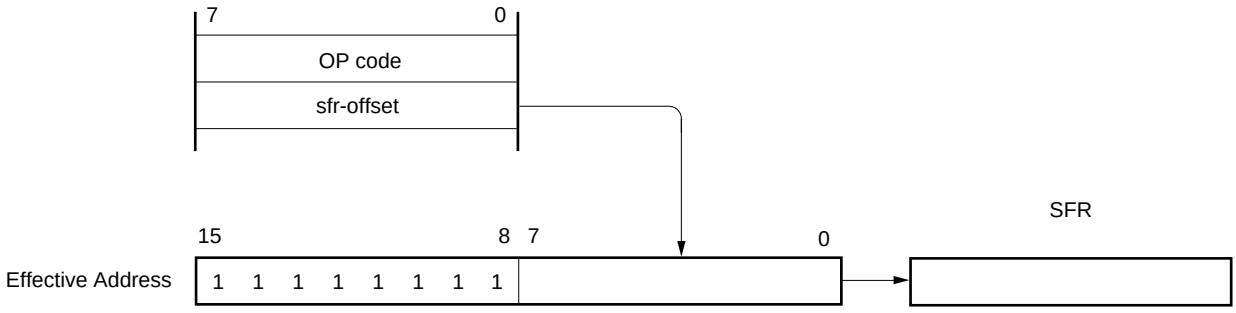
Symbol	Description
sfr	Special-function register name
sfrp	16-bit manipulatable special-function register name (even address only)

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

[Operand format]

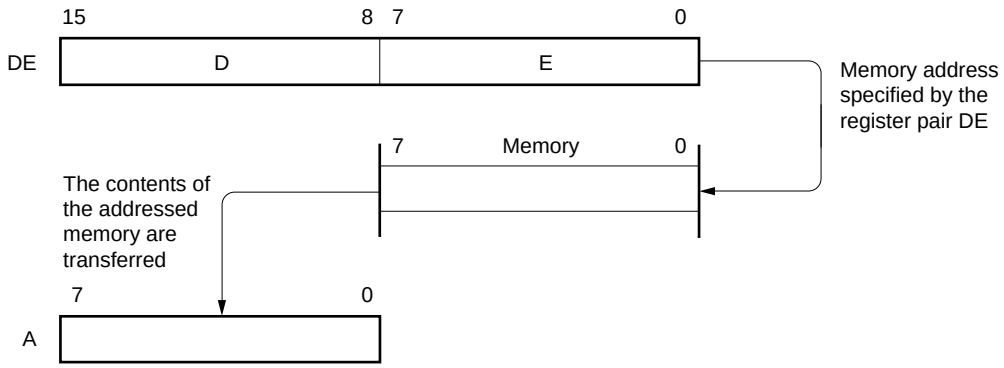
Symbol	Description
—	[DE], [HL]

[Example]

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

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

[Illustration]



3.4.7 Based addressing

[Function]

This addressing mode is used to address a memory location specified by the result of adding the 8-bit immediate data to the contents of the HL register pair which is used as a base register. The HL register pair accessed is the register 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]

Symbol	Description
—	[HL + byte]

[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 mode is used to address a memory location specified by 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 is used as a base register. The HL, B, and C registers accessed are the registers 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]

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

[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/restored upon generation of an interrupt request.

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

[Example]

In the case of PUSH DE

Operation code	1	0	1	1	0	1	0	1

[MEMO]

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD178018A subseries units incorporate one input port, three output ports and 58 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 on-chip hardware input/output pins.

Figure 4-1. Port Types

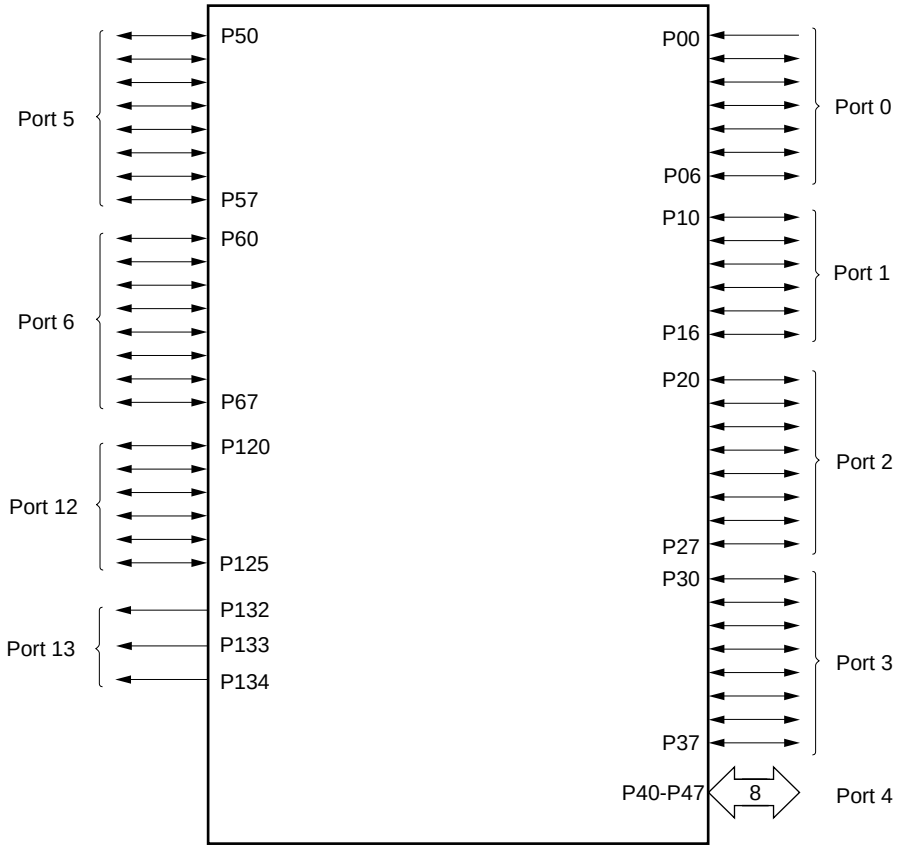


Table 4-1. Port Functions

Pin Name	I/O	Function		Dual-Function Pin
P00	Input	Port 0.	Input only	INTP0
P01-P06	I/O	8-bit input/output port.	Input/output mode can be specified bit-wise.	INTP1-INTP6
P10-P15	I/O	Port 1. 6-bit input/output port. Input/output mode can be specified bit-wise.		ANI0-ANI5
P20	I/O	Port 2.		SI1
P21		8-bit input/output port.		SO1
P22		Input/output mode can be specified bit-wise.		$\overline{\text{SCK1}}$
P23				STB
P24				BUSY
P25				SI0/SB0/SDA0
P26				SO0/SB1/SDA1
P27				$\overline{\text{SCK0}}$/SCL
P30-P32	I/O	Port 3.		—
P33		8-bit input/output port.		TI1
P34		Input/output mode can be specified bit-wise.		TI2
P35				—
P36				BEEP
P37				—
P40-P47	I/O	Port 4. 8-bit input/output port. Input/output mode can be specified bit-wise. Test input flag (KRIF) is set to 1 by falling edge detection.		—
P50-P57	I/O	Port 5. 8-bit input/output port. Input/output mode can be specified bit-wise.		—
P60-P63	I/O	Port 6. 8-bit input/output port.	Middle-voltage N-ch open drain input/ output port.	—
P64-P67		Input/output mode can be specified bit-wise.	LEDs can be driven directly.	
P120-P125	I/O	Port 12. 6-bit input/output port. Input/output mode can be specified bit-wise.		—
P132-P134	Output	Port 13. 3-bit output port. N-ch open-drain output port.		PWM0-PWM2

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 to 3, 5, 6, 12) Port mode register (MM) Note Key return mode register (KRM)
Port	Total: 62 ports (1 input, 3 outputs, 58 I/Os)

Note Port mode register 4 specifies port 4 input/output.

4.2.1 Port 0

Port 0 is a 7-bit input/output port with output latch. P01 to P06 pins can specify the input mode/output mode in 1-bit units with the port mode register 0 (PM0). P00 pin is input-only port.

Dual-functions include external interrupt request input.

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 Block Diagram

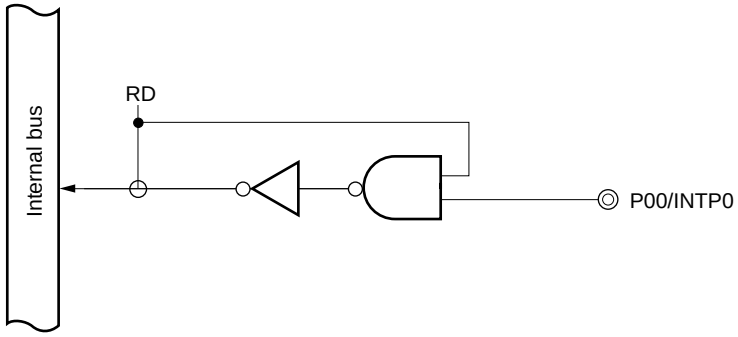
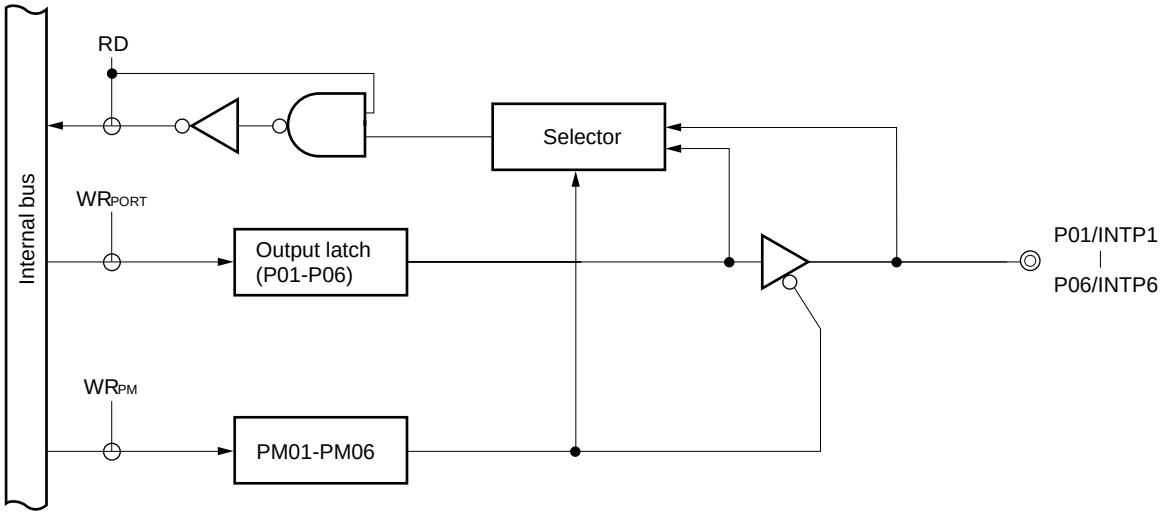


Figure 4-3. P01 to P06 Block Diagram



PM : Port mode register
RD : Port 0 read signal
WR : Port 0 write signal

4.2.2 Port 1

Port 1 is a 6-bit input/output port with output latch. It can specify the input mode/output mode in 1-bit units with a port mode register 1 (PM1).

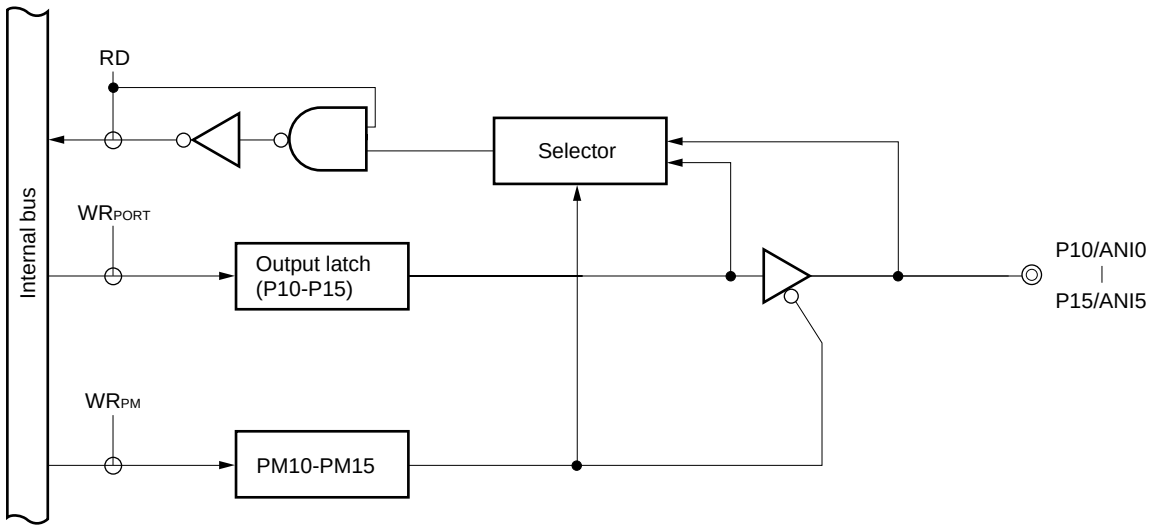
It becomes undefined when a pin specified by A/D converter input is read.

Dual-functions include an A/D converter analog input.

Reset input sets port 1 to input mode.

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

Figure 4-4. P10 to P15 Block Diagram



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).

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

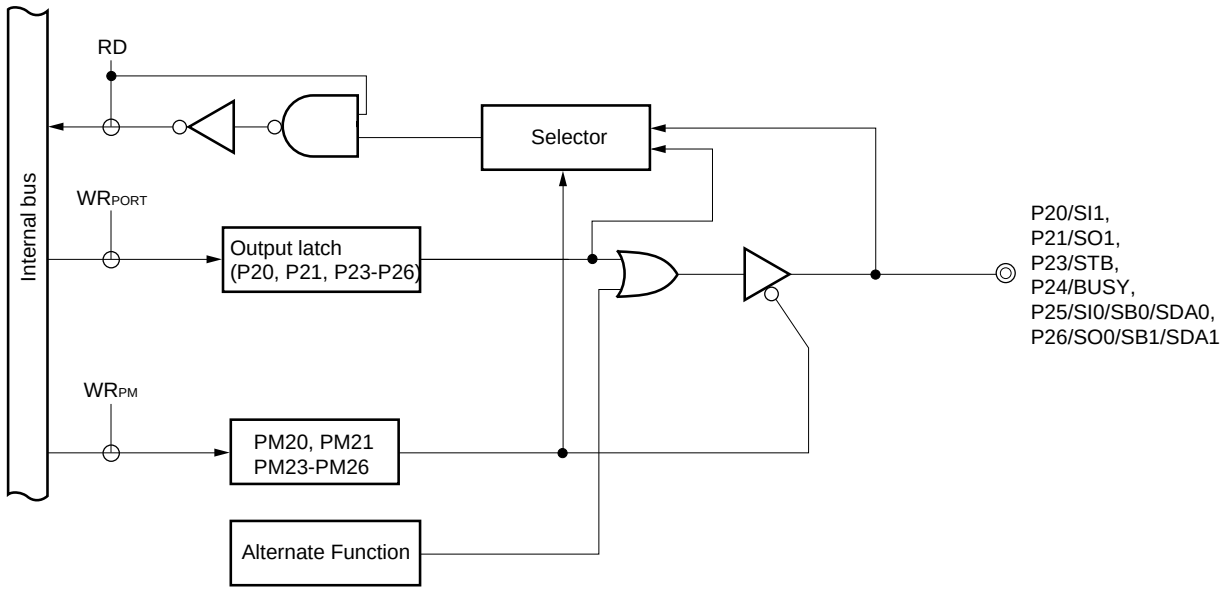
Reset input sets port 2 to input mode.

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

Cautions 1. When used as a serial interface pin, set the input/output and output latch according to its functions. For the setting method, refer to Figure 12-5 Serial Operating Mode Register 0 Format and Figure 13-3 Serial Operating Mode Register 1 Format.

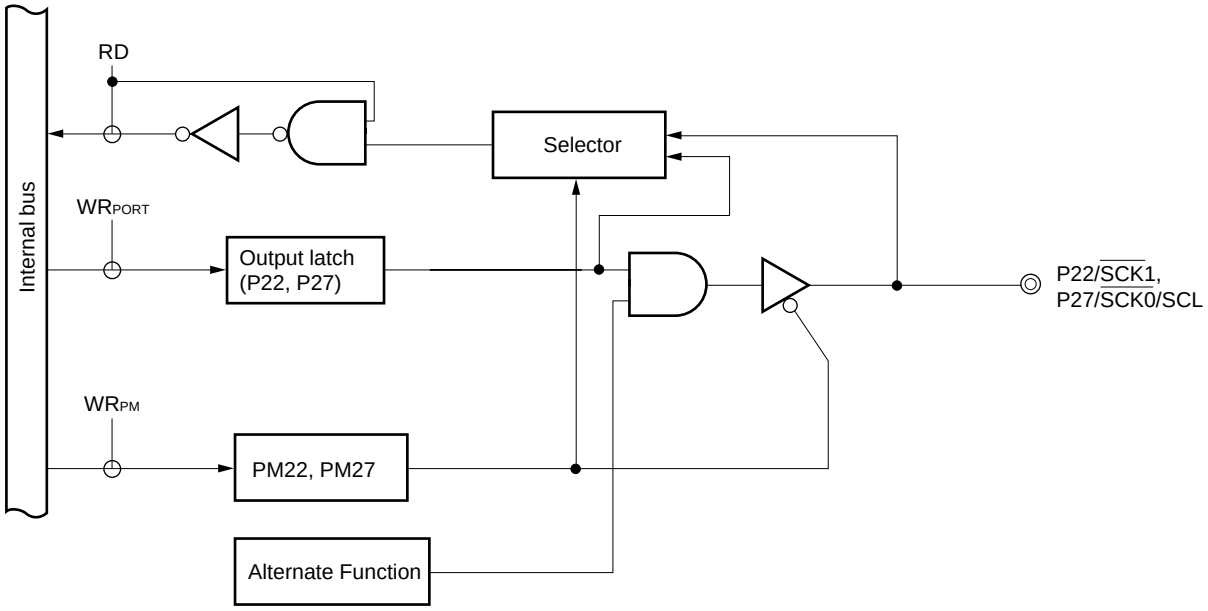
2. When reading the pin state in SBI mode, set PM2n bit of PM2 to 1 (n = 5, 6) (Refer to the description of (10) Method to judge busy state of a slave (e) in 12.4.3 SBI mode operation).

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



PM : Port mode register
RD : Port 2 read signal
WR : Port 2 write signal

Figure 4-6. P22 and P27 Block Diagram



PM : Port mode register
RD : Port 2 read signal
WR : Port 2 write signal

4.2.4 Port 3

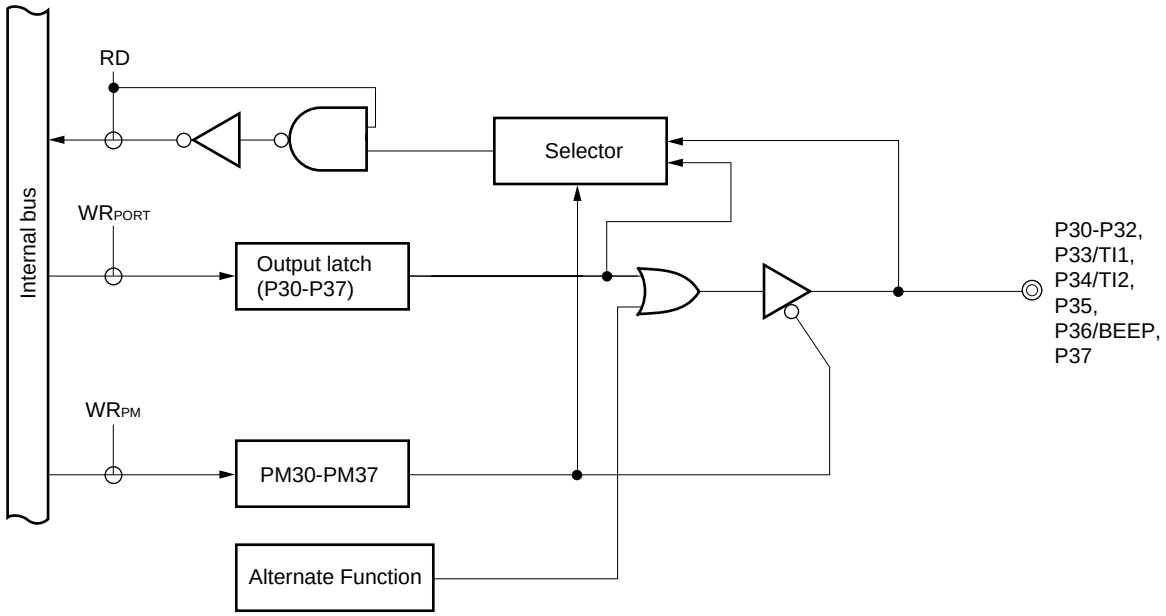
Port 3 is an 8-bit input/output port with output latch. P30 to P37 pins can specify the input mode/output mode in 1-bit units with the port mode register 3 (PM3).

Dual-functions include timer input and buzzer output.

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



PM : Port mode register
RD : Port 3 read signal
WR : Port 3 write signal

4.2.5 Port 4

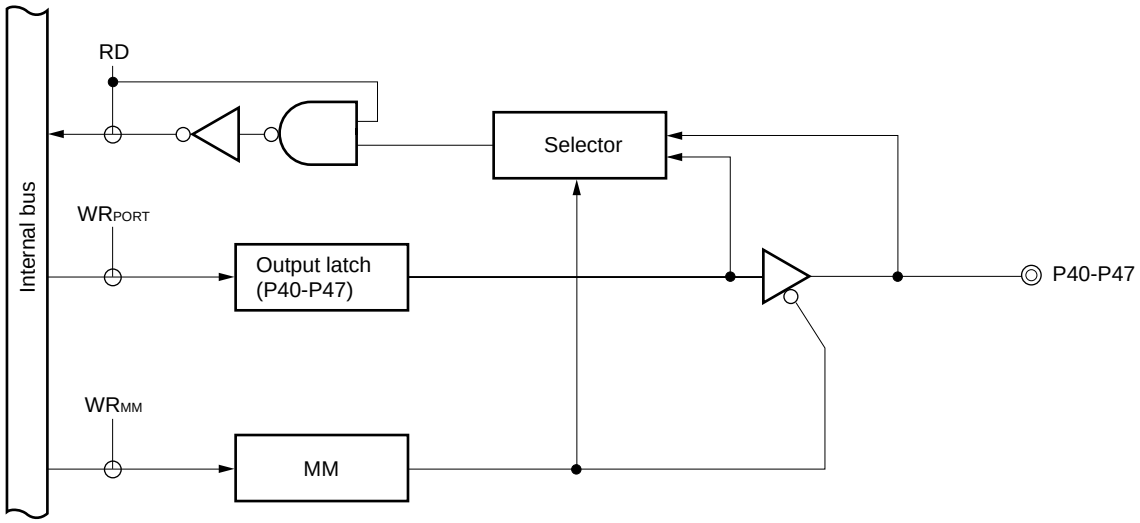
Port 4 is an 8-bit input/output port with output latch. P40 to P47 pins can specify the input mode/output mode in 8-bit units with the port mode register 4 (MM).

The test input flag (KRIF) can be set to 1 by detecting falling edges.

Reset input sets port 4 to input mode.

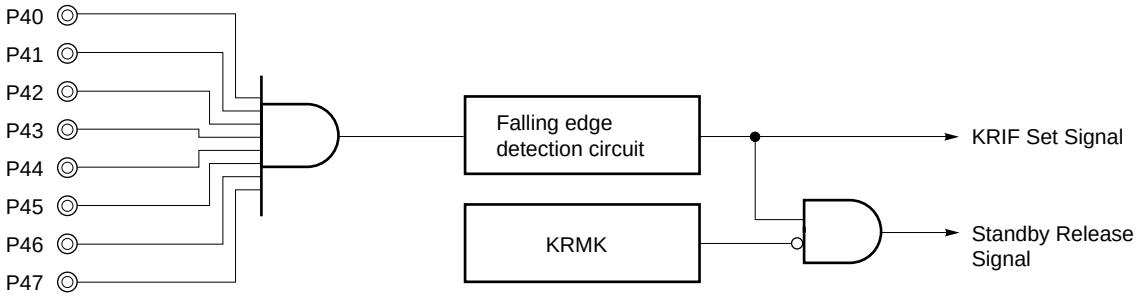
Figures 4-8 and 4-9 show a block diagram of port 4 and block diagram of falling edge detection circuit, respectively.

Figure 4-8. P40 to P47 Block Diagram



MM : Port mode register 4
RD : Port 4 read signal
WR : Port 4 write signal

Figure 4-9. Block Diagram of Falling Edge Detection Circuit



KRIF : Test input flag
KRMK : Test mask flag

4.2.6 Port 5

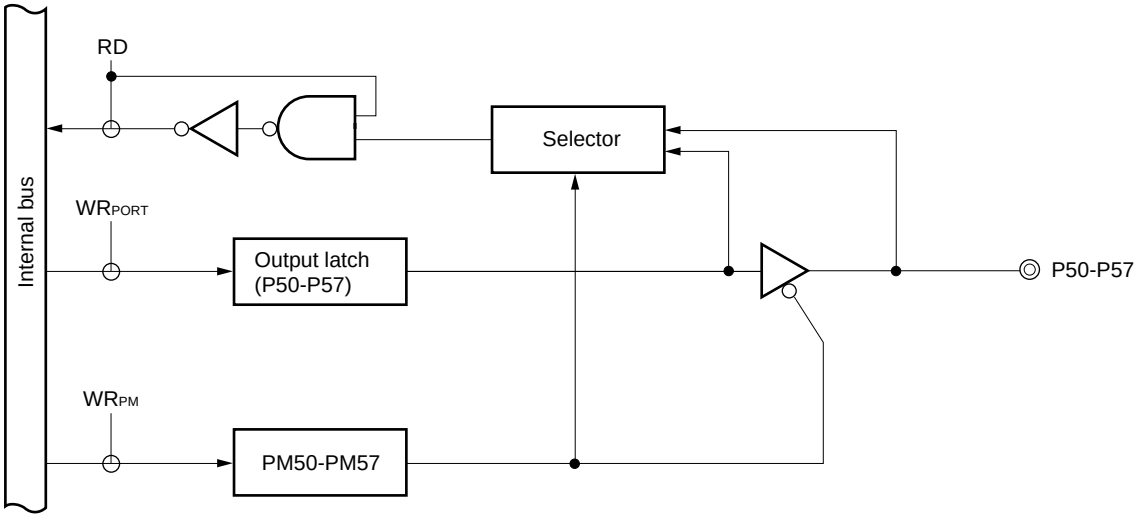
Port 5 is an 8-bit input/output port with output latch. P50 to P57 pins can specify the input mode/output mode in 1-bit units with the port mode register 5 (PM5).

Port 5 can drive LEDs directly.

Reset input sets port 5 to input mode.

Figure 4-10 shows a block diagram of port 5.

Figure 4-10. P50 to P57 Block Diagram



PM : Port mode register
RD : Port 5 read signal
WR : Port 5 write signal

4.2.7 Port 6

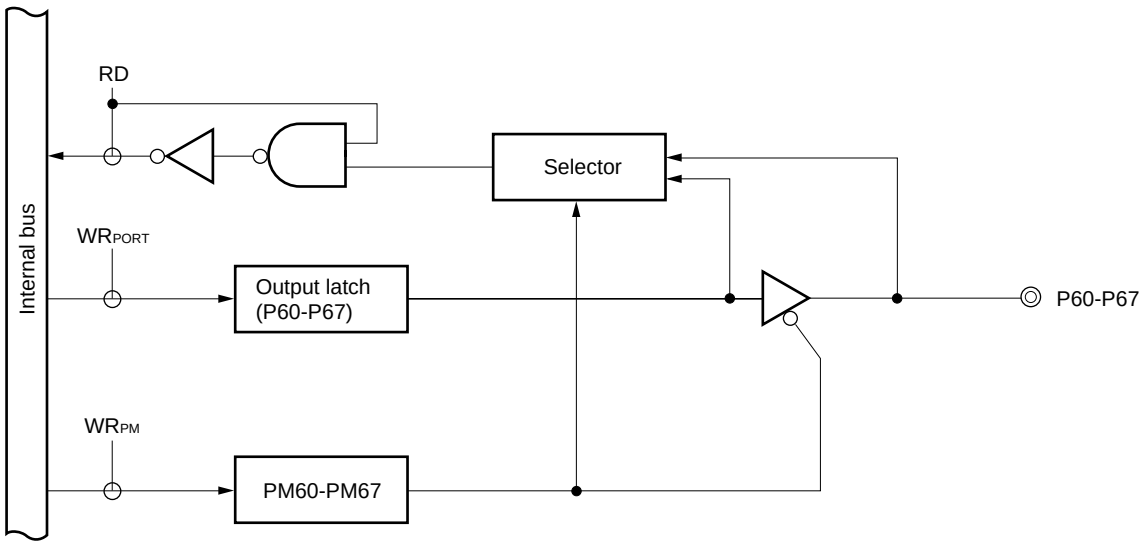
Port 6 is an 8-bit input/output port with output latch. P60 to P67 pins can specify the input mode/output mode in 1-bit units with the port mode register 6 (PM6).

Pins P60 to P63 can drive LEDs directly.

Reset input sets port 6 to input mode.

Figure 4-11 shows block diagram of port 6.

Figure 4-11. P60 to P67 Block Diagram



PM : Port mode register

RD : Port 6 read signal

WR : Port 6 write signal

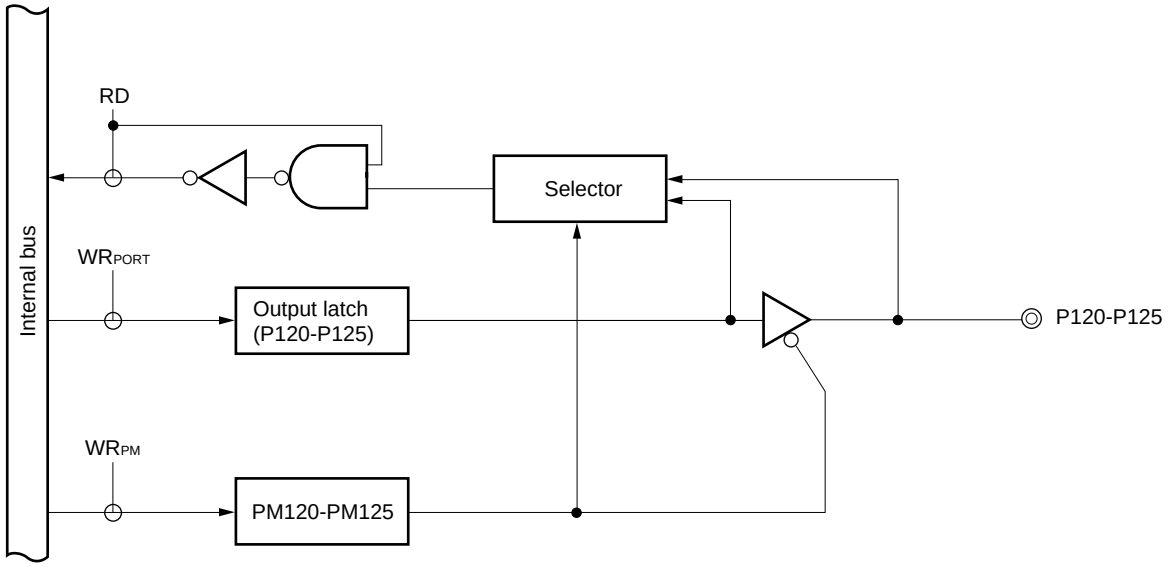
4.2.8 Port 12

This is a 6-bit input/output port with output latches. Input mode/output mode can be specified bit-wise by means of port mode register 12 (PM12).

Reset input sets the input mode.

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

Figure 4-12. P120 to P125 Block Diagram



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

4.2.9 Port 13

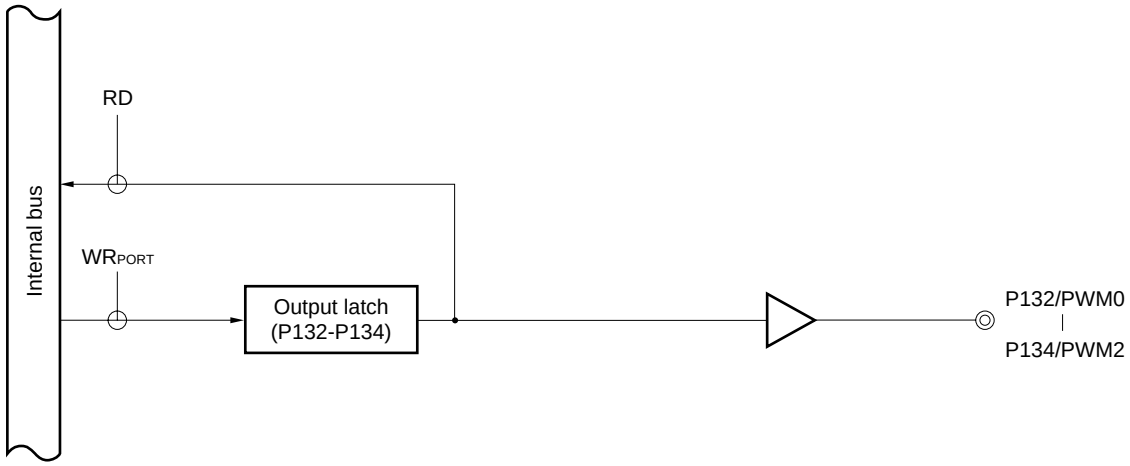
This is a 3-bit output port with an output latch.

The pins of this port are also used as PWM output pins. A pin specified as a PWM output pin is undefined when it is read.

These pins are set in the general-purpose output port mode at reset.

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

Figure 4-13. P132 to P134 Block Diagram



RD : Port 13 read signal

WR : Port 13 write signal

4.3 Port Function Control Registers

The following three types of registers control the ports.

- Port mode registers (PM0 to PM3, PM5, PM6, PM12)
- Port mode register 4 (MM)
- Key return mode register (KRM)

(1) Port mode registers (PM0 to PM3, PM5, PM6, PM12)

These registers are used to set port input/output in 1-bit units.

PM0 to PM3, PM5, PM6, and PM12 are independently set with a 1-bit or 8-bit memory manipulation instruction

Reset input sets registers to FFH.

When port pins are used as the dual-function pins, set the port mode register and output latch according to Table 4-3.

Cautions 1. Pin P00 is input-only pin.

2. As port 0 has a dual 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.

3. The port mode register 4 (MM) specifies P40 to P47 as input/output pins.

Table 4-3. Port Mode Register and Output Latch Settings when Using Dual-Functions

Pin Name	Alternate Functions		PM××	P××
	Name	Input/Output		
P00	INTP0	Input	None	None
P01-P06	INTP1-INTP6	Input	1	×
P10-P15[Note]	ANI0-ANI5	Input	1	×
P33	TI1	Input	1	×
P34	TI2	Input	1	×
P36	BEEP	Output	0	0
P132, P134	PWM0-PWM2	Output	None	×

Note If these ports are read out when these pins are used in the alternate function mode, undefined values are read.

Caution When port 2 is used for serial interface, the I/O latch or output latch must be set according to its function. For the setting methods, refer to Figure 12-5 Serial Operating Mode Register 0 Format and Figure 13-3 Serial Operating Mode Register 1 Format.

Remark × : don't care

PM×× : port mode register

P×× : 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	1	PM06	PM05	PM04	PM03	PM02	PM01	1	FF20H	FFH	R/W
PM1	1	1	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM12	1	1	PM125	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn Pin Input/Output Mode Selection (m=0-3, 6, 12 : n=0-7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(2) Port mode register 4 (MM)

This register is used to set input/output of port 4.

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

Reset input sets this register to 10H.

Figure 4-15. Port Mode Register 4 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
MM	0	0	0[Note]	1[Note]	0	MM2	MM1	MM0	FFF8H	10H	R/W

MM2	MM1	MM0	I/O Mode Selection of P40-P47
0	0	0	Input mode
0	0	1	Output mode
Other than above			Setting prohibited

Note Be sure to set MM bit 4 to 1, and bit 5 to 0.

Caution Be sure to set MM2 and MM1 to 0.

(3) Key return mode register (KRM)

This register sets enabling/disabling of standby function release by a key return signal (falling edge detection of port 4).

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

Reset input sets KRM to 02H.

Figure 4-16. Key Return Mode Register Format

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

KRIF	Key Return Signal Detection Flag
0	Not Detected
1	Detected (Falling edge detection of port 4)

KRMK	Standby Mode Control by Key Return Signal
0	Standby mode release enabled
1	Standby mode release disabled

Caution When falling edge detection of port4 is used, KRIF should be cleared to 0 (not cleared to 0 automatically).

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 as an 8-bit unit. 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) 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) Input mode

The output latch contents are undefined, but since the output buffer is OFF, the pin status does not change.

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

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

The clock generator generates the clock to be supplied to the CPU and peripheral hardware. This system clock oscillator oscillates at frequencies of 4.5 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).

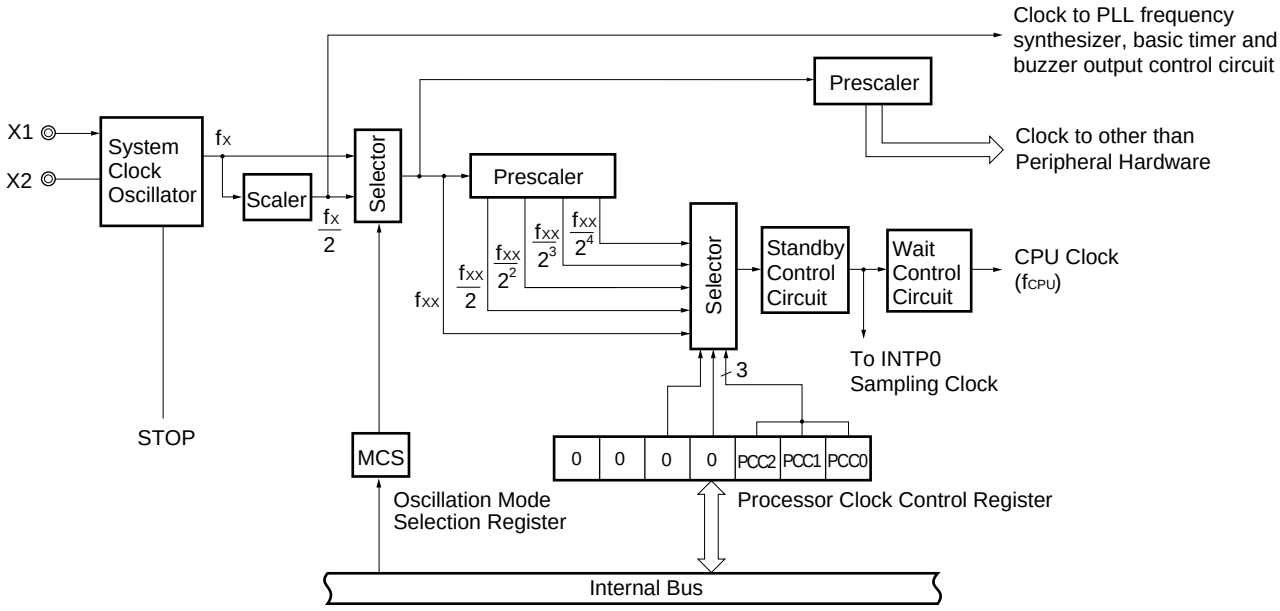
5.2 Clock Generator Configuration

The clock generator consists of the following hardware.

Table 5-1. Clock Generator Configuration

Item	Configuration
Control register	Processor clock control register (PCC) Oscillation mode selection register (OSMS)
Oscillator	System clock oscillator

Figure 5-1. Block Diagram of Clock Generator



5.3 Clock Generator Control Register

The clock generator is controlled by the following two registers:

- Processor clock control register (PCC)
- Oscillation mode selection register (OSMS)

(1) Processor clock control register (PCC)

The PCC sets CPU clock.

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

Reset input sets the PCC to 04H.

Figure 5-2. Processor Clock Control Register Format

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

R/W		
	CMS	Be sure to set 0.

R/W	PCC2	PCC1	PCC0	CPU CIock (f_{CPU}) Selection		
					MCS = 1	MCS = 0
	0	0	0	f_{XX}	f_X	$f_X/2$
	0	0	1	$f_{XX}/2$	$f_X/2$	$f_X/2^2$
	0	1	0	$f_{XX}/2^2$	$f_X/2^2$	$f_X/2^3$
	0	1	1	$f_{XX}/2^3$	$f_X/2^3$	$f_X/2^4$
	1	0	0	$f_{XX}/2^4$	$f_X/2^4$	$f_X/2^5$
	Other than above			Setting prohibited		

Note Bits 4 to 7 are Read Only.

Caution Be sure to set CMS (bit 3) to 0.

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillator frequency

3. MCS : Bit 0 of oscillation mode selection register (OSMS)

The fastest instruction of the μPD178018A subseries is executed with in 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	0.44 μs
$f_X/2$	0.89 μs
$f_X/2^2$	1.78 μs
$f_X/2^3$	3.56 μs
$f_X/2^4$	7.11 μs
$f_X/2^5$	14.22 μs

f_X = 4.5 MHz

f_X : System clock oscillation frequency

(2) Oscillation mode selection register (OSMS)

This register specifies whether the clock output from the system clock oscillator without passing through the scaler is used as the system clock, or the clock output via the scaler is used as the main system clock.

OSMS is set with 8-bit memory manipulation instruction.

Reset input sets MCS flag to 0, but when MCS flag is read, it becomes undefined.

Figure 5-3. Oscillation Mode Selection Register Format

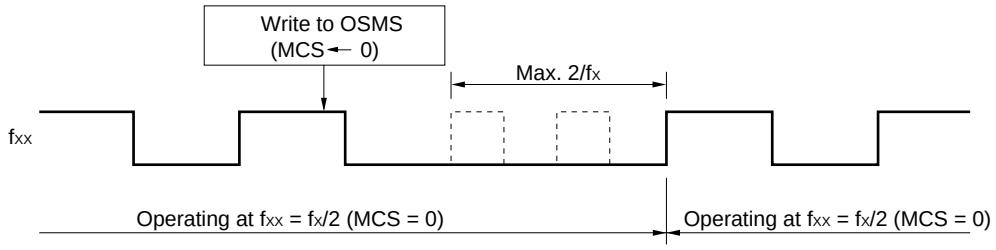
Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
OSMS	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	MCS	FFF2H	00H	W

MCS	System Clock Scaler Control
0	Scaler used
1	Scaler not used

Caution 1. As shown in Figure 5-4 below, writing data (including same data as previous) to OSMS cause delay of system clock cycle up to 2/f_X during the write operation. Therefore, if this register is written during the operation, in peripheral hardware which operates with the system clock, a temporary error occurs in the count clock cycle of timer, etc. In addition, because the oscillation mode is switched by this register, the clocks for peripheral hardware as well as that for the CPU are switched. However, the PLL synthesizer, basic timer and the clock supplied for buzzer output control circuit are not affected.

2. When writing "1" to MCS, V_{DD} must be 4.5 V or higher before the write execution.

Figure 5-4. System Clock Waveform due to Writing to OSMS



Remark f_{XX} : System clock frequency (fx or fx/2)

f_X : System clock oscillation frequency

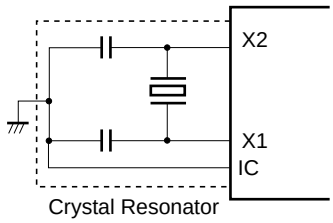
5.4 System Clock Oscillator

5.4.1 System clock oscillator

The system clock oscillator oscillates with a crystal resonator (4.5 MHz TYP.) connected to the X1 and X2 pins. Figure 5-5 shows an external circuit of the system clock oscillator.

Figure 5-5. External Circuit of System Clock Oscillator

Crystal oscillation



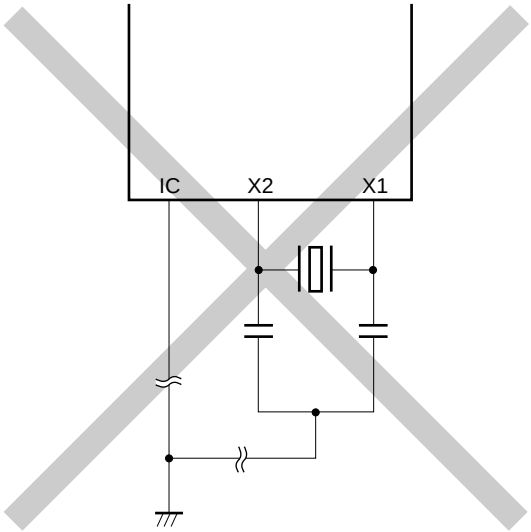
Caution When using a system clock oscillator, carry out wiring in the broken line area in Figure 5-5 to prevent any effects from wiring capacities.

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

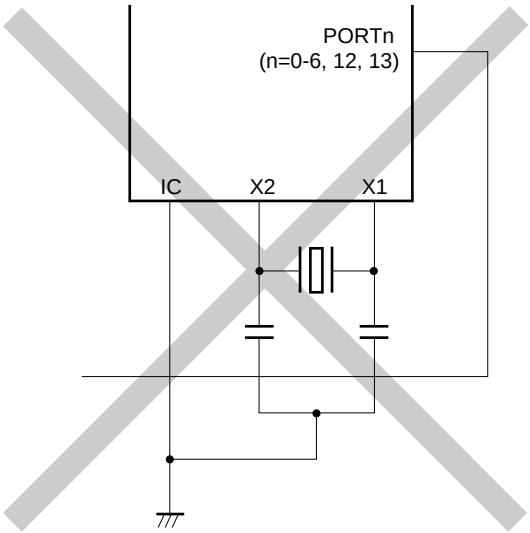
Figure 5-6 shows examples of resonator having bad connection.

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

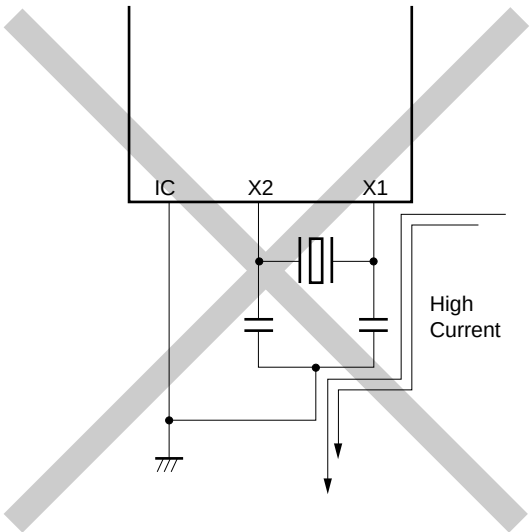
(a) Wiring of connection circuits is too long



(b) Signal conductors intersect each other



(c) Changing high current is too near a signal conductor



(d) Current flows through the grounding line of the oscillator (potential at points A, B, and C fluctuate)

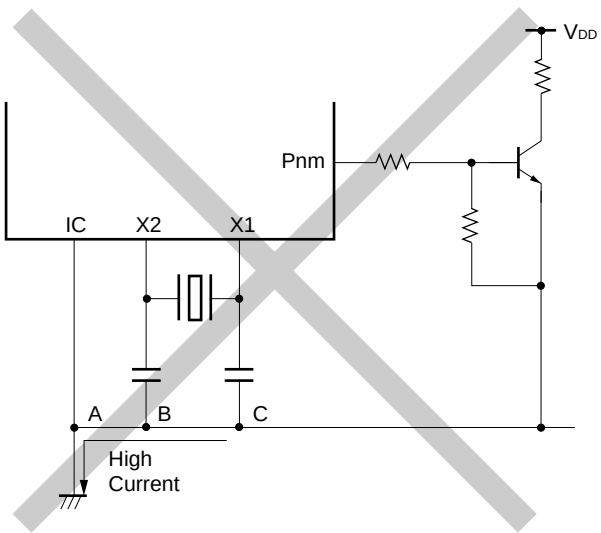
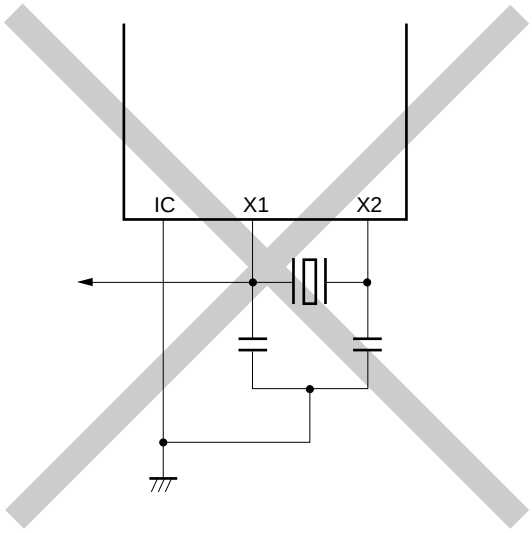


Figure 5-6. Examples of Resonator with Bad Connection (2/2)

(e) Signals are fetched



5.4.2 Scaler

The scaler divides the system clock oscillator output (f_{XX}) and generates various clocks.

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.

- System clock f_{XX}
- CPU clock f_{CPU}
- Clock to peripheral hardware

The following clock generator functions and operations are determined with the processor clock control register (PCC) and the oscillation mode selection register (OSMS).

(a) Upon generation of reset signal, the lowest speed mode of the system clock (14.22 μs when operated at 4.5 MHz) is selected. System clock oscillation stops while low level is applied to $\overline{\text{RESET}}$ pin.

(b) One of the six CPU clock types (0.44, 0.89, 1.78, 3.56, 7.11, 14.22 μs at 4.5 MHz) can be selected by setting the PCC and OSMS.

(c) Two standby modes, the STOP and HALT modes, are available.

(d) The system clock is divided and supplied to the peripheral hardware. The peripheral hardware also stops if the system clock is stopped.

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) 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 (refer to **Table 5-3**).

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

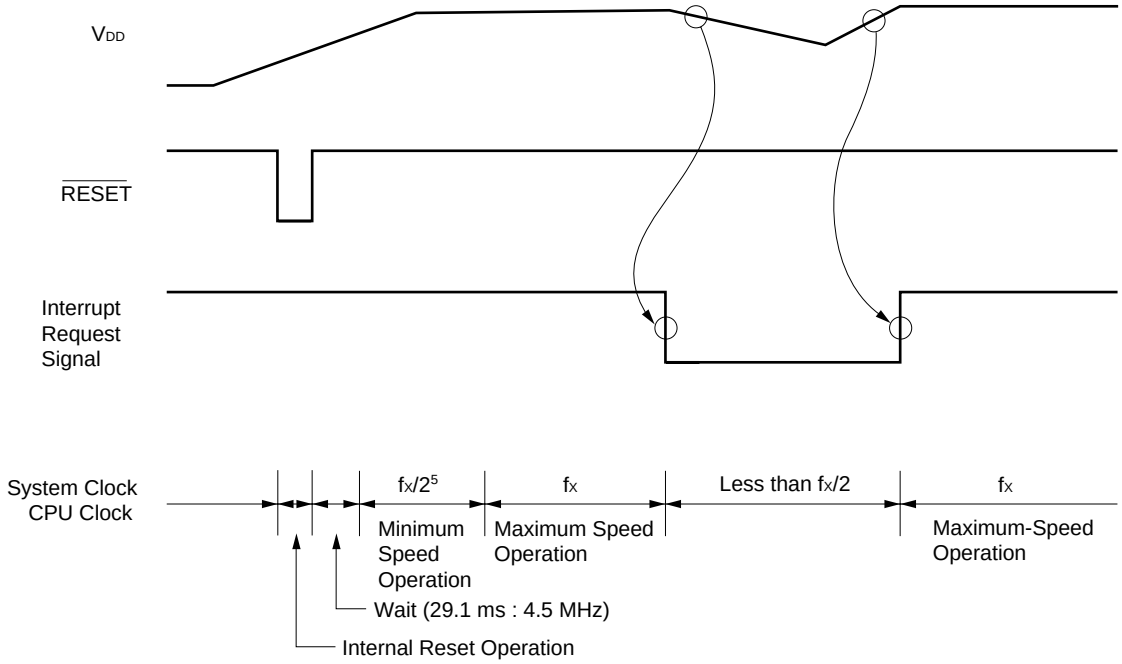
Set Values before Switchover			Set Values After Switchover														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 instructions			16 instructions			16 instructions			16 instructions		
0	0	1	8 instructions						8 instructions			8 instructions			8 instructions		
0	1	0	4 instructions			4 instructions						4 instructions			4 instructions		
0	1	1	2 instructions			2 instructions			2 instructions						2 instructions		
1	0	0	1 instruction			1 instruction			1 instruction			1 instruction					

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

5.6.2 System clock and CPU clock switching procedure

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

Figure 5-7. System Clock and CPU Clock Switching



(1) The CPU is reset by setting the $\overline{\text{RESET}}$ signal to low level, or V_{DD} becomes less than power-on cleare voltage after power-on. After that, when reset is released by setting the $\overline{\text{RESET}}$ signal to high level, or V_{DD} become higher than power-on clear voltage, 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 system clock (14.22 μs when operated at 4.5 MHz).

(2) After the lapse of a sufficient time for the V_{DD} voltage to increase to enable operation at maximum speeds, the PCC and OSMS are 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, the PCC and OSMS are rewritten.

(4) Upon detection of V_{DD} voltage restored due to an interrupt request, the PCC and OSMS are rewritten and the maximum-speed operation is resumed.

CHAPTER 6 8-BIT TIMER/EVENT COUNTERS 1 AND 2

6.1 8-Bit Timer/Event Counters 1 and 2 Functions

For the 8-bit timer/event counters 1 and 2, two modes are available. One is a mode for two-channel 8-bit timer/event counters to be used separately (the 8-bit timer/event counter mode) and the other is a mode for the 8-bit timer/event counter to be used as 16-bit timer/event counter (the 16-bit timer/event counter mode).

6.1.1 8-bit timer/event counter mode

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

- Interval timer
- External event counter

(1) 8-bit interval timer

Interrupt requests are generated at the preset time intervals.

Table 6-1. 8-Bit Timer/Event Counters 1 and 2 Interval Times

Minimum Interval Time		Maximum Interval Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)
$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)
$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)
$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{13} \times 1/f_X$ (1.82 ms)	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)
$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)	$2^{13} \times 1/f_X$ (1.82 ms)	$2^{14} \times 1/f_X$ (3.64 ms)	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)
$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)	$2^{14} \times 1/f_X$ (3.64 ms)	$2^{15} \times 1/f_X$ (7.28 ms)	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)
$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)	$2^{15} \times 1/f_X$ (7.28 ms)	$2^{16} \times 1/f_X$ (14.6 ms)	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)
$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{16} \times 1/f_X$ (14.6 ms)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)
$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)
$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)

Remarks 1. f_X : System clock oscillation frequency

2. MCS : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at f_X = 4.5 MHz.

(2) External event counter

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

6.1.2 16-bit timer/event counter mode

(1) 16-bit interval timer

Interrupt requests can be generated at the preset time intervals.

Table 6-2. Interval Times when 8-Bit Timer/Event Counters 1 and 2 are Used as 16-Bit Timer/Event Counters

Minimum Interval Time		Maximum Interval Time		Resolution	
MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)
$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)	$2^{18} \times 1/f_X$ (58.3 ms)	$2^{19} \times 1/f_X$ (116.5 ms)	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)
$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)
$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)	$2^{20} \times 1/f_X$ (233.0 ms)	$2^{21} \times 1/f_X$ (466.0 ms)	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)
$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)	$2^{21} \times 1/f_X$ (466.0 ms)	$2^{22} \times 1/f_X$ (932.0 ms)	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)
$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)	$2^{22} \times 1/f_X$ (932.0 ms)	$2^{23} \times 1/f_X$ (1.9 s)	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)
$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)	$2^{23} \times 1/f_X$ (1.9 s)	$2^{24} \times 1/f_X$ (3.7 s)	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)
$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{24} \times 1/f_X$ (3.7 s)	$2^{25} \times 1/f_X$ (7.5 s)	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)
$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{25} \times 1/f_X$ (7.5 s)	$2^{26} \times 1/f_X$ (14.9 s)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)
$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{27} \times 1/f_X$ (29.8 s)	$2^{28} \times 1/f_X$ (59.7 s)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)

Remarks 1. f_X : System clock oscillation frequency

2. MCS : Oscillation mode selection register (OSMS) bit 0

3. Values in parentheses when operated at f_X = 4.5 MHz.

(2) External event counter

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

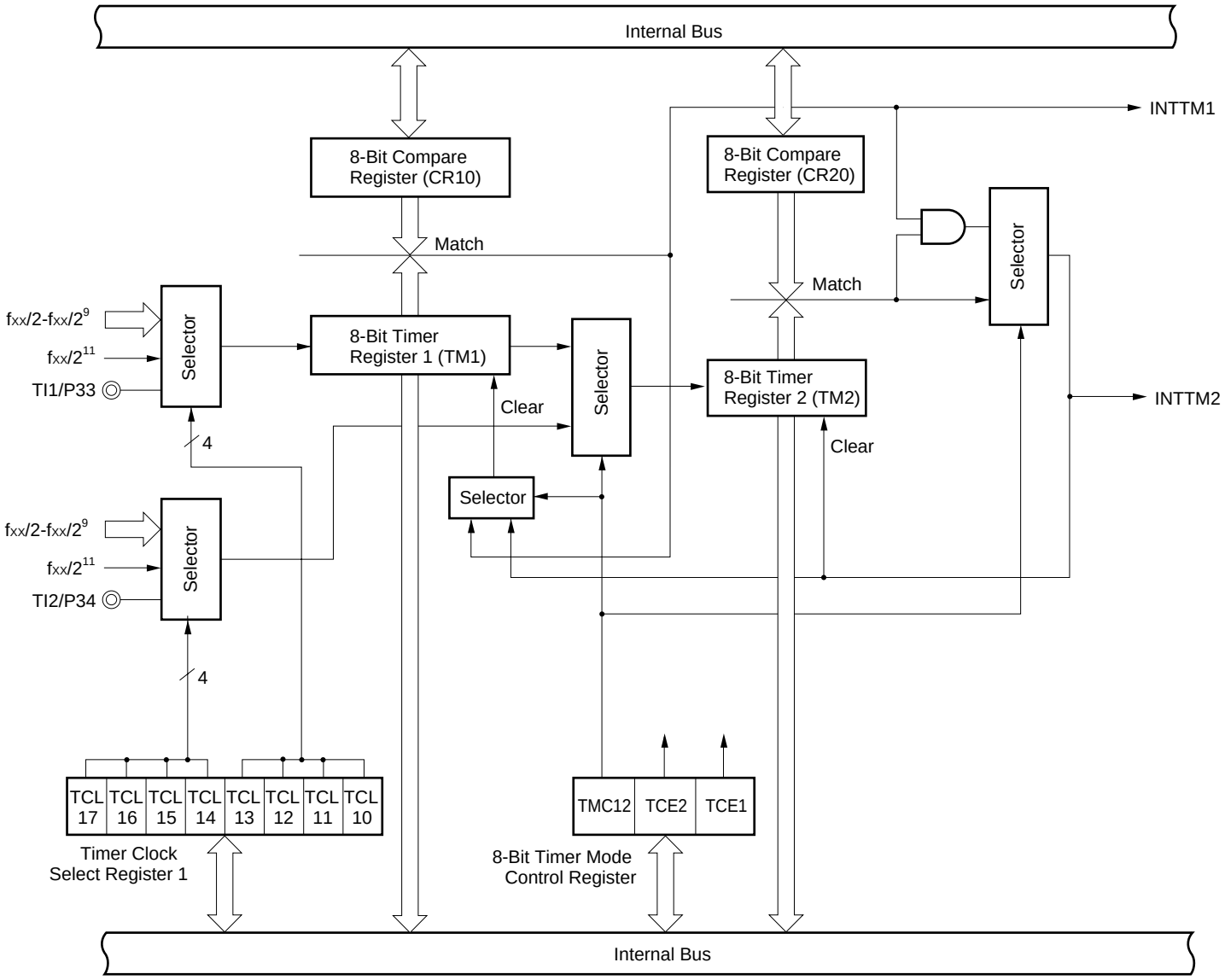
6.2 8-Bit Timer/Event Counters 1 and 2 Configurations

The 8-bit timer/event counters 1 and 2 consist of the following hardware.

Table 6-3. 8-Bit Timer/Event Counters 1 and 2 Configurations

Item	Configuration
Timer register	8 bits × 2 (TM1, TM2)
Register	Compare register: 8 bits × 2 (CR10, CR20)
Control register	Timer clock select register 1 (TCL1) 8-bit timer mode control register 1 (TMC1)

Figure 6-1. 8-Bit Timer/Event Counters 1 and 2 Block Diagram



(1) Compare registers 10 and 20 (CR10, CR20)

These are 8-bit registers 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, generate 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.

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 new values of CR10 and CR20 are less than the value of the 8-bit timer registers (TM1, TM2), TM1 and TM2 continue counting, overflows and count again from 0. If the new values of CR10 and CR20 are less than the previous values, it is necessary to restart the timer.

(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 8-bit timer × 2-channel mode, they are read with an 8-bit memory manipulation instruction. When TM1 and TM2 are used as 16-bit timer × 1-channel mode, 16-bit timer (TMS) is read with a 16-bit memory manipulation instruction.

Reset input sets TM1 and TM2 to 00H.

6.3 8-Bit Timer/Event Counters 1 and 2 Control Registers

The following two 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 1 (TMC1)

(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.

Reset input sets TCL1 to 00H.

Figure 6-2. 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		
					MCS = 1	MCS = 0
0	0	0	0	TI1 falling edge		
0	0	0	1	TI1 rising edge		
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.25 MHz)	$f_X/2^2$ (1.13 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.13 MHz)	$f_X/2^3$ (563 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (563 kHz)	$f_X/2^4$ (281 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (281 kHz)	$f_X/2^5$ (141 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (141 kHz)	$f_X/2^6$ (70.3 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (70.3 kHz)	$f_X/2^7$ (35.2 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (35.2 kHz)	$f_X/2^8$ (17.6 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (17.6 kHz)	$f_X/2^9$ (8.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (8.8 kHz)	$f_X/2^{10}$ (4.4 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.2 kHz)	$f_X/2^{12}$ (1.1 kHz)
Other than above				Setting prohibited		

TCL17	TCL16	TCL15	TCL14	8-Bit Timer Register 2 Count Clock Selection		
					MCS = 1	MCS = 0
0	0	0	0	TI2 falling edge		
0	0	0	1	TI2 rising edge		
0	1	1	0	$f_{XX}/2$	$f_X/2$ (2.25 MHz)	$f_X/2^2$ (1.13 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.13 MHz)	$f_X/2^3$ (563 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (563 kHz)	$f_X/2^4$ (281 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (281 kHz)	$f_X/2^5$ (141 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (141 kHz)	$f_X/2^6$ (70.3 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (70.3 kHz)	$f_X/2^7$ (35.2 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (35.2 kHz)	$f_X/2^8$ (17.6 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (17.6 kHz)	$f_X/2^9$ (8.8 kHz)
1	1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (8.8 kHz)	$f_X/2^{10}$ (4.4 kHz)
1	1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.2 kHz)	$f_X/2^{12}$ (1.1 kHz)
Other than above				Setting prohibited		

Caution When rewriting TCL1 to other data, stop the timer operation beforehand.

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)
2. f_X : System clock oscillation frequency
3. TI1 : 8-bit timer register 1 input pin
4. TI2 : 8-bit timer register 2 input pin
5. MCS : Oscillation mode selection register (OSMS) bit 0
6. Figures in parentheses apply to operation with f_X = 4.5 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.

Reset input sets TMC1 to 00H.

Figure 6-3. 8-Bit Timer Mode Control Register 1 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.

6.4 8-Bit Timer/Event Counters 1 and 2 Operations

6.4.1 8-bit timer/event counter mode

(1) Interval timer operations

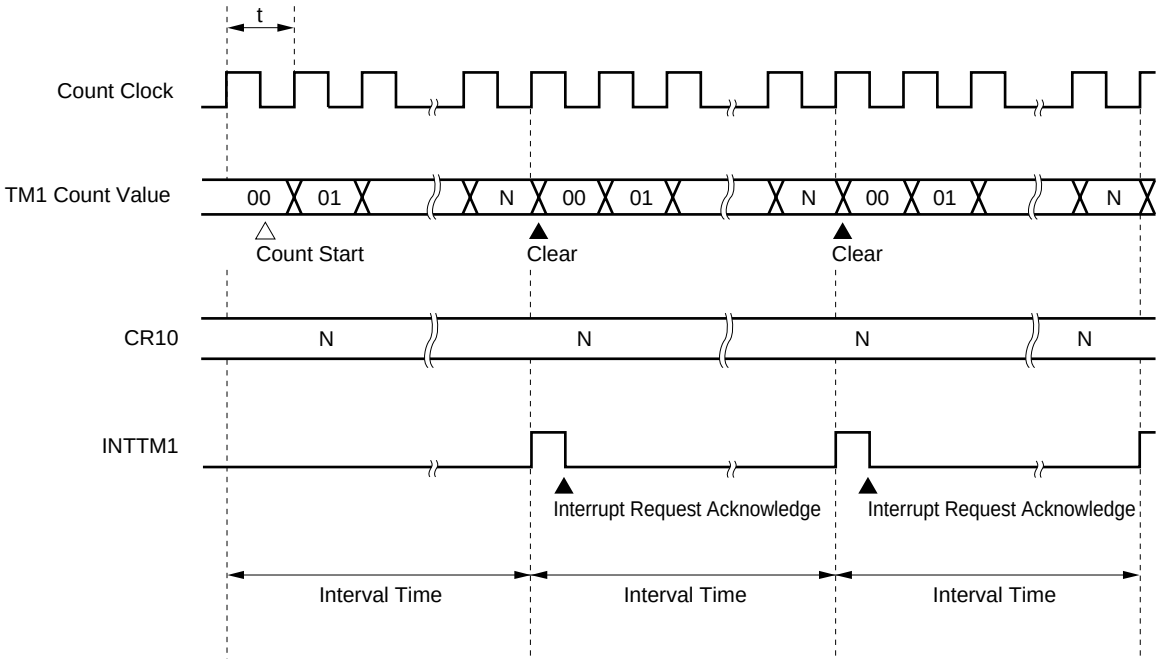
The 8-bit timer/event counters 1 and 2 operate as interval timers which generate interrupt requests repeatedly at intervals of the count value preset to 8-bit compare registers 10 and 20 (CR10 and CR20).

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 the 8-bit timer register 1 (TM1) can be selected with bits 0 to 3 (TCL10 to TCL13) of the timer clock select register 1 (TCL1). Count clock of the 8-bit timer register 2 (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 during timer count operation, refer to **6.5 Cautions on 8-Bit Timer/Event Counters (3)**.

Figure 6-4. Interval Timer Operation Timings



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

Table 6-4. 8-Bit Timer/Event Counter 1 Interval Time

TCL13	TCL12	TCL11	TCL10	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TI1 input cycle		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	0	0	1	TI1 input cycle		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)
0	1	1	1	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)
1	0	0	0	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)
1	0	0	1	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{13} \times 1/f_X$ (1.82 ms)	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)
1	0	1	0	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)	$2^{13} \times 1/f_X$ (1.82 ms)	$2^{14} \times 1/f_X$ (3.64 ms)	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)
1	0	1	1	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)	$2^{14} \times 1/f_X$ (3.64 ms)	$2^{15} \times 1/f_X$ (7.28 ms)	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)
1	1	0	0	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)	$2^{15} \times 1/f_X$ (7.28 ms)	$2^{16} \times 1/f_X$ (14.6 ms)	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)
1	1	0	1	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{16} \times 1/f_X$ (14.6 ms)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)
1	1	1	0	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)
1	1	1	1	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)
Other than above				Setting prohibited					

Remarks 1. f_X : System clock oscillation frequency

2. MCS : Oscillation mode selection register (OSMS) bit 0

3. TCL10-TCL13: Timer clock select register 1 (TCL1) bits 0-3

4. Values in parentheses when operated at f_X = 4.5 MHz.

Table 6-5. 8-Bit Timer/Event Counter 2 Interval Time

TCL17	TCL16	TCL15	TCL14	Minimum Interval Time		Maximum Interval Time		Resolution	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TI2 input cycle		$2^8 \times$ TI2 input cycle		TI2 input edge cycle	
0	0	0	1	TI2 input cycle		$2^8 \times$ TI2 input cycle		TI2 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)
0	1	1	1	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)
1	0	0	0	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)
1	0	0	1	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{13} \times 1/f_X$ (1.82 ms)	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)
1	0	1	0	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)	$2^{13} \times 1/f_X$ (1.82 ms)	$2^{14} \times 1/f_X$ (3.64 ms)	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)
1	0	1	1	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)	$2^{14} \times 1/f_X$ (3.64 ms)	$2^{15} \times 1/f_X$ (7.28 ms)	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)
1	1	0	0	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)	$2^{15} \times 1/f_X$ (7.28 ms)	$2^{16} \times 1/f_X$ (14.6 ms)	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)
1	1	0	1	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{16} \times 1/f_X$ (14.6 ms)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)
1	1	1	0	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)
1	1	1	1	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)
Other than above				Setting prohibited					

Remarks 1. f_X : System clock oscillation frequency

2. MCS : Oscillation mode selection register (OSMS) bit 0

3. TCL14-TCL17: Timer clock select register (TCL1) bits 4-7

4. Values in parentheses when operated at f_X = 4.5 MHz

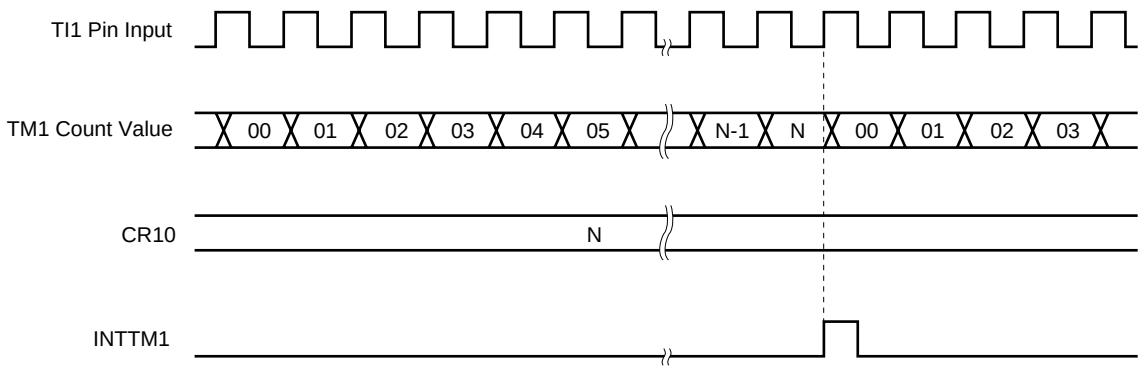
(2) External event counter operation

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 (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 6-5. External Event Counter Operation Timings (with Rising Edge Specified)



Remark N = 00H to FFH

6.4.2 16-bit timer/event counter mode

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

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

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

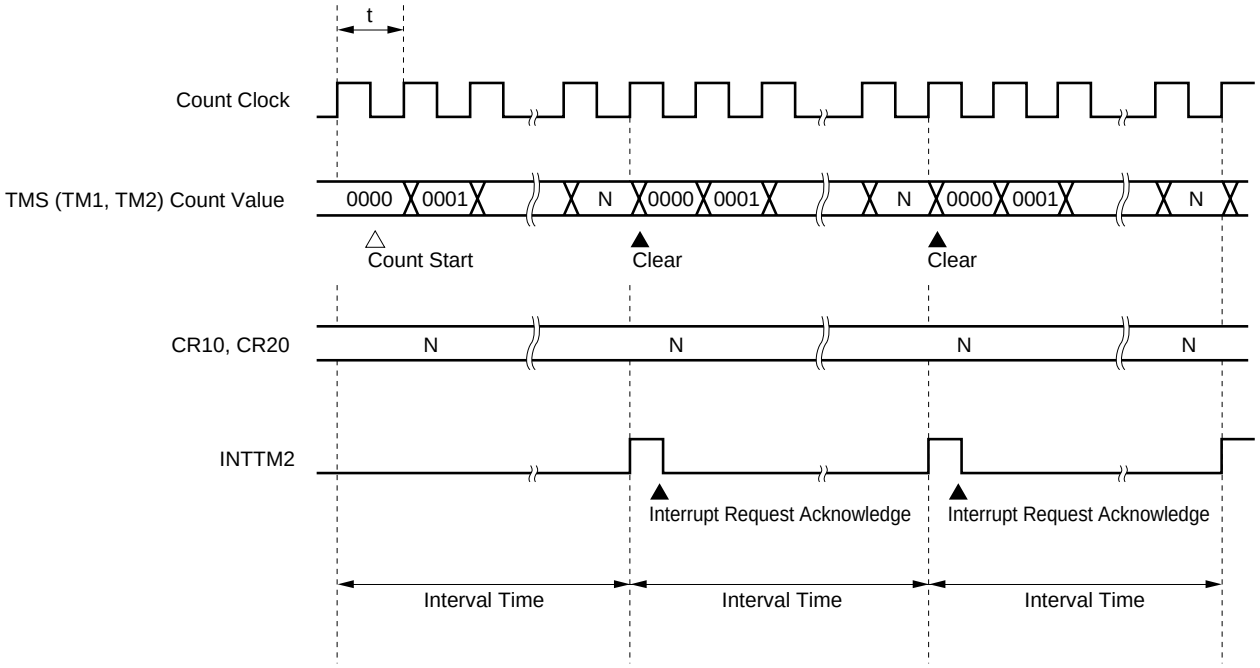
(1) Operation as interval timer

The 16-bit timer/event counter operates as an interval timer that repeatedly generates an interrupt request at interval specified by the count value assigned in advance to the 8-bit compare registers (CR10 and CR20) of the two channels. To set a count value, assigned the value of high-order 8 bits to CR20, and the value of the low-order 8 bits to CR10. For details of the count values (interval time) that can be set, refer to **Table 6-6**.

When the value of 8-bit timer register 1 (TM1) coincides with the value of CR10, and when the value of 8-bit timer register 2 (TM2) coincides with the value of CR20, the values of TM1 and TM2 are cleared to 0, and TM1 and TM2 continue counting. At the same time, an interrupt request signal is generated (INTTM2). For details of the operation timing of the interval timer, refer to **Figure 6-6**.

Bits 0 through 3 (TCL10 through TCL13) select the count clock of timer clock select register 1. The overflow signal of TM1 is used as the count clock to TM2.

Figure 6-6. Interval Timer Operation Timing



Remark Interval time = (N + 1) × t : 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. 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 the 16-bit timer (TMS) count value, use the 16-bit memory manipulation instruction.

Table 6-6. 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	
				MCS = 1	MCS = 0	MCS = 1	MCS = 0	MCS = 1	MCS = 0
0	0	0	0	TI1 input cycle		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	0	0	1	TI1 input cycle		$2^8 \times$ TI1 input cycle		TI1 input edge cycle	
0	1	1	0	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)	$2 \times 1/f_X$ (444 ns)	$2^2 \times 1/f_X$ (888 ns)
0	1	1	1	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)	$2^{18} \times 1/f_X$ (58.3 ms)	$2^{19} \times 1/f_X$ (116.5 ms)	$2^2 \times 1/f_X$ (888 ns)	$2^3 \times 1/f_X$ (1.78 μs)
1	0	0	0	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)	$2^3 \times 1/f_X$ (1.78 μs)	$2^4 \times 1/f_X$ (3.56 μs)
1	0	0	1	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)	$2^{20} \times 1/f_X$ (233.0 ms)	$2^{21} \times 1/f_X$ (466.0 ms)	$2^4 \times 1/f_X$ (3.56 μs)	$2^5 \times 1/f_X$ (7.11 μs)
1	0	1	0	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)	$2^{21} \times 1/f_X$ (466.0 ms)	$2^{22} \times 1/f_X$ (932.0 ms)	$2^5 \times 1/f_X$ (7.11 μs)	$2^6 \times 1/f_X$ (14.2 μs)
1	0	1	1	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)	$2^{22} \times 1/f_X$ (932.0 ms)	$2^{23} \times 1/f_X$ (1.9 s)	$2^6 \times 1/f_X$ (14.2 μs)	$2^7 \times 1/f_X$ (28.4 μs)
1	1	0	0	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)	$2^{23} \times 1/f_X$ (1.9 s)	$2^{24} \times 1/f_X$ (3.7 s)	$2^7 \times 1/f_X$ (28.4 μs)	$2^8 \times 1/f_X$ (56.9 μs)
1	1	0	1	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{24} \times 1/f_X$ (3.7 s)	$2^{25} \times 1/f_X$ (7.5 s)	$2^8 \times 1/f_X$ (56.9 μs)	$2^9 \times 1/f_X$ (113.8 μs)
1	1	1	0	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)	$2^{25} \times 1/f_X$ (7.5 s)	$2^{26} \times 1/f_X$ (14.9 s)	$2^9 \times 1/f_X$ (113.8 μs)	$2^{10} \times 1/f_X$ (227.5 μs)
1	1	1	1	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)	$2^{27} \times 1/f_X$ (29.8 s)	$2^{28} \times 1/f_X$ (59.7 s)	$2^{11} \times 1/f_X$ (455.1 μs)	$2^{12} \times 1/f_X$ (910.2 μs)
Other than above				Setting prohibited					

Remarks 1. f_X : System clock oscillation frequency

2. MCS : Oscillation mode selection register (OSMS) bit 0

3. TCL10-TCL13: Timer clock select register 1 (TCL1) bits 0-3

4. Values in parentheses when operated at f_X = 4.5 MHz.

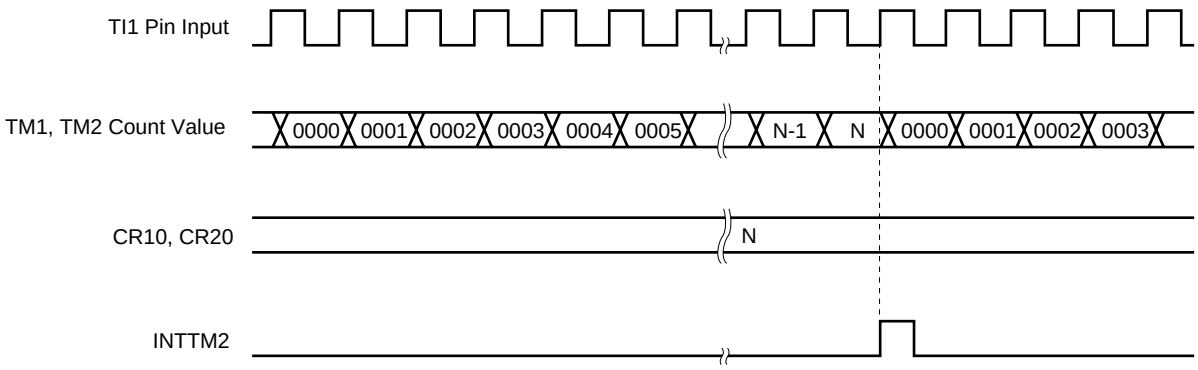
(2) External event counter operations

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

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

When the TM1 and TM2 counted values match the values of 8-bit compare registers 10 and 20 (CR10 and CR20), TM1 and TM2 are cleared to 0 and the interrupt request signal (INTTM2) is generated.

Figure 6-7. External Event Counter Operation Timings (with Rising Edge Specified)



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. 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.

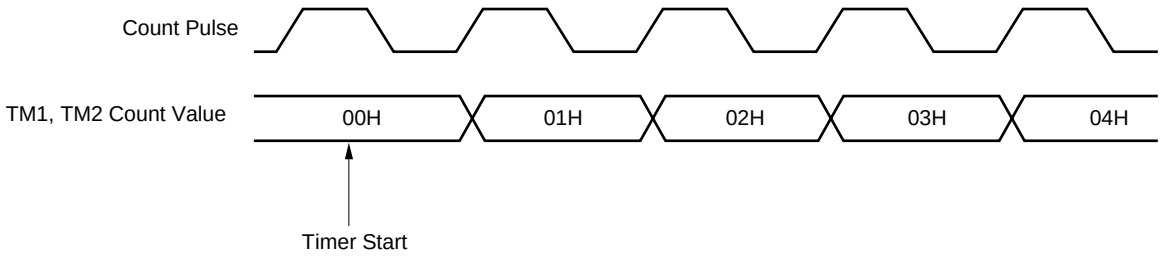
When reading the 16-bit timer (TMS) count value, use the 16-bit memory manipulation instruction.

6.5 Cautions on 8-Bit Timer/Event Counters

(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 6-8. 8-Bit Timer Registers 1 and 2 Start Timing



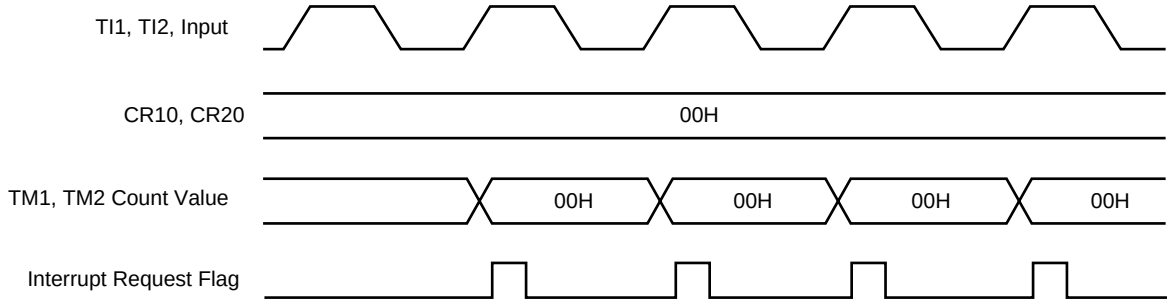
(2) Compare register 10 and 20 setting

The 8-bit compare registers 10 and 20 (CR10 and CR20) can be set to 00H.

Thus, when these 8-bit compare registers are used as event counters, 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 1 (TMC1) to 0 and stopping timer operation.

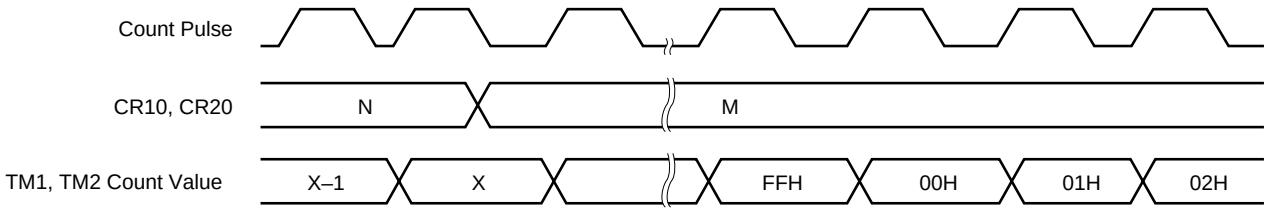
Figure 6-9. Event Counter Operation Timing



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

If the values after the 8-bit compare registers 10 and 20 (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 (M) after CR10 and CR20 change is smaller than value (N) before the change, it is necessary to restart the timer after changing CR10 and CR20.

Figure 6-10. Timing after Compare Register Change during Timer Count Operation



Remark N > X > M

CHAPTER 7 8-BIT TIMER

7.1 Function of 8-Bit Timer

The 8-bit timer has the following functions:

- Interval timer
- D/A converter (PWM output)

(1) Interval timer

The 8-bit timer is used as an 8-bit interval timer, and it counts the basic clock (1.125 MHz or 0.1125 MHz) with an 8-bit binary counter, and compares its count value with the set value of the PWM timer register. When the value set to the PWM timer register and the count value of the 8-bit binary counter match, an interrupt is generated.

Table 7-1. Interval Time of 8-Bit Timer

Minimum Interval Time	Maximum Interval Time	Resolution
0.889 μs	227.56 μs	0.889 μs
8.89 μs	2275.6 μs	8.89 μs

(2) D/A converter (PWM output)

The D/A converter outputs signals in the PWM (Pulse Width Modulation) mode in which the duty can be varied. Each pin of the D/A converter (PWM0 to PWM2) outputs a signal whose duty factor is variable.

The output frequency can be selected in four steps: 4.4 kHz, 440 Hz, 2.2 kHz, and 220 Hz. The duty factor can be varied in 256 or 512 steps.

By connecting an external lowpass filter to the D/A converter, digital signals can be converted into analog signals.

Table 7-2. Output Frequency and Duty Factor of D/A Converter (PWM Output)

Output Frequency	Duty Factor
4.40 kHz	256 steps
440 Hz	
2.20 kHz	512 steps
220 Hz	

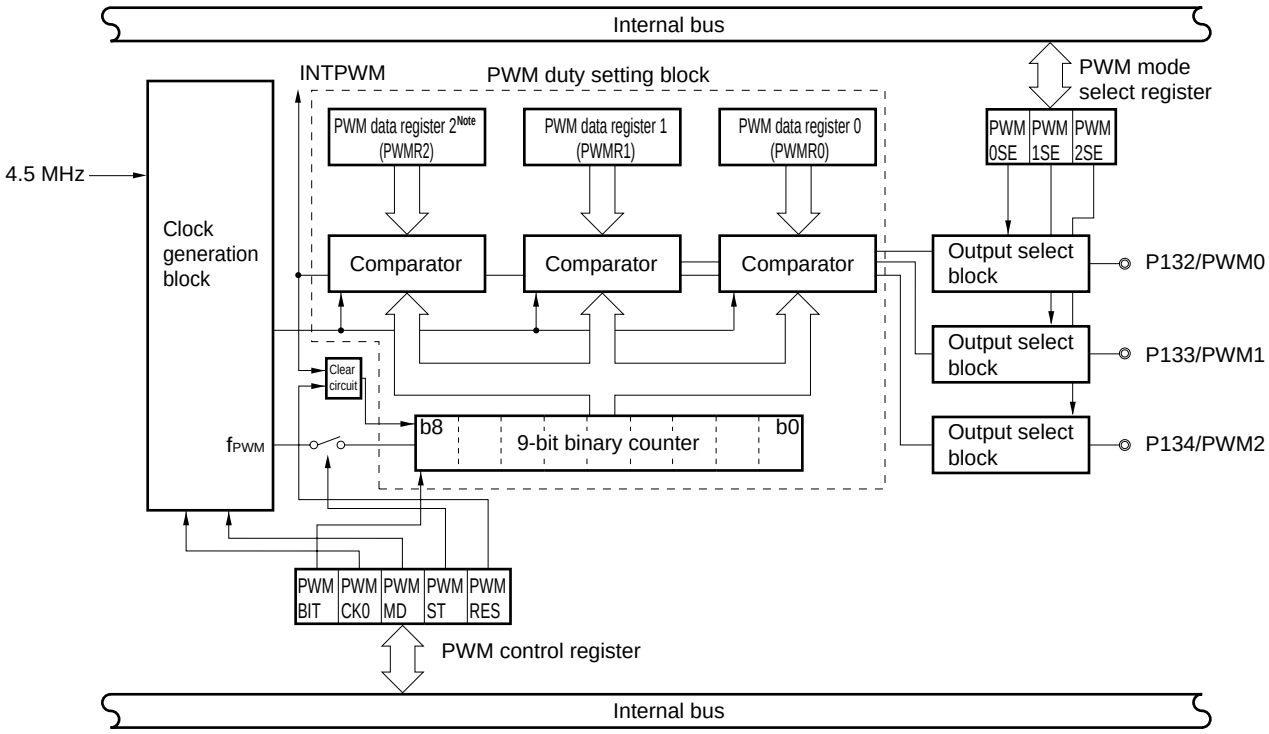
7.2 Configuration of 8-Bit Timer

The 8-bit timer consists of the following hardware units.

Table 7-3. 8-Bit Timer Configuration

Item	Configuration
Data register	PWM data register 0 (PWMR0) PWM data register 1 (PWMR1) PWM data register 2 (PWMR2) PWM timer register (PWMTMR): shared by PWMR2
Control register	PWM mode select register (PWMSEL) PWM control register (PWMCR)

Figure 7-1. 8-Bit Timer Block Diagram



Note The PWM data register 2 (PWMR2) is multiplexed with the PWM timer register (PWMTMR).

(1) PWM data registers (PWMR0 through PWMR2)

Each PWM data register (PWMR0 to PWMR2) sets the duty factor of the PWM output signal of the corresponding pin (PWM0 to PWM2).

Bits 9 through 15 of a PWM data register are fixed to 0 by hardware.

If the binary counter is set in the 8-bit mode, bits 0 through 7 of the PWM data register are valid, and the value of bit 8 is meaningless.

Because PWMR2 and PWM timer register (PWMTMR) are the same register, the D/A converter function and interval timer function cannot be used at the same time.

When PWMR2 is used as a PWM timer register, PWMR0 and PWMR1 can be used as 9-bit data latches.

The values of these PWM data registers are set to 1FFH at RESET and in the STOP mode. In the HALT mode, these registers retain the values immediately before the HALT mode is set.

(2) PWM timer register (PWMTMR)

The count value of the 8-bit binary counter is set to the PWM timer register (PWMTMR).

The PWM timer register is multiplexed with PWM data register 2. The PWM timer register is selected when bit 3 (PWMMD) of the PWM control register (PWMCR) is set to 1.

The value of the PWM timer register is set to FFH at RESET and in STOP mode. In the HALT mode, this register retains the value immediately before the HALT mode was set.

7.3 Registers Controlling 8-Bit Timer

The 8-bit timer is controlled by the following two registers.

- PWM mode select register (PWMSEL)
- PWM control register (PWMCR)

(1) PWM mode select register (PWMSEL)

This register selects whether the P132/PWM0 through P134/PWM2 pins are used in the general-purpose output port mode or PWM output mode.

Set these pins in the general-purpose output port mode when the 8-bit timer is used as an interval timer.

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

The value of this register is set to 00H at RESET and in STOP mode.

In the HALT mode, this register retains the value immediately before the HALT mode was set.

Figure 7-2. PWM Mode Select Register Format

Symbol	7	6	5	4	3	②	①	⓪	Address	After Reset	R/W
PWMSEL	0	0	0	0	0	PWM2SE	PWM1SE	PWM0SE	FFB0H	00H	R/W

PWM2SE	Selects Function of P133/PWM2 Pin
0	General-purpose output port
1	PWM output

PWM1SE	Selects Function of P133/PWM1 Pin
0	General-purpose output port
1	PWM output

PWM0SE	Selects Function of P132/PWM0 Pin
0	General-purpose output port
1	PWM output

Remark Bits 3 through 7 are fixed to 0 by hardware.

(2) PWM control register (PWMCR)

This register resets or starts the binary counter, selects the D/A converter (PWM output)/interval timer mode, selects the basic clock/PWM output frequency, and sets the number of bits of the binary counter.

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

The value of this register is set to 00H at RESET and in STOP mode.

In the HALT mode, this register retains the value immediately before the HALT mode was set.

Figure 7-3. PWM Control Register Format

Symbol	7	⑥	5	④	③	2	①	⓪	Address	After Reset
PWMCR	0	PWMBIT	0	PWMCK0	PWMMD	0	PWMST	PWMRES	FFB1H	00H
R/W	R	R/W	R	R/W	R/W	R	R/W	W		

R/W	PWMMD	Selects D/A Converter (PWM Output)/Interval Timer Mode
	0	D/A converter (PWM output)
	1	Interval timer

R/W	PWMST	Controls Counting of Binary Counter
	0	Stops counting of binary counter.
	1	Starts counting of binary counter.

W	PWMRES	Resets Binary Counter
	0	Nothing is affected.
	1	Resets binary counter.

- **In interval timer mode**

R/W	PWMCK0	Selects basic clock
	0	1.125 MHz
	1	0.1125 MHz

- **In D/A converter mode (PWM output)**

R/W	PWMBIT	Sets Number of Bits of Binary Counter (8 or 9 Bits)[Note]
	0	8 bits
	1	9 bits

R/W	PWMCK0	Selects PWM Output Fequency (Basic Clock) Common to PWM0 Through PWM2 Pins
	0	4.4 kHz (with binary counter set to 8 bits)/2.2 kHz (with binary counter set to 9 bits) (basic clock: 1.125 MHz)
	1	440 Hz (with binary counter set to 8 bits)/220 Hz (with binary counter set to 9 bits) (basic clock: 0.1125 kHz)

Note PWMBIT = 0 when PWMMD = 1.

Remark Bits 2, 5, and 7 are fixed to 0 by hardware.

7.4 Operation of 8-Bit Timer

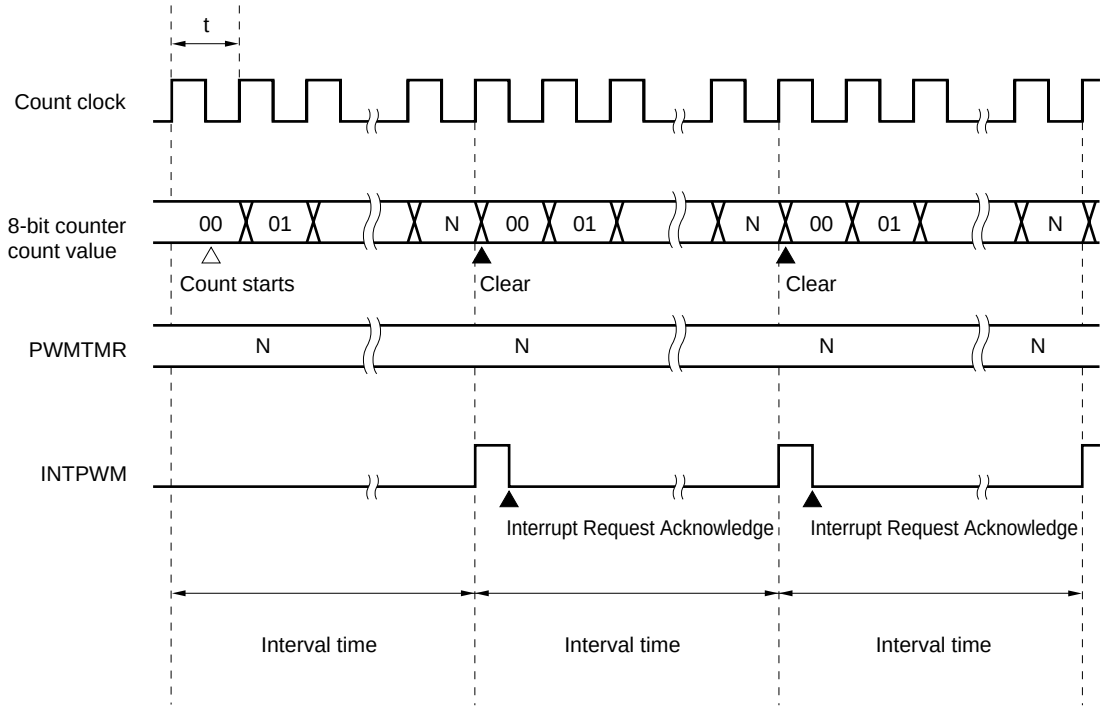
7.4.1 Operation as interval timer

When the 8-bit timer is used as an interval timer, it repeatedly generates an interrupt at intervals specified by the count value set in advance to the PWM timer register (PWMTMR).

When the count value of the 8-bit binary counter matches with the value set to PWMTMR, the binary counter is reset to 0 and then continues counting. At the same time, an interrupt request signal (INTPWM) is generated.

Bit 4 of the PWM control register (PWMCR) is used to select a basic clock (count clock).

Figure 7-4. Timing of Interval Timer Operation



Remark Interval time $T = (N + 1) \times t$

where, $N = 0$ to 255

$t = 1/1.125$ MHz or $1/0.1125$ MHz

7.4.2 Operation as D/A converter (PWM output)

(1) Operation of output select block

Figure 7-5 shows the configuration of the output select block.

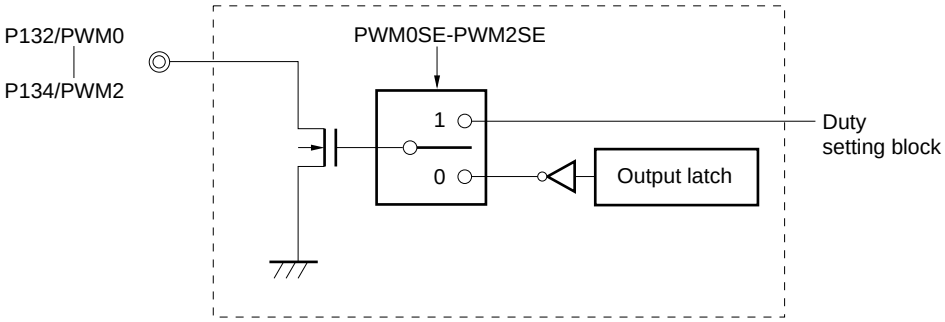
The output select block selects whether each pin (P132/PWM0 through P134/PWM2) is used in the general-purpose output port mode or D/A converter output (PWM output) mode.

Each pin can be set in either of these modes by using the corresponding bit (PWM0SE to PWM2SE) of the PWM mode select register (PWMSEL).

Because each pin is an N-ch open-drain output pin, an external pull-up resistor must be connected.

When the 8-bit timer is used as an interval timer, an undefined value is output as PWM output. Therefore, set these pins in the general-purpose output port mode (P132 through P134).

Figure 7-5. Output Select Block Configuration



(2) Operation of clock generation block

The clock generation block outputs the count clock for the 8-bit binary counter, and the basic clock by which the duty factor of each output signal (PWM0 to PWM2) is set. The basic clock frequency (f_{PWM}) is 1.125 or 0.1125 MHz.

When bit 4 (PWMCK0) of the PWM control register is 0, the basic clock frequency is 1.125 MHz; when this bit is 1, the frequency is 0.1125 MHz.

(3) Operation of PWM duty setting block

For the PWM output, the duty factor can be changed in 256 steps when the number of bits of the binary counter is set to 8, and in 512 steps when the binary counter is set to 9 bits.

The value set to each PWM data register (PWMR0 to PWMR2) is compared with the value of the basic clock counted by the 8-/9-bit binary counter. If the value of the PWM data register is greater than the set value of the PWM data register, a high level is output. If the value of PWM data register is less than that of the PWM data register, a low level is output.

Each pin can output a signal with an independent duty factor according to the data set to the corresponding PWM data register. Note, however, that the basic clock, the number of bits of the binary counter, and the output frequency are the same for each pin. This means that each pin cannot output a signal at a different cycle.

(a) PWM duty

The duty factor is as follows where the value set to the PWM data register is "x".

- **Duty when binary counter is set to 8 bits**

  Duty: $D = \frac{x + 0.25}{256} \times 100\%$ x = integer from 0 to 255

- **Duty when binary counter is set to 9 bits**

  Duty: $D = \frac{x + 0.25}{512} \times 100\%$ x = integer from 0 to 511

In the above expressions, 0.25 is an offset. The high level is output even when x = 0.

(b) Output frequency and cycle

Because the basic clock frequency is 1.125 or 0.1125 MHz, the frequency and cycle of the output signal are as follows.

(i) When binary counter is set to 8 bits

- When basic clock frequency is set to 1.125 MHz (PWMCK0 = 0)

  Frequency: $f = \frac{1.125\ \text{MHz}}{256} = 4.3945\ \text{kHz}$

  Cycle: $T = \frac{256}{1.125\ \text{MHz}} = 227.56\ \mu\text{s}$

- When basic clock frequency is set to 0.1125 MHz (PWMCK0 = 1)

  Frequency: $f = \frac{0.1125\ \text{MHz}}{256} = 439.45\ \text{Hz}$

  Cycle: $T = \frac{256}{0.1125\ \text{MHz}} = 2.2756\ \text{ms}$

(ii) When binary counter is set to 9 bits

- When basic clock frequency is set to 1.125 MHz (PWMCK0 = 0)

  Frequency: $f = \frac{1.125\ \text{MHz}}{512} = 2.1972\ \text{kHz}$

  Cycle: $T = \frac{512}{1.125\ \text{MHz}} = 455.11\ \mu\text{s}$

- When basic clock frequency is set to 0.1125 MHz (PWMCK0 = 1)

  Frequency: $f = \frac{0.1125\ \text{MHz}}{512} = 219.73\ \text{Hz}$

  Cycle: $T = \frac{512}{0.1125\ \text{MHz}} = 4.5511\ \mu\text{s}$

(4) PWM output wave form

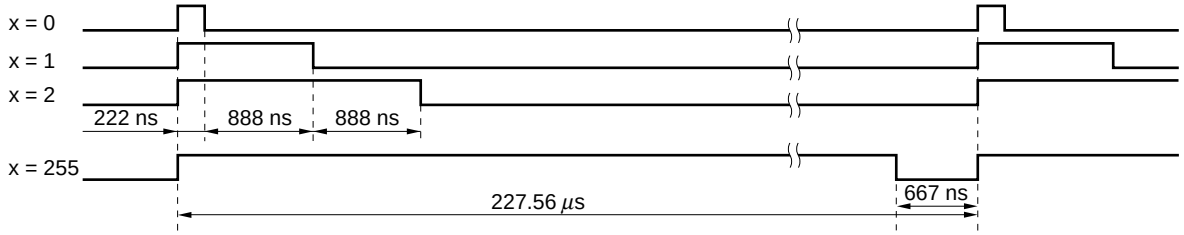
(a) Duty of PWM output waveform

Figure 7-6 shows the relationship between the set value x of the PWM data register and output waveform. Figure 7-7 shows the relations of output waveforms among the respective pins.

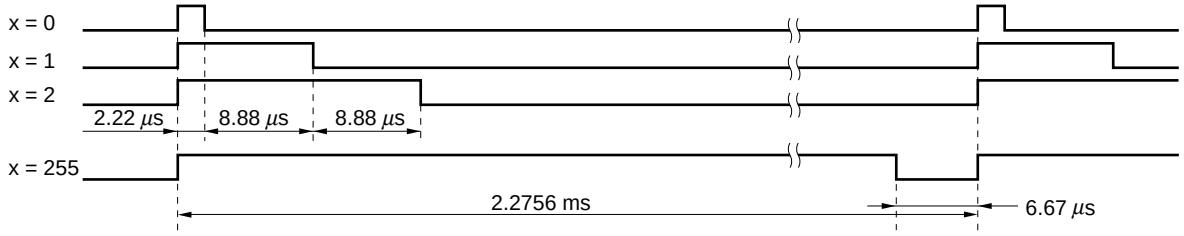
Figure 7-6. Set Value x of PWM Data Register and Output Waveform

(a) When binary counter is set to 8 bits (PWMBIT = 0)

(i) When basic clock frequency is set to 1.125 MHz (output frequency: 4.4 kHz) (PWMCK0 = 0)

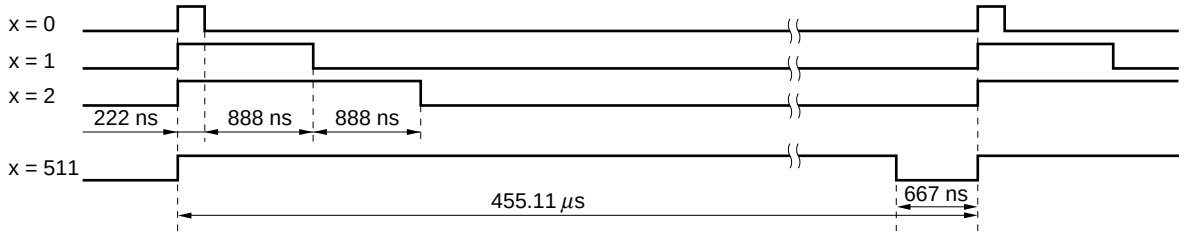


(ii) When basic clock frequency is set to 0.1125 MHz (output frequency: 440 Hz) (PWMCK0 = 1)



(b) When binary counter is set to 9 bits (PWMBIT = 1)

(i) When basic clock frequency is set to 1.125 MHz (output frequency: 2.2 kHz) (PWMCK0 = 0)



(ii) When basic clock frequency is set to 0.1125 MHz (output frequency: 220 kHz) (PWMCK0 = 1)

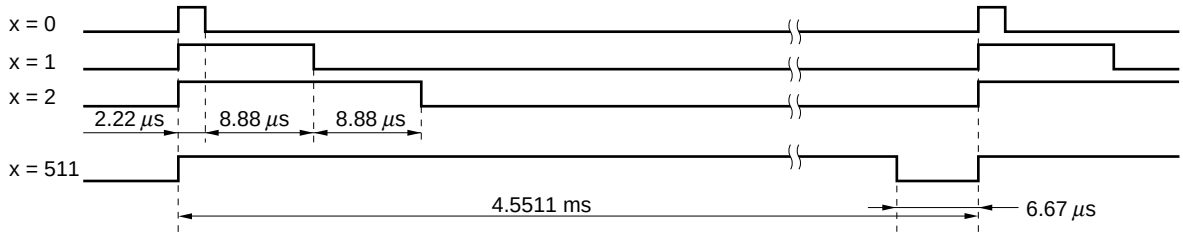
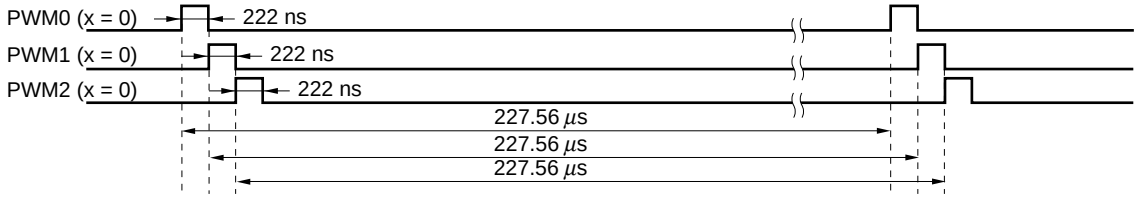


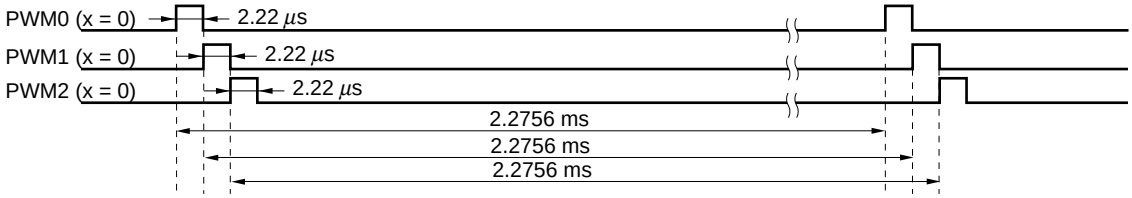
Figure 7-7. Relationship of Output Waveforms among Respective Pins

(a) When binary counter is set to 8 bits (PWMBIT = 0, x = 0)

(i) When basic clock frequency is set to 1.125 MHz (output frequency: 4.4 kHz) (PWMCK0 = 0)

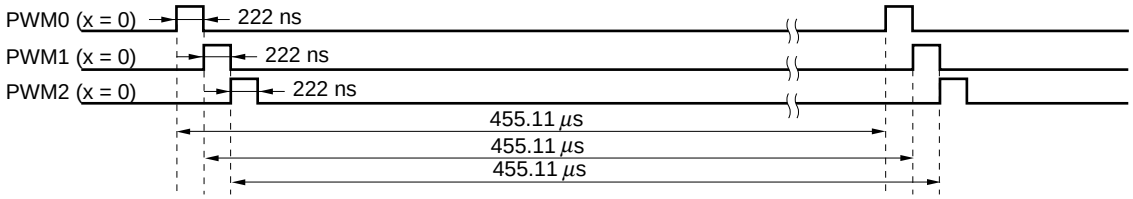


(ii) When basic clock frequency is set to 0.1125 MHz (output frequency: 440 Hz) (PWMCK0 = 1)

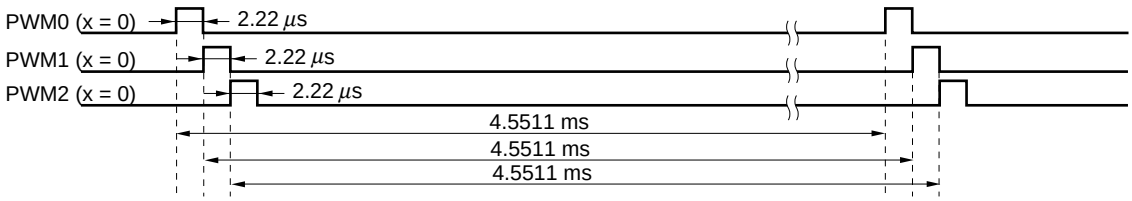


(b) When binary counter is set to 9 bits (PWMBIT = 1, x = 0)

(i) When basic clock frequency is set to 1.125 MHz (output frequency: 2.2 kHz) (PWMCK0 = 0)



(ii) When basic clock frequency is set to 0.1125 MHz (output frequency: 220 kHz) (PWMCK0 = 1)



7.5 Notes on 8-Bit Timer

(1) Notes on using 8-bit timer

(a) In interval timer mode

- Be sure to reset the binary counter (PWMRES = 1) before operating the timer.
- Do not rewrite PWMCK0 and PWMMD of the PWM control register (PWMCR) while the timer is operating.
- Do not rewrite the set value of the PWM timer register while the timer is operating.
- Set the P132/PWM0 through P134/PWM2 pins in the general-purpose output port mode when using the timer (PWM0SE = PWM1SE = PWM2SE of PWM mode select register = 0).
- When the interval timer mode is set (PWMMD = 1), the binary counter is set to 8 bits. As a result, the value of PWMBIT of the PWM control register (PWMCR) becomes meaningless.
- The 8-bit timer cannot be used in the D/A converter mode (for PWM output) while it is being used as an interval timer.

(b) In D/A converter mode (PWM output)

- Do not rewrite PWMBIT, PWMCK0, and PWMMD of the PWM control register (PWMCR) while the 8-bit timer is operating.
- The timer cannot be used as an interval timer while it is being used in the D/A converter mode (for PWM output).

(2) Status at RESET and in standby mode

(a) At RESET and in STOP mode

The P132/PWM0 through P134/PWM2 pins are set in the general-purpose output port mode, and output a low level (output latch = 0).
The value of the PWM timer register (PWMTMR) is set to FFH.
The value of each PWM data register (PWMR0 through PWMR2) is set to 1FFH.
The 8-bit timer stops, if it has been operating, at RESET and in STOP mode, and is set in the D/A converter (PWM output) mode.

(b) In HALT mode

The P132/PWM0 through P134/PWM2 pins hold the set status.
The PWM timer register (PWMTMR) holds the previous value.
The PWM control register holds the previous value. Even if the HALT mode has been released, it holds the set value (the basic clock cannot be stopped by the HALT instruction).
While the timer is used as an interval timer, it continues counting. When the timer is used as a D/A converter (PWM output), it continues PWM output as is.

[MEMO]

CHAPTER 8 BASIC TIMER

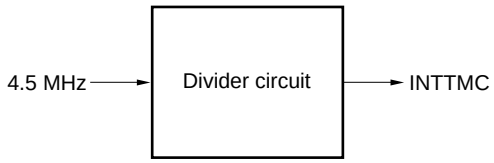
The basic timer is used for time management during program execution.

8.1 Function of Basic Timer

The basic timer generates an interrupt request signal (INTTMC) at time intervals of 100 ms.

8.2 Configuration of Basic Timer

Figure 8-1. Basic Timer Block Diagram



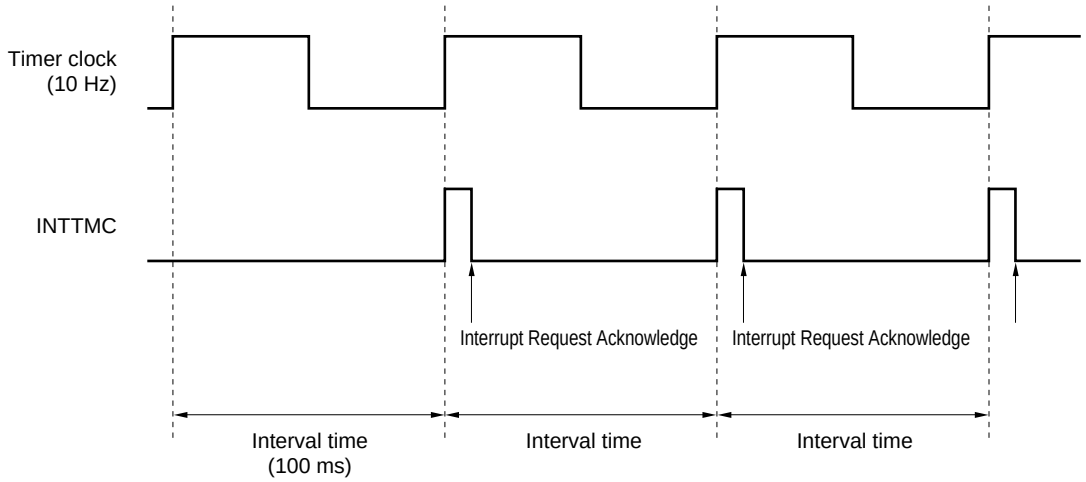
8.3 Operation of Basic Timer

An example of the operation of the basic timer is shown below.

In this example, the basic timer operates as an interval timer that repeatedly generates an interrupt at time intervals of 100 ms. Interrupt request signal INTTMC is generated every 100 ms.

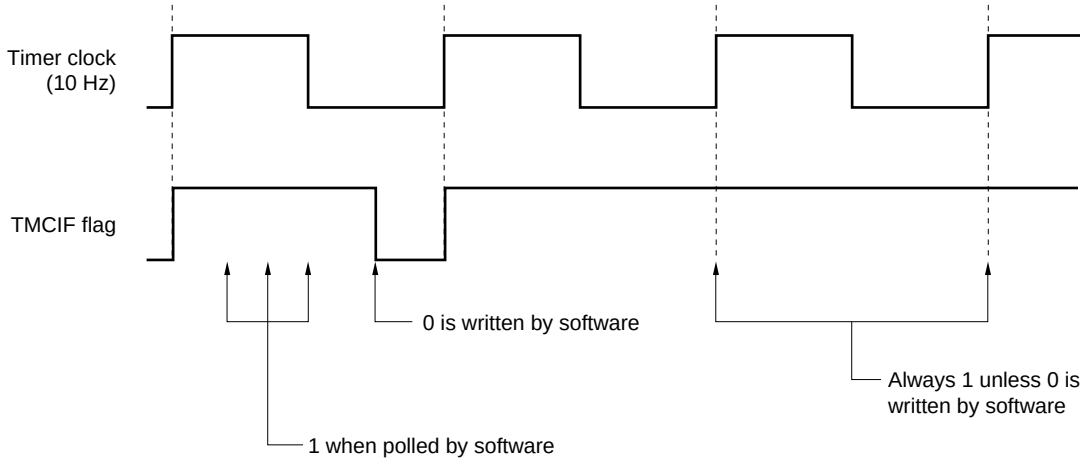
The timer clock frequency is 10 Hz.

Figure 8-2. Basic Timer Operation Timing



By polling the interrupt request flag (TMCIF) of this basic timer by software, time management can be carried out. Note that TMCIF is not a Read & Reset flag.

Figure 8-3. Operating Timing to Poll TMCIF Flag



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 simultaneously.)

(1) Watchdog timer mode

An inadvertent program loop is detected. Upon detection of the inadvertent program loop, a non-maskable interrupt request or reset can be generated.

Table 9-1. Watchdog Timer Inadvertent Program Loop Detection Times

Inadvertent Program Loop Detection Time	MCS = 1	MCS = 0
$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (455 μs)	$2^{12} \times 1/f_X$ (910 μs)
$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_X$ (910 μs)	$2^{13} \times 1/f_X$ (1.82 ms)
$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_X$ (1.82 ms)	$2^{14} \times 1/f_X$ (3.64 ms)
$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_X$ (3.64 ms)	$2^{15} \times 1/f_X$ (7.28 ms)
$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_X$ (7.28 ms)	$2^{16} \times 1/f_X$ (14.6 ms)
$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_X$ (14.6 ms)	$2^{17} \times 1/f_X$ (29.1 ms)
$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)
$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. MCS : Oscillation mode selection register (OSMS) bit 0

4. Figures in parentheses apply to operation with f_X = 4.5 MHz.

(2) Interval timer mode

Interrupt requests are generated at the preset time intervals.

Table 9-2. Interval Times

Interval Time	MCS = 1	CS = 0
$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (455 μs)	$2^{12} \times 1/f_X$ (910 μs)
$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_X$ (910 μs)	$2^{13} \times 1/f_X$ (1.82 ms)
$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_X$ (1.82 ms)	$2^{14} \times 1/f_X$ (3.64 ms)
$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_X$ (3.64 ms)	$2^{15} \times 1/f_X$ (7.28 ms)
$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_X$ (7.28 ms)	$2^{16} \times 1/f_X$ (14.6 ms)
$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_X$ (14.6 ms)	$2^{17} \times 1/f_X$ (29.1 ms)
$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)
$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. MCS : Oscillation mode selection register (OSMS) bit 0

4. Figures in parentheses apply to operation with f_X = 4.5 MHz.

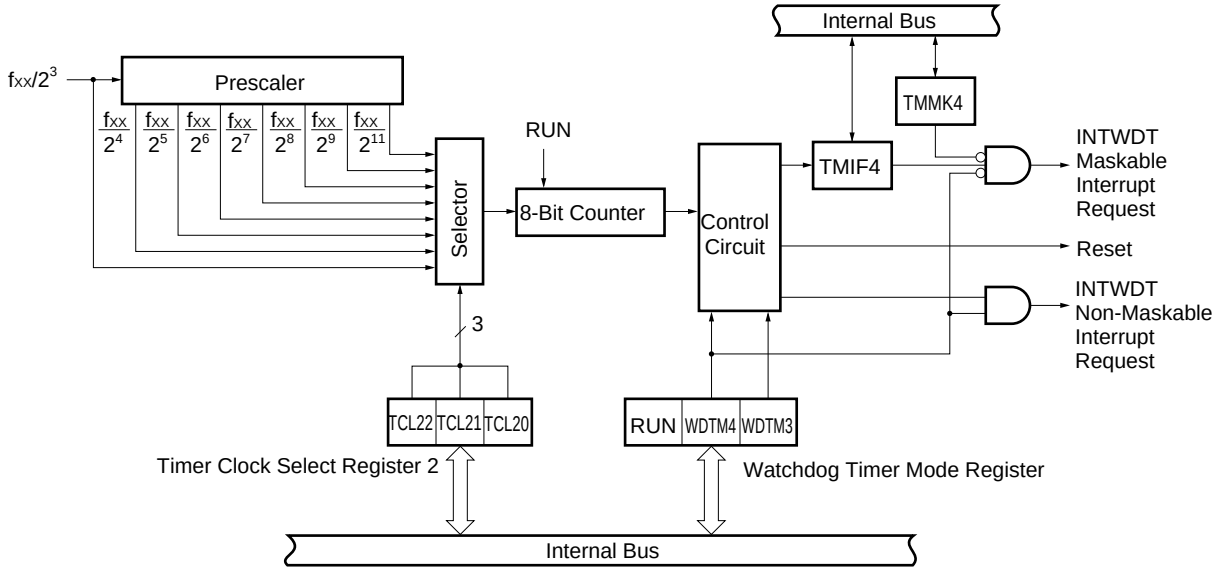
9.2 Watchdog Timer Configuration

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 control 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 8-bit memory manipulation instruction.

Reset input sets TCL2 to 00H.

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

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	0	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection MCS = 0 or 1
0	×	×	Buzzer output disable (port output)
1	0	0	6 kHz
1	0	1	3 kHz
1	1	0	1.5 kHz
1	1	1	Setting prohibited

TCL22	TCL21	TCL20	Watchdog Timer Count Clock Selection		
				MCS = 1	MCS = 0
0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (563 kHz)	$f_X/2^4$ (281 kHz)
0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (281 kHz)	$f_X/2^5$ (141 kHz)
0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (141 kHz)	$f_X/2^6$ (70.3 kHz)
0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (70.3 kHz)	$f_X/2^7$ (35.2 kHz)
1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (35.2 kHz)	$f_X/2^8$ (17.6 kHz)
1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (17.6 kHz)	$f_X/2^9$ (8.8 kHz)
1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (8.8 kHz)	$f_X/2^{10}$ (4.4 kHz)
1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.2 kHz)	$f_X/2^{12}$ (1.1 kHz)

Caution When rewriting TCL2 to other data, stop the timer operation beforehand.

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. × : Don't care

4. MCS : Oscillation mode selection register (OSMS) bit 0

5. Figures in parentheses apply to operation with f_X = 4.5 MHz.

(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.

Reset input sets 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	RUM	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

WDTM4	WDTM3	Watchdog Timer Operation Mode Selection[Note 1]
0	×	Interval timer mode[Note 2] (Maskable interrupt occurs upon generation of an overflow.)
1	0	Watchdog timer mode 1 (Non-maskable interrupt 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 Mode 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. WDTM starts interval timer operation at a time RUN is set to 1.

3. Once set to 1, RUN cannot be cleared to 0 by software. Therefore, use reset function to clear RUN to 0.

Cautions 1. When 1 is set in RUN so that the watchdog timer is cleared, the actual overflow time is up to 0.5 % shorter than the time set by timer clock select register 2.

2. When using watchdog timer modes 1 and 2, confirm that the interrupt request flag (TMIF4) is 0, and then set WDTM4 to 1. If WDTM4 is set to 1 when TMIF4 is 1, the non-maskable interrupt request occurs 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 inadvertent program loop time 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.

Caution The actual inadvertent program loop detection time may be shorter than the set time by a maximum of 0.5 %.

Table 9-4. Watchdog Timer Inadvertent Program Loop Detection Time

TCL22	TCL21	TCL20	Inadvertent Program Loop Detection Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (455 μs)	$2^{12} \times 1/f_X$ (910 μs)
0	0	1	$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_X$ (910 μs)	$2^{13} \times 1/f_X$ (1.82 ms)
0	1	0	$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_X$ (1.82 ms)	$2^{14} \times 1/f_X$ (3.64 ms)
0	1	1	$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_X$ (64 ms)	$2^{15} \times 1/f_X$ (7.28 ms)
1	0	0	$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_X$ (7.28 ms)	$2^{16} \times 1/f_X$ (14.6 ms)
1	0	1	$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_X$ (14.6 ms)	$2^{17} \times 1/f_X$ (29.1 ms)
1	1	0	$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)
1	1	1	$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. MCS : Oscillation mode selection register (OSMS) bit 0

4. Figures in parentheses apply to operation with f_X = 4.5 MHz.

9.4.2 Interval timer operation

The watchdog timer operates as an interval timer which generates interrupt requests repeatedly at an interval of the preset count value when bit 4 (WDTM4) of the watchdog timer mode register (WDTM) is set to 0.

A count clock (interval time) can be selected by using bits 0 through 2 (TCL20 through TCL22) of the timer clock select register 2 (TCL2). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer starts operating as an interval timer.

When the watchdog timer operated as interval timer, the interrupt mask flag (TMMK4) and priority specification flag (TMPR4) are validated and the maskable request interrupt (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 STOP mode. Thus, set bit 7 of WDTM (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 reset is applied.

2. The interval time just after setting with WDTM may be shorter than the set time by a maximum of 0.5 %.

Table 9-5. Interval Timer Interval Time

TCL22	TCL21	TCL20	Interval Time	MCS = 1	MCS = 0
0	0	0	$2^{11} \times 1/f_{XX}$	$2^{11} \times 1/f_X$ (455 μs)	$2^{12} \times 1/f_X$ (910 μs)
0	0	1	$2^{12} \times 1/f_{XX}$	$2^{12} \times 1/f_X$ (910 μs)	$2^{13} \times 1/f_X$ (1.82 ms)
0	1	0	$2^{13} \times 1/f_{XX}$	$2^{13} \times 1/f_X$ (1.82 ms)	$2^{14} \times 1/f_X$ (3.64 ms)
0	1	1	$2^{14} \times 1/f_{XX}$	$2^{14} \times 1/f_X$ (3.64 ms)	$2^{15} \times 1/f_X$ (7.28 ms)
1	0	0	$2^{15} \times 1/f_{XX}$	$2^{15} \times 1/f_X$ (7.28 ms)	$2^{16} \times 1/f_X$ (14.6 ms)
1	0	1	$2^{16} \times 1/f_{XX}$	$2^{16} \times 1/f_X$ (14.6 ms)	$2^{17} \times 1/f_X$ (29.1 ms)
1	1	0	$2^{17} \times 1/f_{XX}$	$2^{17} \times 1/f_X$ (29.1 ms)	$2^{18} \times 1/f_X$ (58.3 ms)
1	1	1	$2^{19} \times 1/f_{XX}$	$2^{19} \times 1/f_X$ (116.5 ms)	$2^{20} \times 1/f_X$ (233.0 ms)

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. MCS : Oscillation mode selection register (OSMS) bit 0

4. Figures in parentheses apply to operation with f_X = 4.5 MHz.

CHAPTER 10 BUZZER OUTPUT CONTROL CIRCUIT

10.1 Buzzer Output Control Circuit Functions

The buzzer output control circuit outputs 1.5 kHz, 3 kHz, or 6 kHz frequency square waves. The buzzer frequency selected with timer clock select register 2 (TCL2) is output from the BEEP/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 to 0 (Set to output mode).

Caution Buzzer output cannot be used when setting P36 output latch to 1.

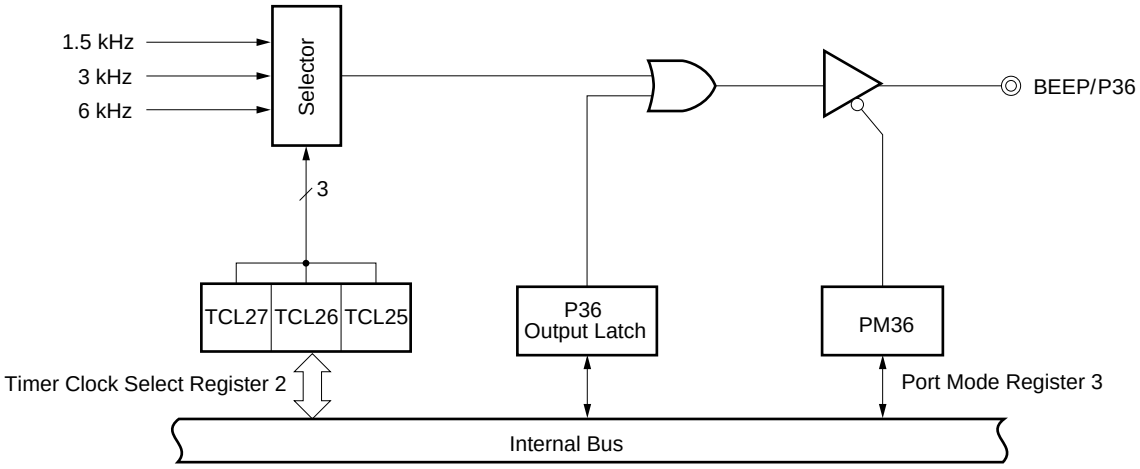
10.2 Buzzer Output Control Circuit Configuration

The buzzer output control circuit consists of the following hardware.

Table 10-1. Buzzer Output Control Circuit Configuration

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

Figure 10-1. Buzzer Output Control Circuit Block Diagram



10.3 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.

Reset input sets TCL2 to 00H.

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

Figure 10-2. 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

TCL27	TCL26	TCL25	Buzzer Output Frequency Selection MCS = 0 or 1
0	×	×	Buzzer output disable (port output)
1	0	0	6 kHz
1	0	1	3 kHz
1	1	0	1.5 kHz
1	1	1	Setting prohibited

TCL22	TCL21	TCL20	Watchdog Timer Count Clock Selection		
				MCS = 1	MCS = 0
0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (563 kHz)	$f_X/2^4$ (281 kHz)
0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (281 kHz)	$f_X/2^5$ (141 kHz)
0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (141 kHz)	$f_X/2^6$ (70.3 kHz)
0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (70.3 kHz)	$f_X/2^7$ (35.2 kHz)
1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (35.2 kHz)	$f_X/2^8$ (17.6 kHz)
1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (17.6 kHz)	$f_X/2^9$ (8.8 kHz)
1	1	0	$f_{XX}/2^9$	$f_X/2^9$ (8.8 kHz)	$f_X/2^{10}$ (4.4 kHz)
1	1	1	$f_{XX}/2^{11}$	$f_X/2^{11}$ (2.2 kHz)	$f_X/2^{12}$ (1.1 kHz)

Caution When rewriting TCL2 to other data, stop the timer operation beforehand.

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. × : don't care

4. MCS : Oscillation mode selection register (OSMS) bit 0

5. Figures in parentheses apply to operation with f_X = 4.5 MHz.

(2) Port mode register 3 (PM3)

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

When using the P36/BEEP 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.

Reset input sets PM3 to FFH.

Figure 10-3. 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 11 A/D CONVERTER

11.1 A/D Converter Functions

The A/D converter converts an analog input into a digital value. It consists of 6 channels (ANI0 to ANI5) 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 (fall of INTP3).

(2) Software start

Conversion is started by setting the A/D converter mode register (ADM).

Select one channel of analog input from ANI0 to ANI5 and carry out A/D conversion. In the case of hardware start, A/D conversion operation stops when an A/D conversion operation ends, 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.

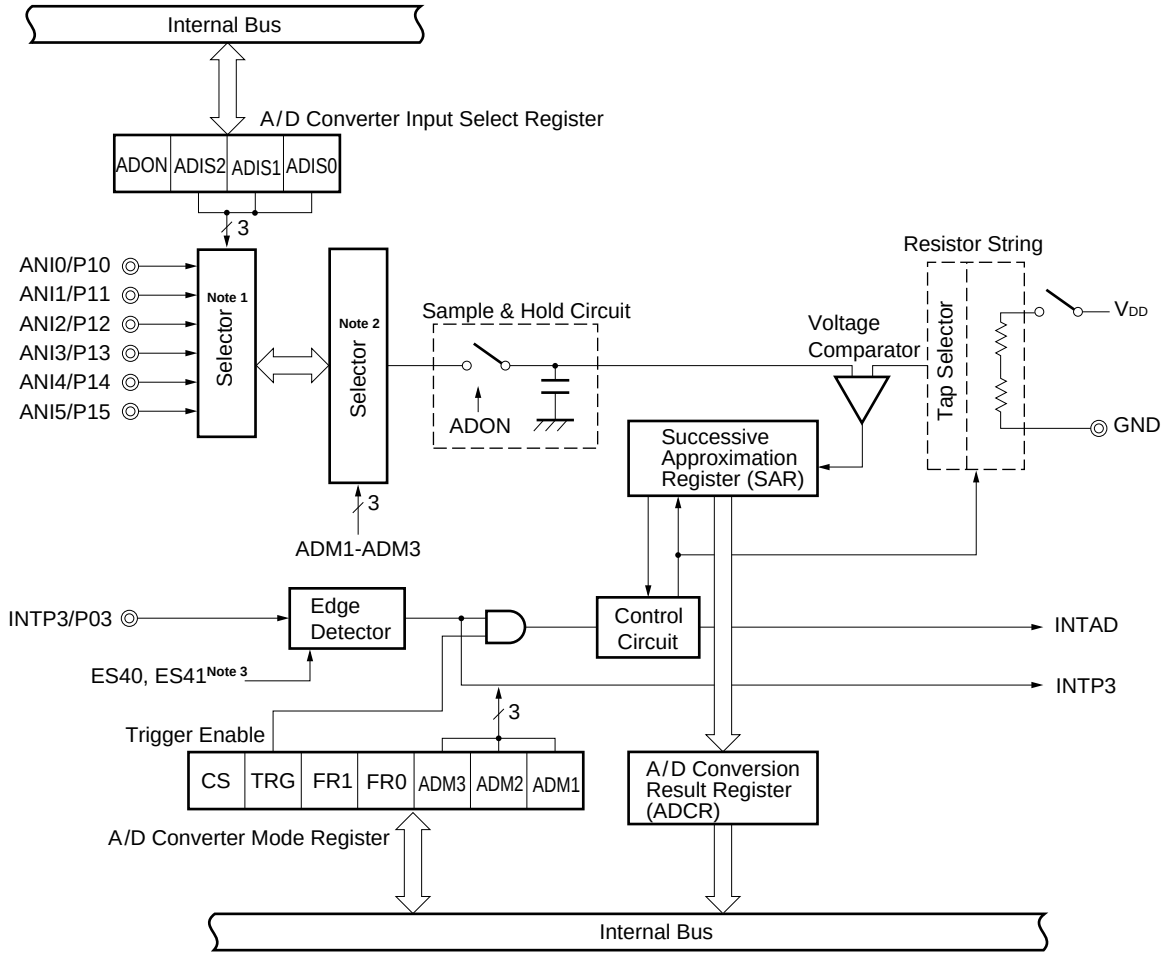
11.2 A/D Converter Configuration

The A/D converter consists of the following hardware.

Table 11-1. A/D Converter Configuration

Item	Configuration
Analog input	6 channels (ANI0 to ANI5)
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 11-1. 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.

3. Bits 0 and 1 of the external interrupt mode register 1 (INTM1) (refer to **Figure 14-6**).

(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 set (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.

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) Resistor string

The resistor string is connected between V_{DD} and GND, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are 6-channel analog input pins to input analog signals to undergo A/D conversion to the A/D converter. Pins other than those selected as analog input by the A/D converter input select register (ADIS) can be used as input/output ports.

Caution Use ANI0 to ANI5 input voltages within the specified range. If a voltage higher than V_{DD} or lower than GND 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.

11.3 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.

Reset input sets ADM to 01H.

Figure 11-2. 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

CS	A/D Conversion Operation Control
0	Operation stop
1	Operation start

TRG	External Trigger Selection
0	No external trigger (software starts)
1	Conversion started by external trigger (hardware starts) A/D conversion is started at the falling edge of the INTP3 signal.

FR1	FR0	HSC	A/D Conversion Time Selection[Note 1] f_X = 4.5 MHz Operation MCS = 0 or 1
0	0	1	160/f_X (35.6 μs)
0	1	1	80/f_X (Setting prohibited[Note 2])
1	0	0	100/f_X (22.2 μs)
1	0	1	200/f_X (44.4 μs)
Other than above			Setting prohibited

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
Other than above			Setting prohibited

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. The following sequence is recommended for power consumption reduction of A/D converter when the standby function is used: Clear bit 7 (CS) to 0 first to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

2. When restarting the stopped A/D conversion operation, start the A/D conversion operation after clearing the interrupt request flag (ADIF) to 0.

Remarks 1. f_X : System clock oscillation frequency

2. MCS : Oscillation mode selection register (OSMS) bit 0

(2) A/D converter input select register (ADIS)

This register determines whether the ANI0/P10 to ANI5/P15 pins should be used for analog input channels or ports. Pins other than those selected as analog input can be used as input/output ports.

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

Reset input sets 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 from among the channels set for analog input with ADIS.

2. When bit 3 (ADON) of ADIS is set to 1, the current consumption of the A/D converter increases in the standby mode. Clear ADON to 0 while A/D conversion is stopped.

Figure 11-3. 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	ADON	ADIS2	ADIS1	ADIS0	FF84H	00H	R/W

ADON	Control of Resistor String Power Supply
0	Power OFF
1	Power ON

ADIS2	ADIS1	ADIS0	Number of Analog Input Channel Selection
0	0	0	No analog input channel (P10-P15)
0	0	1	1 channel (ANI0, P11-P15)
0	1	0	2 channel (ANI0, ANI1, P12-P15)
0	1	1	3 channel (ANI0-ANI2, P13-P15)
1	0	0	4 channel (ANI0-ANI3, P14, P15)
1	0	1	5 channel (ANI0-ANI4, P15)
1	1	0	6 channel (ANI0-ANI5)
Other than above			Setting prohibited

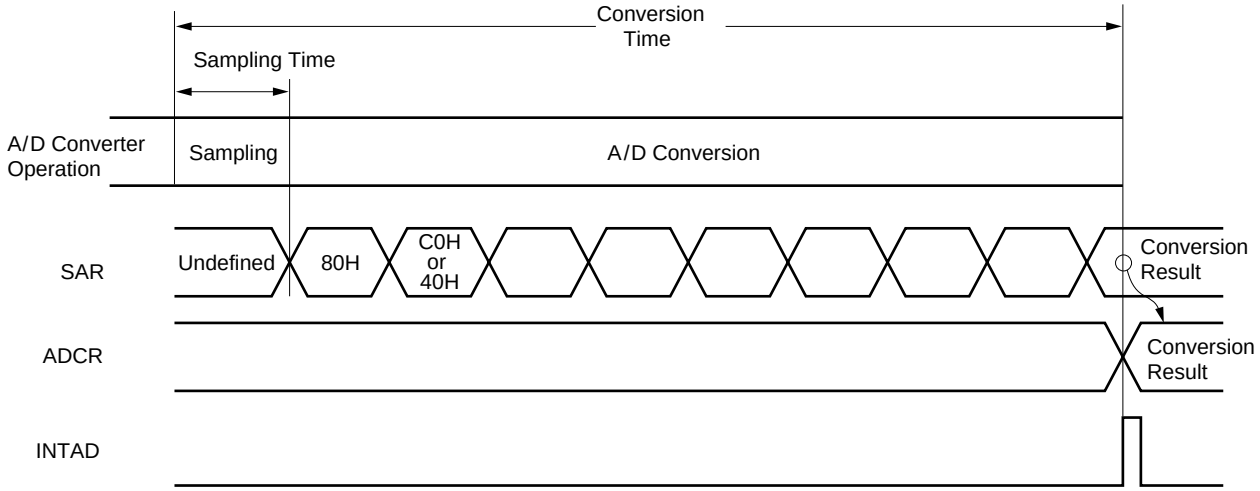
11.4 A/D Converter Operations

11.4.1 Basic operations of A/D converter

(1) Set the number of analog input channels with A/D converter input select register (ADIS) and turn A/D converter power supply on.
(2) From among the analog input channels set with ADIS, select one channel for A/D conversion with 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 the successive approximation register (SAR) is set and the tap selector sets the resistor string voltage tap to (1/2) V_{DD}.
(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 greater than (1/2) V_{DD}, the MSB of SAR remains set. If the input is smaller than (1/2) V_{DD}, 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 resistor string voltage tap is selected according to the preset value of bit 7 as described below.
    - Bit 7 = 1 : (3/4) V_{DD}
    - Bit 7 = 0 : (1/4) V_{DD}

    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 11-4. A/D Converter Basic Operation



A/D conversion operations are performed continuously until the bit 7 (CS) of the A/D converter mode register (ADM) is cleared (0) by software.

If a write to the ADM is performed during an A/D conversion operation, the conversion operation is initialized, and if the CS is set (1), conversion starts again from the beginning.

After reset input, the value of ADCR is undefined.

11.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (ANI0 to ANI5) and the A/D conversion result (the value stored in ADCR) is shown by the following expression.

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

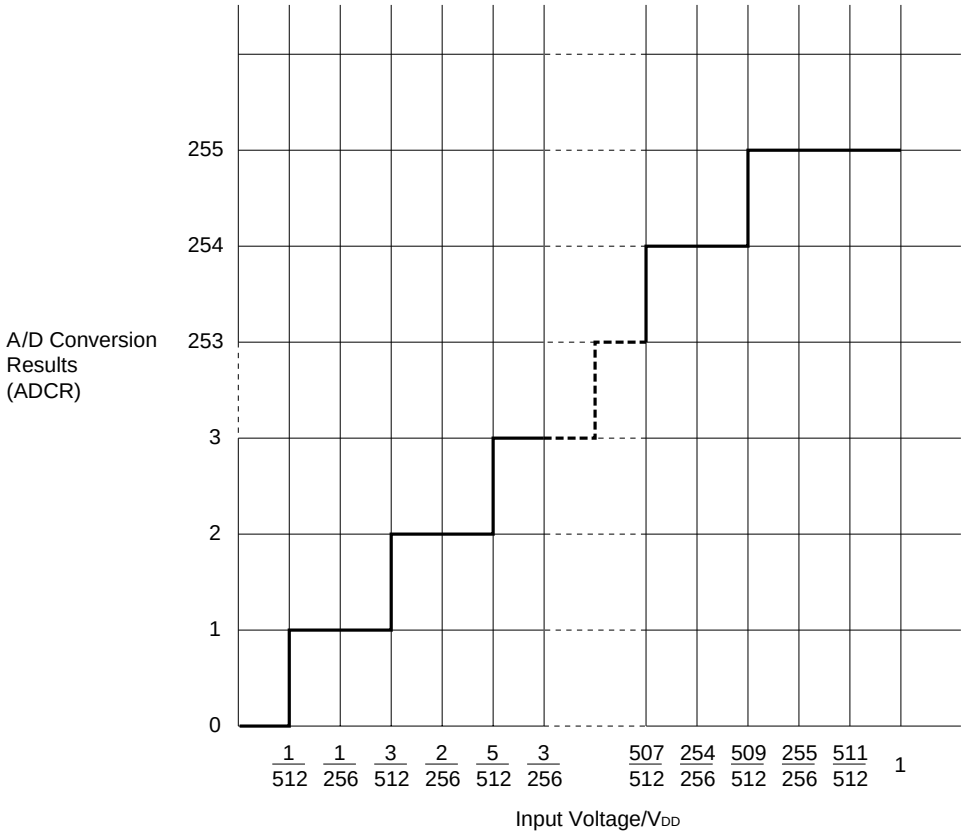
or

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

Where, INT( ) : Function which returns integer parts of value in parentheses.
V_{IN} : Analog input voltage
V_{DD} : Supply voltage
ADCR : ADCR register value

Figure 11-5 shows the relation between the analog input voltage and the A/D conversion result.

Figure 11-5. Relations between Analog Input Voltage and A/D Conversion Result



11.4.3 A/D converter operating mode

Select one analog input channel from among ANI0 to ANI5 with the A/D converter input select register (ADIS) and A/D converter mode register (ADM) and start A/D conversion.

The following two ways are available to start A/D conversion.

- Hardware start: Conversion is started by trigger input (fall of 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

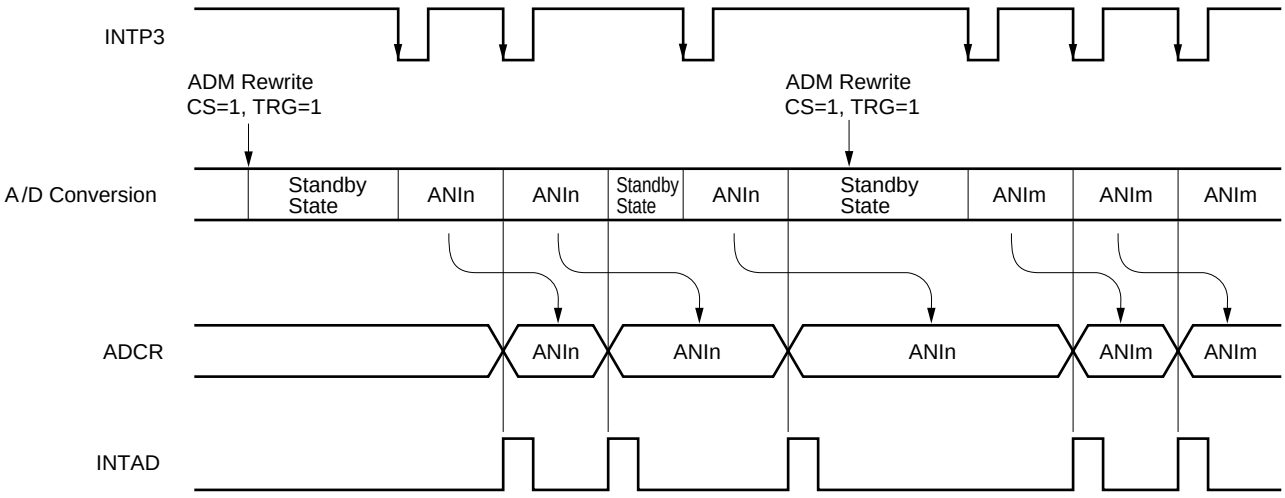
When bit 6 (TRG) and bit 7 (CS) of ADM are set to 1, the A/D conversion standby state is set. At the fall of the external trigger signal (INTP3), the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of A/D converter mode register (ADM).

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 11-6. A/D Conversion by Hardware Start



Remarks 1. n = 0, 1, ... , 5

2. m = 0, 1, ... , 5

(2) A/D conversion operation in software start

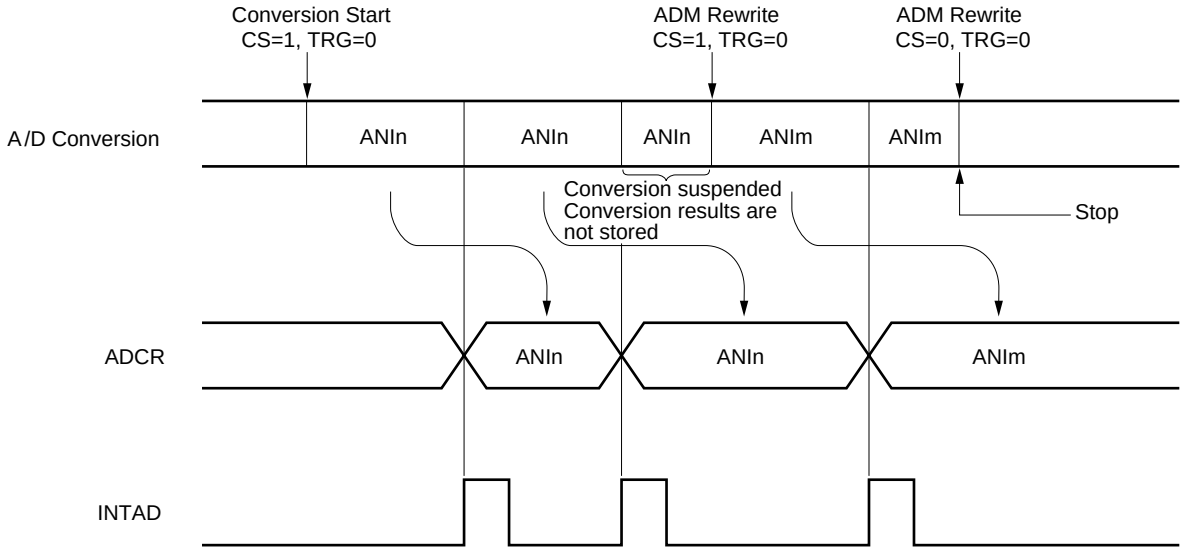
When bit 6 (TRG) and bit 7 (CS) of A/D converter mode register (ADM) are set to 0 and 1, respectively, the A/D conversion starts on the voltage applied to the analog input pins specified with bits 1 to 3 (ADM1 to ADM3) of ADM.

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 11-7. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ... , 5

2. m = 0, 1, ... , 5

11.5 A/D Converter Cautions

(1) Current consumption in standby mode

The A/D converter operates on the system clock. Therefore, its operation stops in STOP mode, but a current still flows in the resistor string.

To reduce the current consumption, clear bit 3 (ADON) of the A/D converter input select register (ADIS) to 0 before executing the HALT or STOP instruction. To reduce the power consumption, clear bit 7 (CS) of the A/D converter mode register (ADM) to 0 and stop A/D conversion before executing the HALT or STOP instruction.

(2) Input range of ANI0 to ANI5

The input voltages of ANI0 to ANI5 should be within the specification range. In particular, if a voltage above V_{DD} or below GND 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) Pins ANI0/P10 to ANI5/P15

The analog input pins ANI0 to ANI5 also function as input/output port (PORT1) pins. When A/D conversion is performed with any of pins ANI0 to ANI5 selected, be sure not to execute a PORT1 input instruction while conversion is in progress, as this may reduce the conversion resolution.

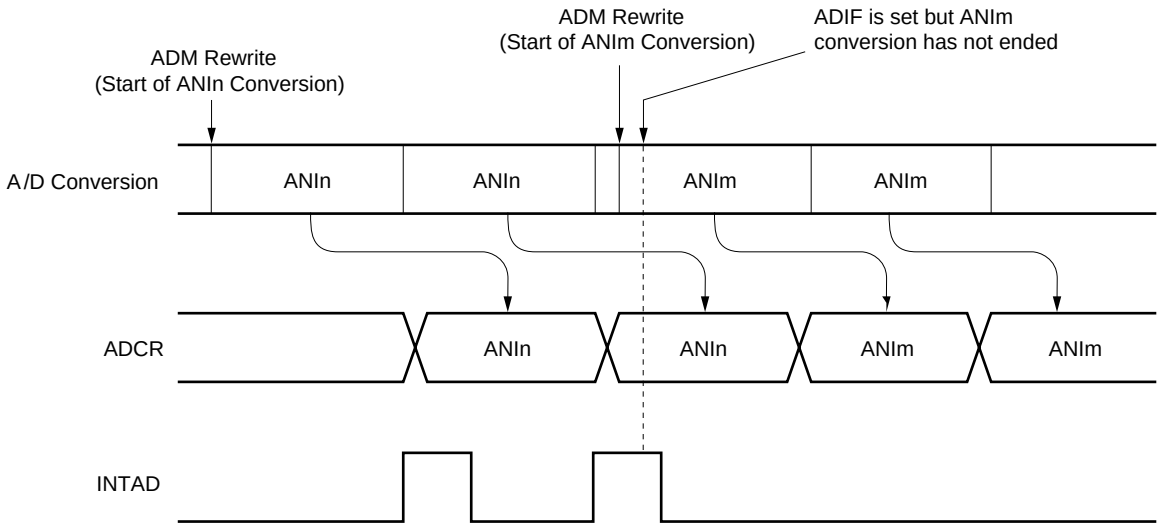
Also, if digital pulses are applied to a pin adjacent to the pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

(4) 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 A/D conversion result and ADIF 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.

When the A/D conversion is stopped and then resumed, clear the interrupt request flag (ADIF) before it is resumed.

Figure 11-8. A/D Conversion End Interrupt Request Generation Timing



CHAPTER 12 SERIAL INTERFACE CHANNEL 0

The μPD178018A subseries incorporates two channels of serial interfaces. Differences between channels 0 and 1 are as follows (Refer to **CHAPTER 13 SERIAL INTERFACE CHANNEL 1** for details of the serial interface channel 1.)

Table 12-1. Differences between Channels 0 and 1

Serial Transfer Mode		Channel 0	Channel 1
3-wire serial I/O	Clock selection	$f_{XX}/2$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$	$f_{XX}/2$, $f_{XX}/2^2$, $f_{XX}/2^3$, $f_{XX}/2^4$, $f_{XX}/2^5$, $f_{XX}/2^6$, $f_{XX}/2^7$, $f_{XX}/2^8$
	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			
I^2C bus (Inter IC Bus) **Note**			

Note When using the I^2C bus mode (including when this mode is implemented by software without using the internal hardware), consult NEC when you place an order for mask.

12.1 Serial Interface Channel 0 Functions

Serial interface channel 0 employs the following five modes.

- Operation stop mode
- 3-wire serial I/O mode
- SBI (serial bus interface) mode
- 2-wire serial I/O mode
- I^2C (Inter IC) bus mode**Note**

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

(1) Operation stop mode

This mode is used when serial transfer is not carried out. Power consumption can be reduced.

(2) 3-wire serial I/O mode (MSB-/LSB-first selectable)

This mode is used for 8-bit data transfer using three lines, one each for serial clock ($\overline{SCK0}$), serial output (SO0) and serial input (SI0). This mode enables simultaneous transmission/reception and therefore reduces the data transfer processing time.

The start bit of transferred 8-bit data is switchable between MSB and LSB, so that devices can be connected regardless of their start bit recognition.

This mode should be used when connecting with peripheral I/O devices or display controllers which incorporate a conventional synchronous clocked serial interface as is the case with the 75X/XL, 78K, and 17K series.

(3) SBI (serial bus interface) mode (MSB-first)

This mode is used for 8-bit data transfer with two or more devices using two lines of serial clock ($\overline{SCK0}$) and serial data bus (SB0 or SB1).

In the SBI mode, transfer data is classified into “addresses”, “commands” and “data” for transmission/reception, in conformance with NEC’s serial bus format.

- Address : Data to select a target device for serial communication
- Command : Data to give instructions to the target device
- Data : Data actually transferred

In fact, transfer is started when the master device outputs an “address” to the serial bus to select one of the slave devices subject to communication. After that, “commands” and “data” are transmitted between the master device and slave device to implement serial communication. The receiver can automatically identify the received data as “addresses”, “commands”, or “data” in hardware.

This function enables the input/output ports to be used effectively and the application program serial interface control portions to be simplified.

In this mode, the wake-up function for handshake and the output function of acknowledge and busy signals can also be used.

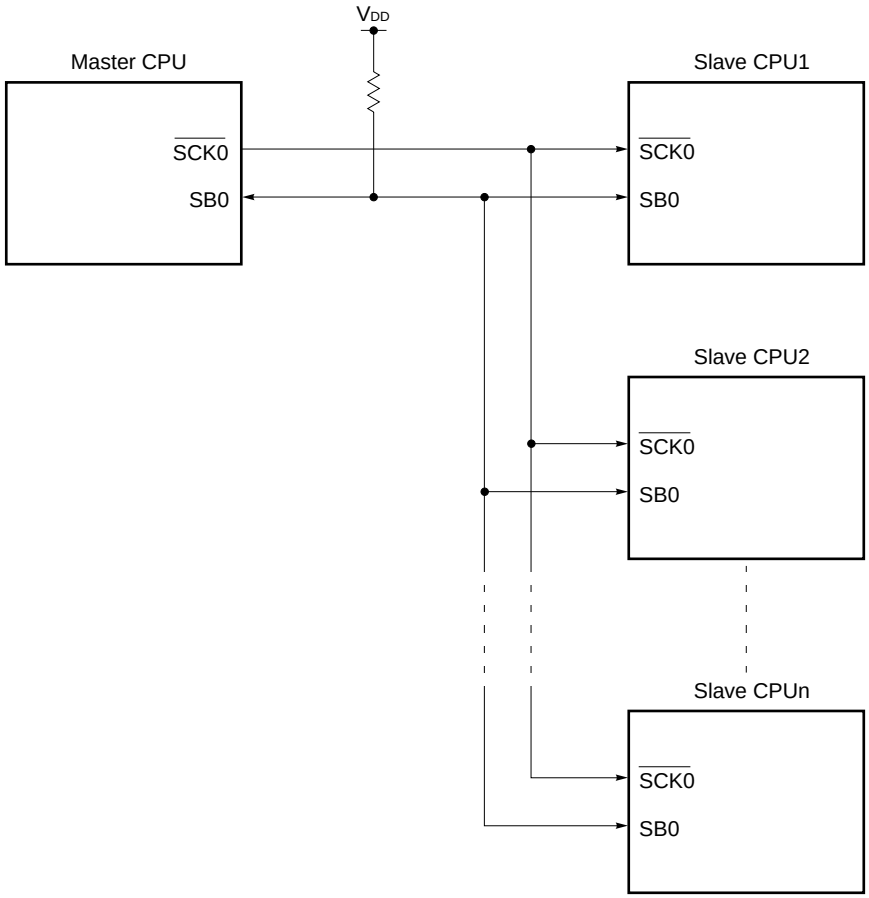
Note When using the I^2C bus mode (including when this mode is implemented by software without using the internal hardware), consult NEC when you place an order for mask.

(4) 2-wire serial I/O mode (MSB-first)

This mode is used for 8-bit data transfer using two lines of serial clock ($\overline{SCK0}$) and serial data bus (SB0 or SB1).

This mode enables to cope with any one of the possible data transfer formats by controlling the $\overline{SCK0}$ level and the SB0 or SB1 output level. Thus, the handshake line previously necessary for connection of two or more devices can be removed, resulting in the increased number of available input/output ports.

Figure 12-1. Serial Bus Interface (SBI) System Configuration Example

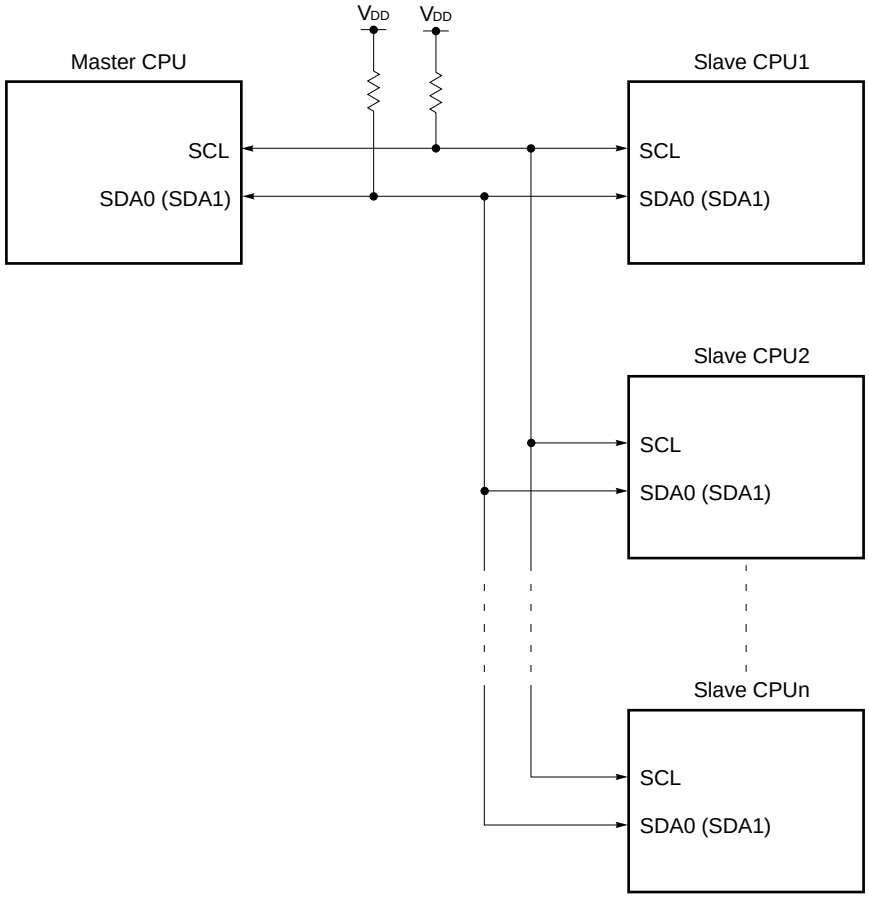


(5) I^2C bus mode (MSB-first)

This mode is used for 8-bit data transfer with two or more devices using two lines of serial clock (SCL) and serial data bus (SDA0 or SDA1).

This mode is in compliance with the I^2C bus format. In this mode, the transmitter outputs three kinds of data onto the serial data bus: “start condition”, “data”, and “stop condition”, to be actually sent or received. The receiver automatically distinguishes the received data into “start condition”, “data”, or “stop condition”, by hardware.

Figure 12-2. Serial Bus Configuration Example Using I^2C Bus



12.2 Serial Interface Channel 0 Configuration

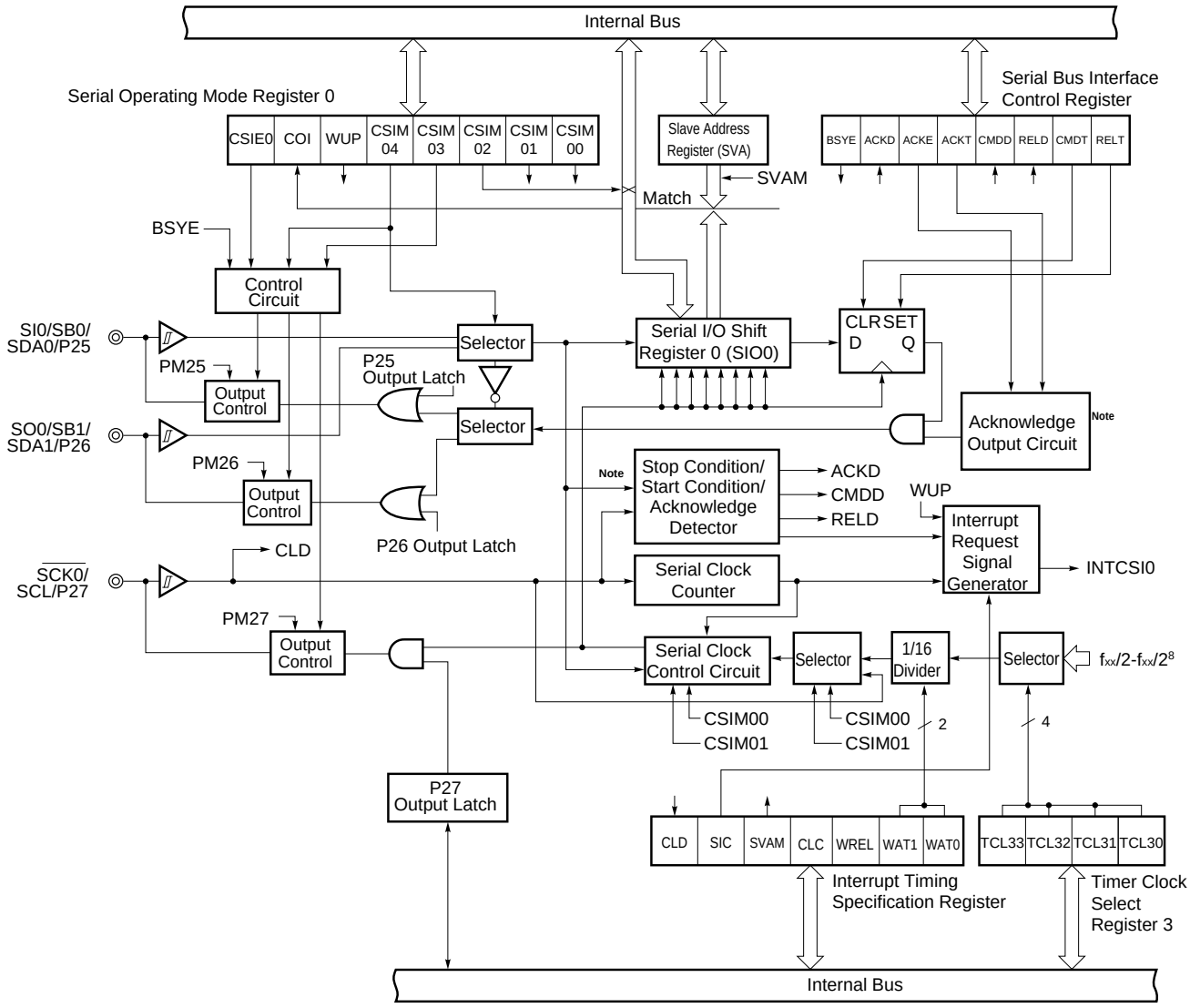
Serial interface channel 0 consists of the following hardware.

Table 12-2. 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 specification register (SINT) Port mode register 2 (PM2)**Note**

Note Refer to **Figure 4-5 P20, P21, P23 to P26 Block Diagram, Figure 4-6 P22 and P27 Block Diagram**.

Figure 12-3. Serial Interface Channel 0 Block Diagram



Note Example in I^2C bus mode operation.

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.

Note that, if a bus is driven in the SBI mode, 2-wire serial I/O mode or I^2C bus mode, the bus pin must 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 I^2C bus mode, set bit 7 (BSYE) of the serial bus interface control register 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.

Caution In the I^2C bus mode, do not execute an instruction that writes to SIO0 while WUP (bit 5 of serial operating mode register 0 (CSIM0)) = 1. Data can be received when the wake-up function is being used (WUP = 1), even if such an instruction is not executed. For the details of the wake-up function, refer to 12.4.5 (1) (c) Wake-up function.

(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. It 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.

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

If no matching is detected in address reception, bit 2 (RELD) of the serial bus interface control register (SBIC) is cleared to 0. By setting bit 5 (WUP) of CSIM0 to 1 in the SBI mode, the wake-up function can be used. In this case, an interrupt request signal (INTCSI0) is generated when the slave address output by the master matches with the value of SVA (the interrupt request is also generated when a stop condition is detected). This interrupt request indicates that the master has requested communication. Set SIC to 1 when using the wake-up function.

Further, when SVA transmits data as master or slave device in the SBI or 2-wire serial I/O mode, errors can be detected by using SVA.

Reset input makes SVA undefined.

(3) SO0 latch

This latch holds SI0/SB0/SDA0/P25 and SO0/SB1/SDA1/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 $\overline{\text{SCK0}}$/SCL/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 wake-up function specification bit. It is bit 5 of serial operating mode register 0 (CSIM0).

- In the I^2C bus mode
  Generates an interrupt request as shown in Table 12-3.

(7) Output circuit and detector of the control signals

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.

- In the SBI mode
  Busy/acknowledge output circuit, bus release/command/acknowledge detector

- In the I^2C bus mode
  Acknowledge output circuit, stop condition/start condition/acknowledge detector

Table 12-3. Serial Interface Channel 0 Interrupt Request Signal Generation

Serial Transfer mode	BSYE	WUP	WAT1	WAT0	ACKE	Description
I^2C bus mode (transmit)	0	0	1	0	0	An interrupt request signal is generated each time 8 serial clocks are counted (8-clock wait). Normally, during transmission the settings WAT21, WAT0=1, 0, are not used. They are used only when wanting to coordinate receive time and processing systematically using software. ACK information is generated by the receiving side, thus ACKE should be set to 0 (disable).
			1	1	0	An interrupt request signal is generated each time 9 serial clocks are counted (9-clock wait). ACK information is generated by the receiving side, thus ACKE should be set to 0 (disable).
	Other than above					Setting prohibited
I^2C bus mode (receive)	1	0	1	0	0	An interrupt request signal is generated each time 8 serial clocks are counted (8-clock wait). ACK information is output by manipulating ACKT by software after an interrupt is generated.
			1	1	0/1	An interrupt request signal is generated each time 9 serial clocks are counted (9-clock wait). To automatically generate ACK information, preset ACKE to 1 before transfer start. However, in the case of the master, set ACKE to 0 (disable) before receiving the last data.
	1	1	1	1	1	After address is received, if the values of the serial I/O shift register 0 (SI00) and the slave address register (SVA) match, an interrupt request signal and a stop condition are generated. To automatically generate ACK information, preset ACKE to 1 (enable) before transfer start.
	Other than above					Setting prohibited

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

12.3 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)

This register sets the serial clock of serial interface channel 0.

TCL3 is set with an 8-bit memory manipulation instruction.

Reset input sets TCL3 to 88H.

Figure 12-4. 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

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.13 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.13 MHz)	$f_X/2^3$ (563 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (563 kHz)	$f_X/2^4$ (281 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (281 kHz)	$f_X/2^5$ (141 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (141 kHz)	$f_X/2^6$ (70.3 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (70.3 kHz)	$f_X/2^7$ (35.2 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (35.2 kHz)	$f_X/2^8$ (17.6 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (17.6 kHz)	$f_X/2^9$ (8.8 kHz)
Other than above				Setting prohibited		

TCL33	TCL32	TCL31	TCL30	Serial Interface Channel 0 Serial Clock Selection		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.13 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.13 MHz)	$f_X/2^3$ (563 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (563 kHz)	$f_X/2^4$ (281 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (281 kHz)	$f_X/2^5$ (141 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (141 kHz)	$f_X/2^6$ (70.3 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (70.3 kHz)	$f_X/2^7$ (35.2 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (35.2 kHz)	$f_X/2^8$ (17.6 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (17.6 kHz)	$f_X/2^9$ (8.8 kHz)
Other than above				Setting prohibited		

Caution When rewriting TCL3 to other data, stop the serial transfer operation beforehand.

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. MCS : Oscillation mode selection register (OSMS) bit 0

4. Figures in parentheses apply to operation with f_X = 4.5 MHz.

(2) Serial operating mode register 0 (CSIM0)

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.

Reset input sets CSIM0 to 01H.

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

Figure 12-5. 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	CSIM04	CSIM03	CSIM02	CSIM01	1	FF60H	01H	R/W Note 1

R/W

CSIM01	Serial Interface Channel 0 Clock Selection
0	Input Clock to $\overline{\text{SCK0}}$/SCL/P27 pin from off-chip
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	Operation Mode	Start Bit	SIO/SB0/SDA0/P25 Pin Function	SO0/SB1/SDA1/P26 Pin Function	$\overline{\text{SCK0}}$/SCL/P27 Pin Function
0	×	0	Note 2 1	Note 2 ×	0	0	0	1	3-wire serial I/O mode	MSB	SI0 Note 2 (Input)	SO0 (CMOS output)	$\overline{\text{SCK0}}$ (CMOS I/O)
		1								LSB			
1	0	0	Note 3 ×	Note 3 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{\text{SCK0}}$ (CMOS I/O)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	
1	1	0	Note 3 ×	Note 3 ×	0	0	0	1	2-wire serial I/O mode or I^2C bus mode	MSB	P25 (CMOS I/O)	SB1/SDA1 (N-ch open-drain I/O)	$\overline{\text{SCK0}}$/SCL (N-ch open-drain I/O)
		1	0	0	Note 3 ×	Note 3 ×	0	1			SB0/SDA0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

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

2. Can be used as P25 (CMOS input/output) when used only for transmission.

3. Can be used freely as port function.

Remark × : don't care

PM×× : port mode register

P×× : output latch of port

Figure 12-5. Serial Operating Mode Register 0 Format (2/2)

R/W	WUP	Wake-up Function Control Note 1
	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 data in SBI mode • Interrupt request signal generation when the address received after start condition detected (CMDD = 1) matches the slave address register data in I^2C bus mode

R	COI	Slave Address Comparison Result Flag Note 2
	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. When using wake-up function (WUP = 1), set bit 5 (SIC) of the interrupt timing specification register (SINT) to 1. Do not execute write instruction to the serial I/O shift register 0 (SIO0) during WUP = 1.

2. When CSIE0 = 0, COI becomes 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.

Reset input sets SBIC to 00H.

Figure 12-6. Serial Bus Interface Control Register Format (1/3)

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
SBIC	BSYE	ACKD	ACKE	ACKT	CMDD	RELD	CMDT	RELT	FF61H	00H	R/W[Note]

R/W

RELT	Used for bus release signal output in SBI mode and stop condition 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	Used for command signal output in SBI mode and start condition 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 (SBI Mode)/Stop Condition Detection (I^2C Bus Mode)	
	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) or stop condition is detected

R

CMDD	Command Detection (SBI Mode)/Start Condition Detection (I^2C Bus Mode)	
	Clear Conditions (CMDD = 0)	Set Conditions (CMDD = 1)
	• When transfer start instruction is executed • When bus release signal (REL) or stop condition is detected • When CSIE0 = 0 • When reset input is applied	• When command signal (CMD) or stop condition is detected

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

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)

Figure 12-6. Serial Bus Interface Control Register Format (2/3)

(a) SBI mode

R/W

ACKT	Acknowledge signal is output in synchronization with the falling edge clock of $\overline{SCK0}$ 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.

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 $\overline{SCK0}$ (automatically output when ACKE = 1).
	After completion of transfer	Acknowledge signal is output in synchronization with the falling edge of $\overline{SCK0}$ 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)
• Falling edge of the $\overline{SCK0}$ immediately after the busy mode is released while executing the transfer start instruction • When CSIE0 = 0 • When reset input is applied		• When acknowledge signal ($\overline{ACK}$) is detected at the rising edge of $\overline{SCK0}$ clock after completion of transfer

R/W

Note BSYE	Synchronizing Busy Signal Output Control
0	Disables busy signal which is output in synchronization with the falling edge of $\overline{SCK0}$ clock just after execution of the instruction to be cleared to 0.
1	Outputs busy signal at the falling edge of $\overline{SCK0}$ clock following the acknowledge signal.

Note The busy mode can be canceled 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)

Figure 12-6. Serial Bus Interface Control Register Format (3/3)

(b) I^2C bus mode

R/W

ACKT	SDA0 (SDA1) is set to low after the set instruction (1) execution (ACKT = 1) before the next SCL falling edge. Used for generating an $\overline{ACK}$ signal by software if the 8-clock wait mode is selected. Cleared to 0 if CSIE = 0 when a transfer by the serial interface is started.

R/W

ACKE	Acknowledge Signal Output Control Note 1
0	Disables acknowledge signal automatic output. (However, output with ACKT is enabled) Used for reception when 8-clock wait mode is selected or for transmission. Note 2
1	Enables acknowledge signal automatic output. Outputs acknowledge signal in synchronization with the falling edge of the 9th SCL clock cycle (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output. Used in reception with 9-clock wait mode selected.

R

ACKD	Acknowledge Detection	
Clear Conditions (ACKD = 0)		Set Conditions (ACKD = 1)
• While executing the transfer start instruction • When CSIE0 = 0 • When reset input is applied		• When acknowledge signal ($\overline{ACK}$) is detected at the rising edge of SCL clock after completion of transfer

R/W

Note 3 BSYE	Control of N-ch Open-Drain Output for Transmission in I^2C Bus Mode Note 4
0	Output enabled (transmission)
1	Output disabled (reception)

Notes 1. Setting should be performed before transfer.

2. If 8-clock wait mode is selected, the acknowledge signal at reception time must be output using ACKT.

3. The busy mode can be canceled by start of serial interface transfer or reception of address signal. However, the BSYE flag is not cleared to 0.

4. When using the wake-up function, be sure to set BSYE to 1.

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

(4) Interrupt timing specification register (SINT)

This register sets the bus release interrupt and address mask functions and displays the P27/$\overline{\text{SCK0}}$/SCL pin level status. SINT is set with a 1-bit or 8-bit memory manipulation instruction.

Reset input sets SINT to 00H.

Figure 12-7. Interrupt Timing Specification Register Format (1/2)

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

R/W

WAT1	WAT0	Wait and Interrupt Control
0	0	Generates interrupt service request at rising edge of 8th $\overline{\text{SCK0}}$ clock cycle. (keeping clock output in high impedance)
0	1	Setting prohibited
1	0	Used in I^2C bus mode. (8-clock wait) Generates interrupt service request at rising edge of 8th SCK0 clock cycle. (In the case of master device, makes SCL output low to enter wait state after 8 clock pulses are output. In the case of slave device, makes SCL output low to request wait state after 8 clock pulses are input.)
1	1	Used in I^2C bus mode. (9-clock wait) Generates interrupt service request at rising edge of 9th SCK0 clock cycle. (In the case of master device, makes SCL output low to enter wait state after 9 clock pulses are output. In the case of slave device, makes SCL output low to request wait state after 9 clock pulses are input.)

R/W

WREL	Wait Sate Cancellation Control
0	Wait state has been cancelled.
1	Cancels wait state. Automatically cleared to 0 when the state is cancelled. (Used to cancel wait state by means of WAT0 and WAT1.)

R/W

CLC	Clock Level Control[Note2]
0	Used in I^2C bus mode. Make output level of SCL pin low unless serial transfer is being performed.
1	Used in I^2C bus mode. Make SCL pin enter high-impedance state unless serial transfer is being performed. (except for clock line which is kept high) Used to enable master device to generate start condition and stop condition signals.

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

2. When not using the I^2C mode, set CLC to 0.

Figure 12-7. Interrupt Timing Specification Register Format (2/2)

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 Source Selection[Note1]
0	CSIIF0 is set to 1 upon termination of serial interface channel 0 transfer
1	CSIIF0 is set to 1 upon stop condition detection or termination of serial interface channel 0 transfer

R

CLD	P27/$\overline{SCK0}$/SCL Pin Level[Note2]
0	Low level
1	High level

Notes 1. When using wake-up function in the I^2C mode, set SIC to 0.
When using wake-up function in SBI mode, set SIC to 0.

2. When CSIE0 = 0, CLD becomes 0.

Remark SVA : Slave address register
CSIIF0: Interrupt request flag corresponding to INTCSI0
CSIE0 : Bit 7 of the serial operating mode register 0 (CSIM0)

12.4 Serial Interface Channel 0 Operations

The following five 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
- I^2C (Inter IC) bus mode

12.4.1 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 ordinary 8-bit register.

In the operation stop mode, the P25/SI0/SB0/SDA0, P26/SO0/SB1/SDA1 and P27/$\overline{SCK0}$/SCL 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.

Reset input sets CSIM0 to 01H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	1	FF60H	01H	R/W

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enabled

12.4.2 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 ($\overline{SCK0}$), 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.

Reset input sets CSIM0 to 01H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	1	FF60H	01H	R/W Note 1

R/W	CSIM01	Serial Interface Channel 0 Clock Selection
	0	Input Clock to $\overline{SCK0}$/SCL/P27 pin from off-chip
	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	Operation Mode	Start Bit	SIO/SB0/SDA0/P25 Pin Function	SO0/SB1/SDA1/P26 Pin Function	$\overline{SCK0}$/SCL/P27 Pin Function
	0	×	0	Note 2 1	Note 2 ×	0	0	0	1	3-wire serial I/O mode	MSB	SI0 Note 2 (Input)	SO0 (CMOS output)	$\overline{SCK0}$ (CMOS I/O)
			1								LSB			
	1	0	SBI mode (Refer to **12.4.3 SBI mode operation**.)											
	1	1	2-wire serial I/O mode (refer to **12.4.4 2-wire serial I/O mode operation**) or I^2C bus mode (refer to **12.4.5 I^2C bus mode operation**.)											

R/W	WUP	Wake-up Function Control Note 3
	0	Interrupt request signal generation with each serial transfer in any mode
	1	Setting prohibited

R/W	CSIE0	Serial Interface Channel 0 Operation Control
	0	Operation stopped
	1	Operation enabled

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

2. Can be used as P25 (CMOS input/output) when used only for transmission.

3. Be sure to set WUP to 0 when the 3-wire serial I/O mode is selected.

Remark × : don't care

PM××: port mode register

P×× : output latch of port

(b) Serial bus interface control register (SBIC)

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

Reset input sets 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.

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

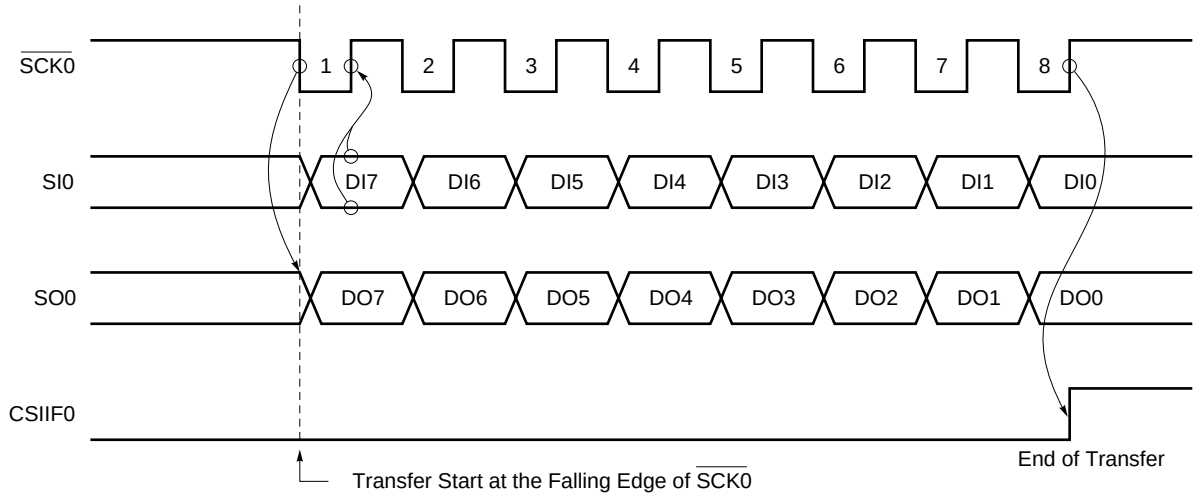
(2) Communication operation

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 with 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 ($\overline{SCK0}$). The transmitted data is held in the SO0 latch and is output from the SO0 pin. The received data input to the SI0 pin is latched in SIO0 at the rising edge of $\overline{SCK0}$.

Upon termination of 8-bit transfer, SIO0 operation stops automatically and the interrupt request flag (CSIIF0) is set.

Figure 12-8. 3-Wire Serial I/O Mode Timings



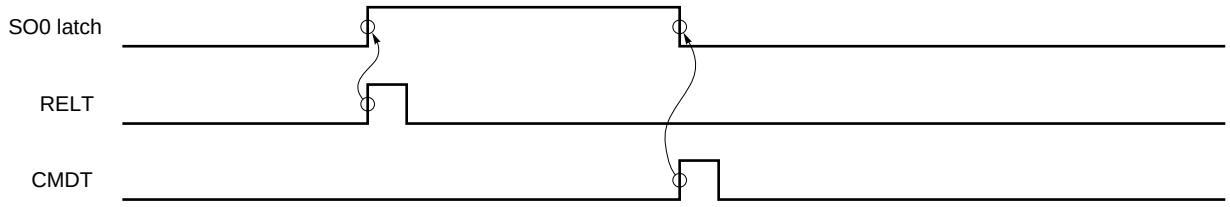
The SO0 pin is a CMOS output pin and outputs current SO0 latch statuses. Thus, the SO0 pin output status can be manipulated by setting the bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **12.4.8 $\overline{SCK0}$/SCL/P27 pin output manipulation**).

(3) Other signals

Figure 12-9 shows RELT and CMDT operations.

Figure 12-9. RELT and CMDT Operations



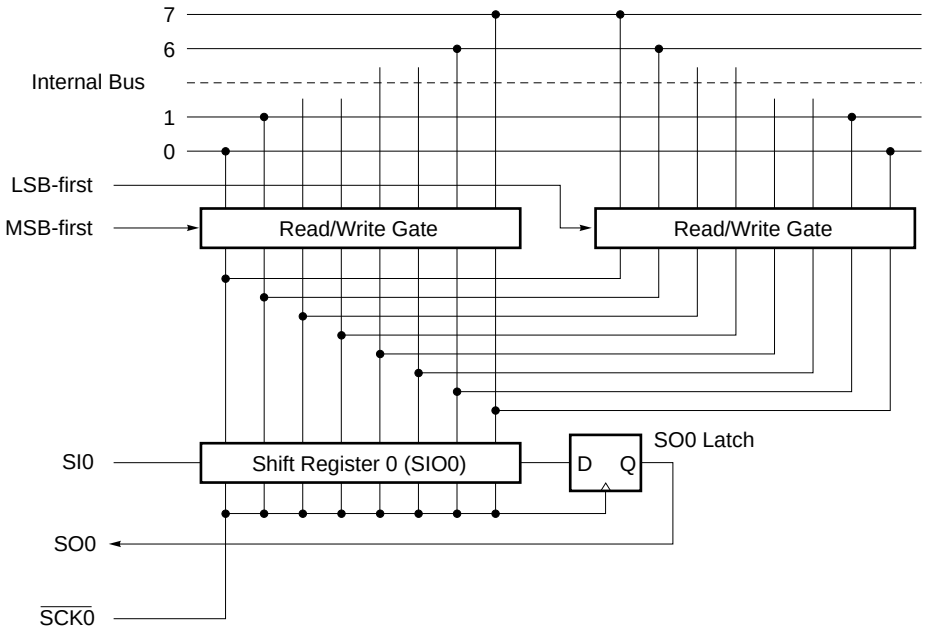
(4) MSB/LSB switching as the start bit

The 3-wire serial I/O mode enables to select transfer to start from MSB or LSB.

Figure 12-10 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 12-10. 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, switching between MSB-first and LSB-first must be performed 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.

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

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (CSIIF0) is set.

12.4.3 SBI mode operation

SBI (Serial Bus Interface) is a high-speed serial interface in compliance with the NEC serial bus format.

SBI uses a single master device and employs the clocked serial I/O format with the addition of a bus configuration function. This function enables devices to communicate using only two lines. 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 outputs three kinds of data to slave devices on the serial data bus: “addresses” to select a device to be communicated with, “commands” to instruct the selected device, and “data” which is actually required.

The slave device can identify the received data into “address”, “command”, or “data”, by hardware. This function enables the application program to control the serial interface channel 0 to be simplified.

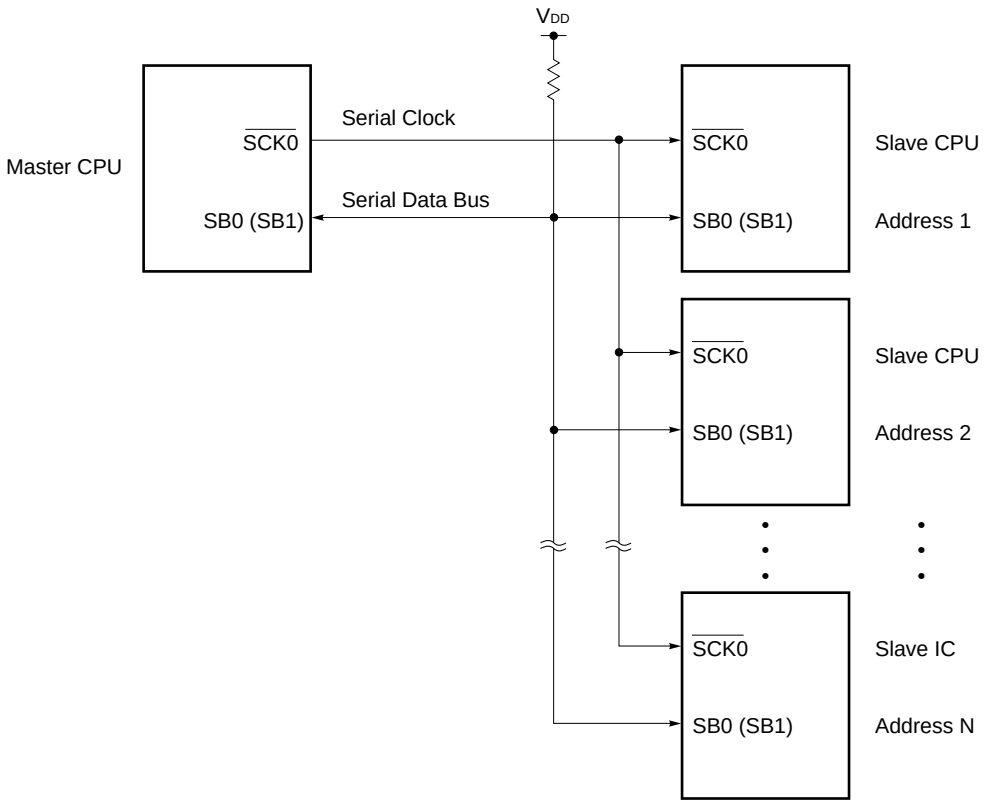
The SBI function is incorporated into various devices including 75X/XL series and 78K series devices.

Figure 12-11 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 is an open-drain output pin and therefore the serial data bus line behaves in the same way as the wired-OR configuration. In addition, a pull-up resistor must be connected to the serial data bus line.

When the SBI mode is used, refer to **(11) SBI mode precautions (d)** described later.

Figure 12-11. Example of Serial Bus Configuration with SBI



Caution When exchanging the master CPU/slave CPU, a pull-up resistor is necessary for the serial clock line ($\overline{\text{SCK0}}$) as well because serial clock line ($\overline{\text{SCK0}}$) input/output switching is carried out asynchronously between the master and slave CPUs.

(1) SBI functions

In the conventional serial I/O format, when a serial bus is configured by connecting two or more devices, many ports and wiring are necessary, to provide chip select signal to identify command and 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 configured with two signal lines of serial clock $\overline{\text{SCK0}}$ and serial data bus SB0 (SB1). Thus, use of SBI leads to reduction in the number of microcontroller 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.

(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 (INTCSI0) 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 underway serial communications.

(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

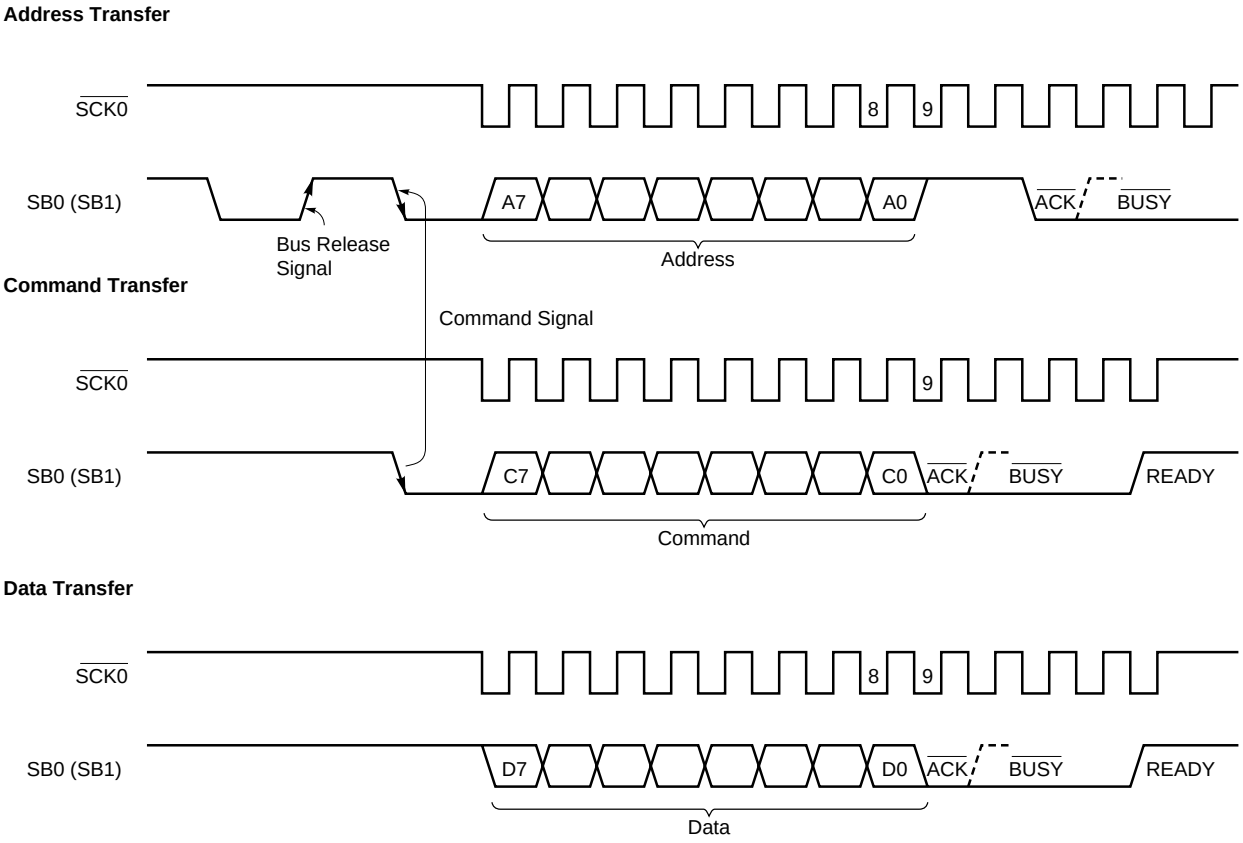
The busy signal to report the slave busy state is controlled.

(2) SBI definition

The SBI serial data format and the signals to be used are defined as follows.

Serial data to be transferred with SBI consists of three kinds of data: “address”, “command”, and “data”. Figure 12-12 shows the address, command, and data transfer timings.

Figure 12-12. SBI Transfer Timings



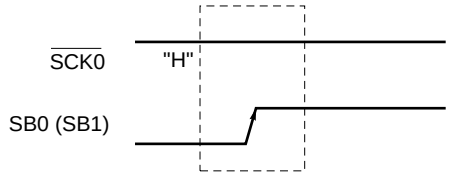
Remark The dotted line indicates the READY status.

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.

(a) Bus release signal (REL)

The bus release signal is a signal with the SB0 (SB1) line which has changed from the low level to the high level when the $\overline{SCK0}$ line is at the high level (without serial clock output).
This signal is output by the master device.

Figure 12-13. Bus Release Signal



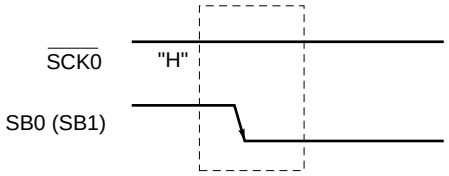
The bus release signal indicates that the master device is going to transmit an address to the slave device.
The slave device incorporates hardware to detect the bus release signal.

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

(b) Command signal (CMD)

The command signal is a signal with the SB0 (SB1) line which has changed from the high level to the low level when the $\overline{SCK0}$ line is at the high level (without serial clock output). This signal is output by the master device.

Figure 12-14. Command Signal



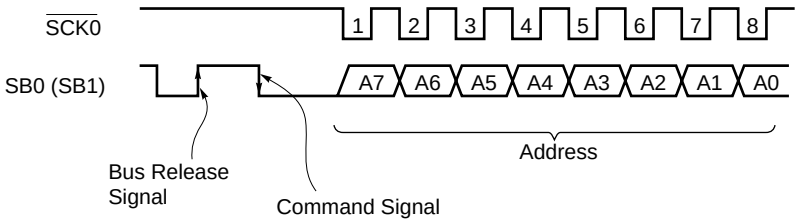
The command signal indicates that the master is going to transmit a command to the slave (however, the command signal following the bus release signal indicates that an address is to be transmitted).
The slave device incorporates hardware to detect the command signal.

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

(c) Address

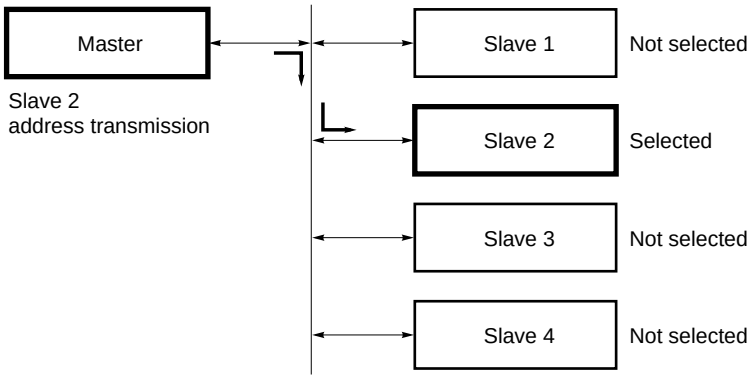
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Figure 12-15. Addresses



8-bit data following bus release and command signals is defined as an “address”. In the slave device, this condition is detected by hardware and whether or not 8-bit data matches the own specification number (slave address) is checked by hardware. If the 8-bit data matches the slave address, the slave device has been selected. After that, communication with the master device continues until a release instruction is received from the master device.

Figure 12-16. Slave Selection with Address



(d) Command and data

The master device transmits commands to, and transmits/receives data to/from the slave device selected by address transmission.

Figure 12-17. Commands

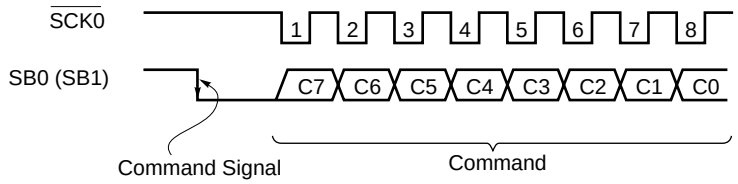
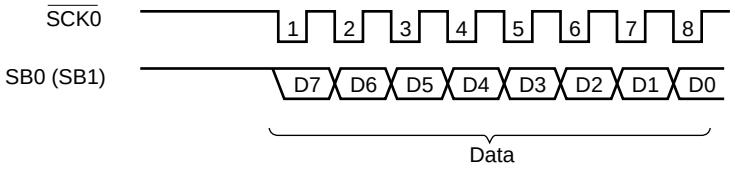


Figure 12-18. Data



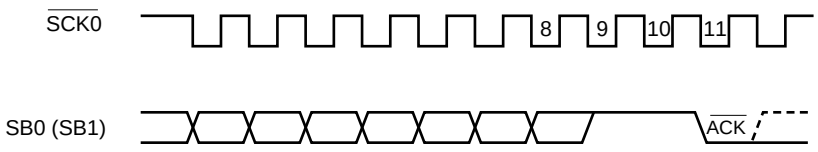
8-bit data following a command signal is defined as “command” data. 8-bit data without command signal is defined as “data”. Command and data operation procedures are allowed to determine by user according to communications specifications.

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

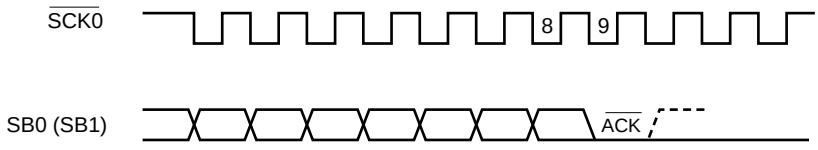
The acknowledge signal is used to check serial data reception between transmitter and receiver.

Figure 12-19. Acknowledge Signal

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



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



The acknowledge signal is one-shot pulse to be generated at the falling edge of $\overline{SCK0}$ after 8-bit data transfer. It can be positioned anywhere and can be synchronized with any clock $\overline{SCK0}$.

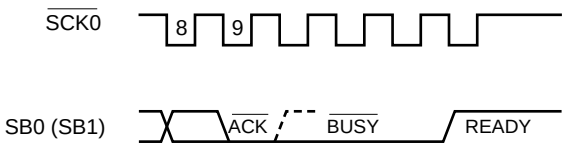
After 8-bit data transmission, the transmitter checks whether the receiver has returned the acknowledge signal. If the acknowledge signal is not returned for the preset period of time after data transmission, it can be judged that data reception has not been carried out correctly.

(f) Busy signal ($\overline{\text{BUSY}}$) and ready signal (READY)

The $\overline{\text{BUSY}}$ signal is intended to report to the master device that the slave device is preparing for data transmission/reception.

The READY signal is intended to report to the master device that the slave device is ready for data transmission/reception.

Figure 12-20. $\overline{\text{BUSY}}$ and READY Signals



In SBI, the slave device notifies the master device of the busy state by setting SB0 (SB1) line to the low level.

The $\overline{\text{BUSY}}$ signal output follows the acknowledge signal output from the master or slave device. It is set/reset at the falling edge of $\overline{\text{SCK0}}$. When the $\overline{\text{BUSY}}$ signal is reset, the master device automatically terminates the output of $\overline{\text{SCK0}}$ serial clock.

When the $\overline{\text{BUSY}}$ signal is reset and the READY signal is set, the master device can start the next transfer.

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

(3) Register setting

The SBI mode is set 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.

Reset input sets CSIM0 to 01H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	1	FF60H	01H	R/W Note 1

R/W

CSIM01	Serial Interface Channel 0 Clock Selection
0	Input Clock to $\overline{\text{SCK0}}$/SCL/P27 pin from off-chip
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	Operation Mode	Start Bit	SI0/SB0/SDA0/P25 Pin Function	SO0/SB1/SDA1/P26 Pin Function	$\overline{\text{SCK0}}$/SCL/P27 Pin Function
0	×	3-wire serial I/O mode (Refer to **12.4.2 3-wire serial I/O mode operation**.)											
1	0	0	Note 2 ×	Note 2 ×	0	0	0	1	SBI mode	MSB	P25 (CMOS I/O)	SB1 (N-ch open-drain I/O)	$\overline{\text{SCK0}}$ (CMOS I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0 (N-ch open-drain I/O)	P26 (CMOS I/O)	
1	1	2-wire serial I/O mode (refer to **12.4.4 2-wire serial I/O mode operation**) or I^2C bus mode (refer to **12.4.5 I^2C bus 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 data 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 enabled

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

2. Can be used as a port.

3. Clear bit 5 (SIC) of the interrupt timing specification register (SINT) to 0 when using the wake-up function (WUP=1).

4. When CSIE0=0, COI becomes 0.

Remark × : don't care

PM×× : port mode register

P×× : output latch of port

(b) Serial bus interface control register (SBIC)

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

Reset input sets 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	Used 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	Used 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 $\overline{\text{SCK0}}$ 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.

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 falling edge clock 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.

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. Bit 7 of the serial operating mode register 0 (CSIM0)

R

ACKD	Acknowledge Detection	
Clear Conditions (ACKD = 0)		Set Conditions (ACKD = 1)
• $\overline{SCK0}$ fall immediately after the busy mode is released during the transfer start instruction execution. • When CSIE0 = 0 • When reset input is applied		• When acknowledge signal ($\overline{ACK}$) is detected at the rising edge of $\overline{SCK0}$ clock after completion of transfer

R/W

Note BSYE	Synchronizing Busy Signal Output Control
0	Disables busy signal which is output in synchronization with the falling edge of $\overline{SCK0}$ clock just after execution of the instruction to be cleared to (0) (makes READY status).
1	Outputs busy signal at the falling edge of $\overline{SCK0}$ clock following the acknowledge signal.

Note Busy mode can be cleared by start of serial interface transfer. However, 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.

Reset input sets 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 Source 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	$\overline{\text{SCK0}}$/SCL/P27 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 12-21 to 12-26 show various signals and flag operations in SBI. Table 12-4 lists various signals in SBI.

Figure 12-21. RELT, CMDT, RELD, and CMDD Operations (Master)

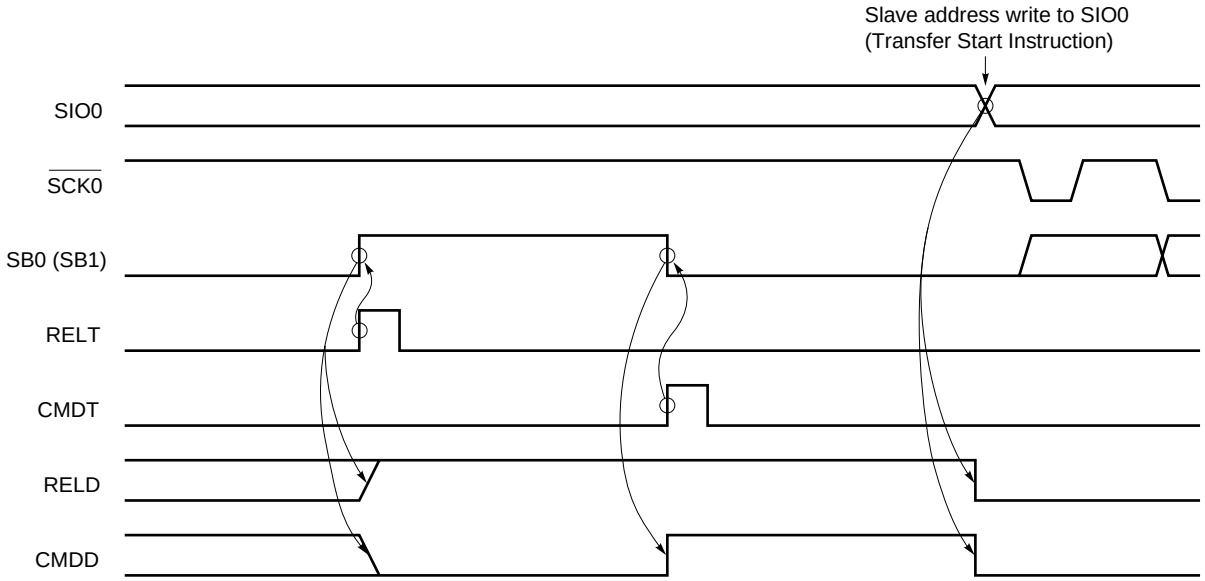


Figure 12-22. RELT and CMDD Operations (Slave)

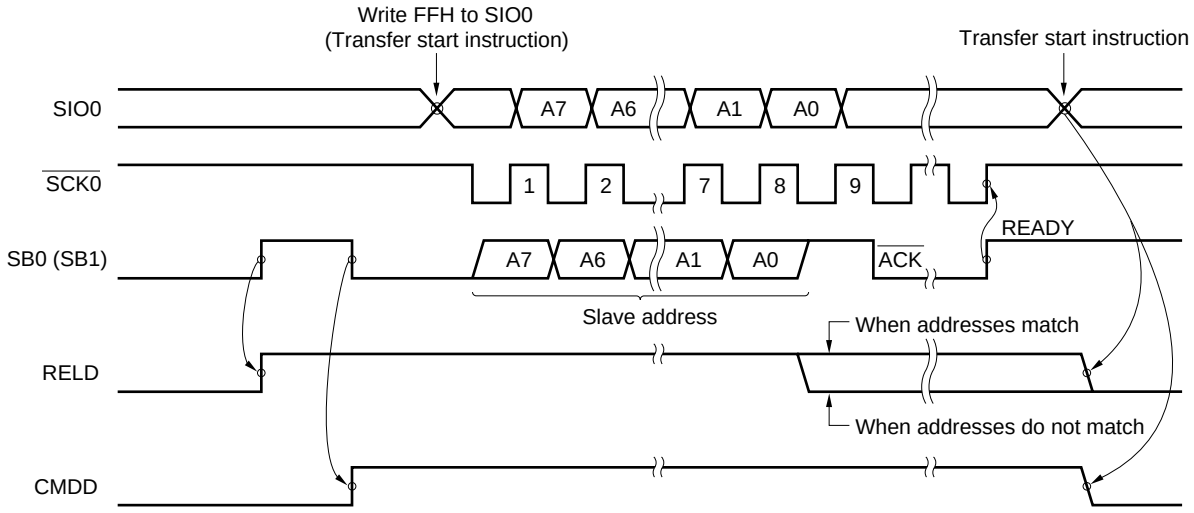
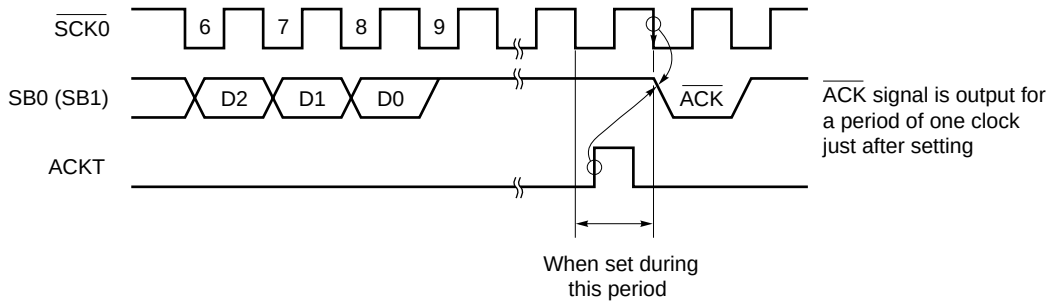


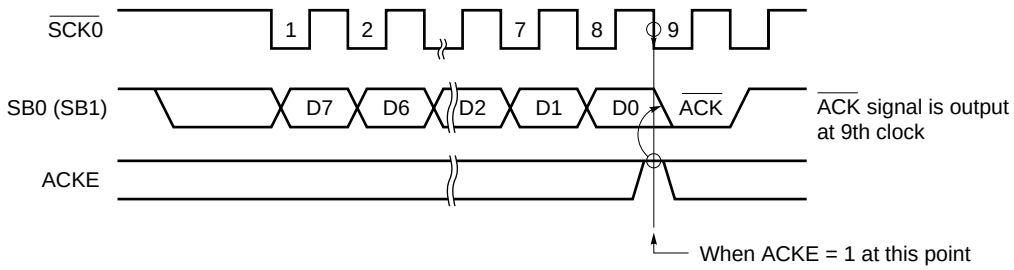
Figure 12-23. ACKT Operation



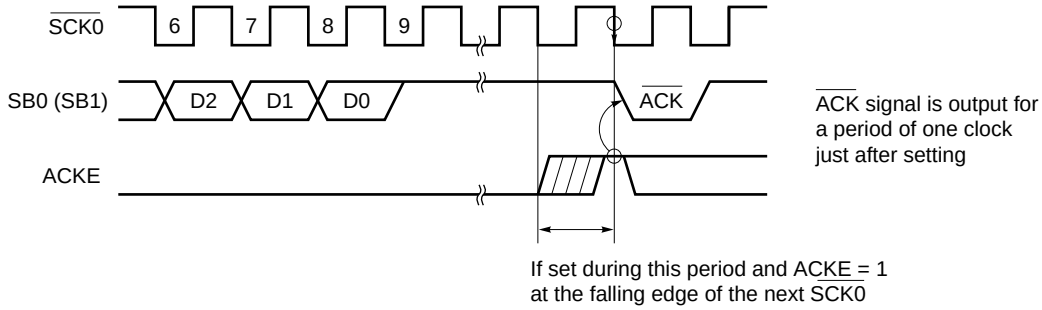
Caution Do not set ACKT before termination of transfer.

Figure 12-24. ACKE Operations

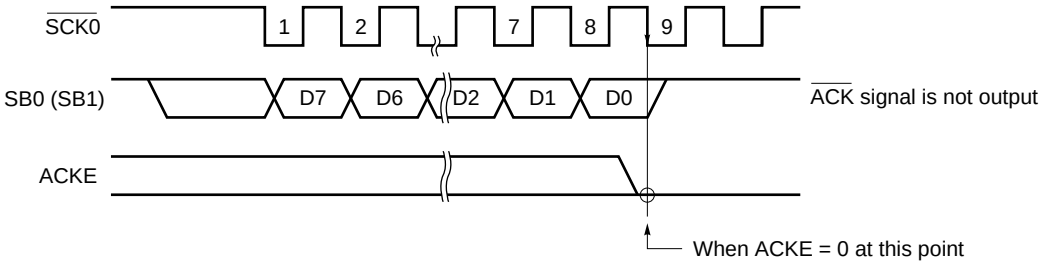
(a) When ACKE = 1 upon completion of transfer



(b) When set after completion of transfer



(c) When ACKE = 0 upon completion of transfer



(d) When "ACKE = 1" period is short

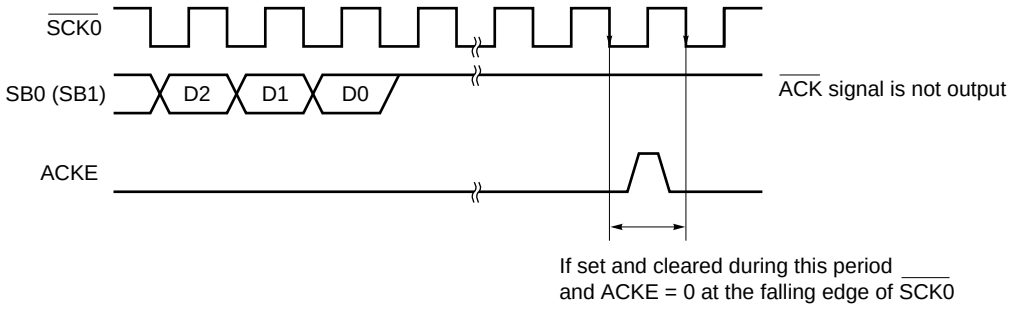
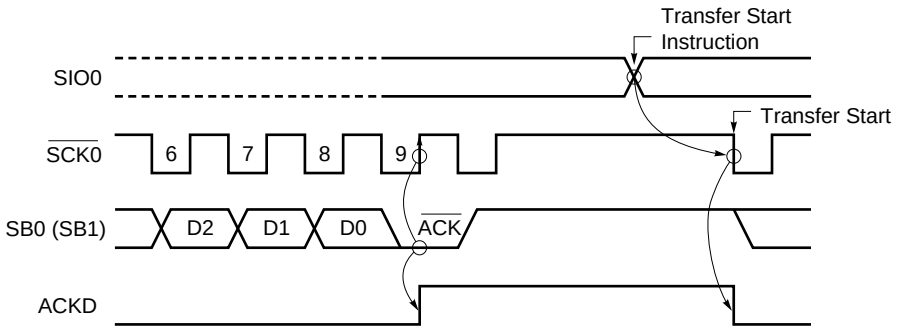
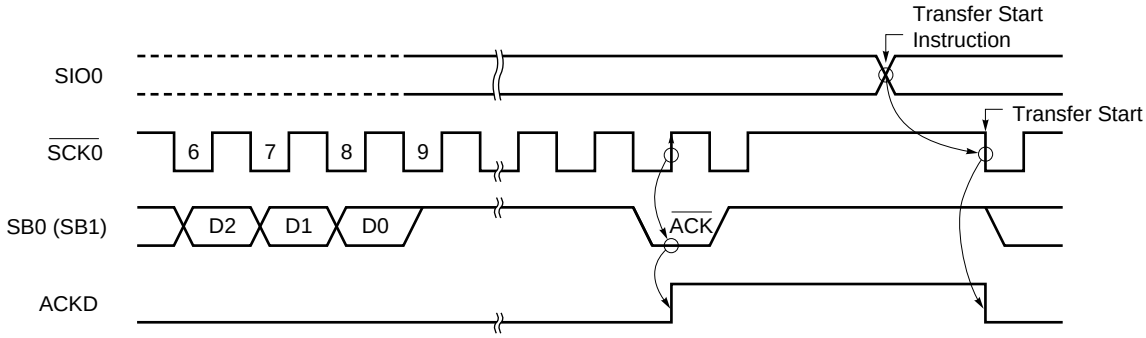


Figure 12-25. ACKD Operations

(a) When $\overline{ACK}$ signal is output at 9th clock of $\overline{SCK0}$



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



(c) Clear timing when transfer start is instructed in BUSY

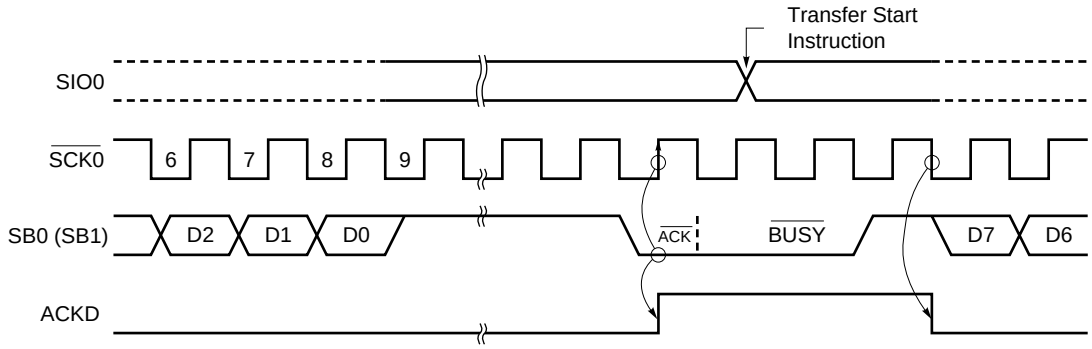


Figure 12-26. BSYE Operation

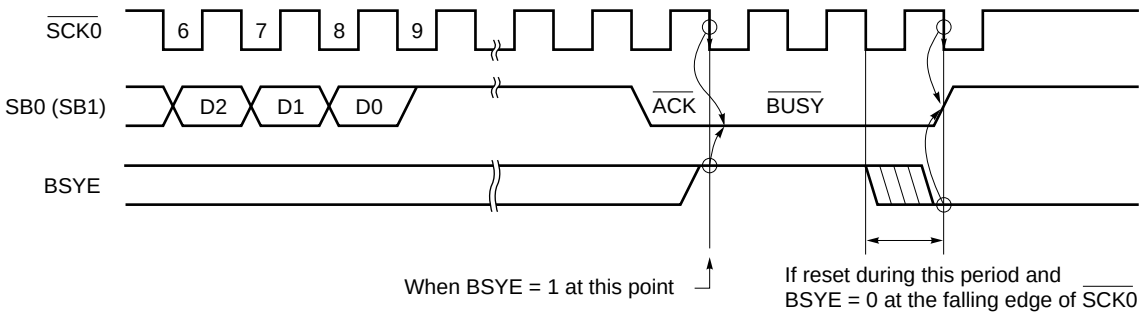


Table 12-4. Various Signals in SBI Mode (1/2)

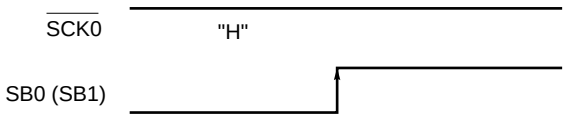
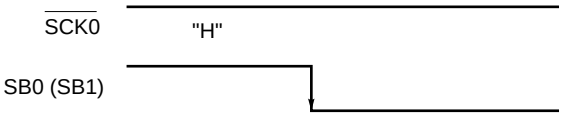
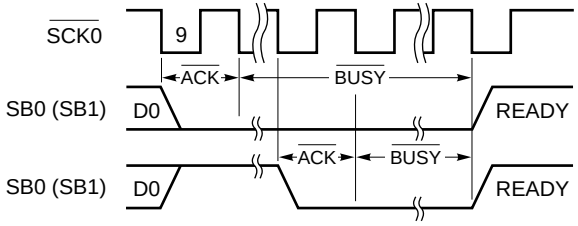
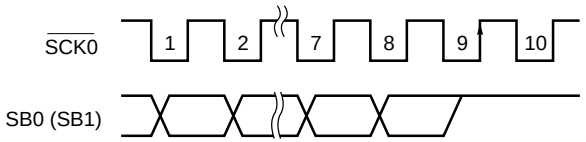
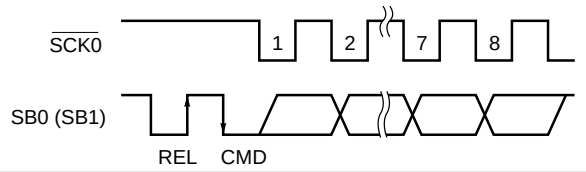
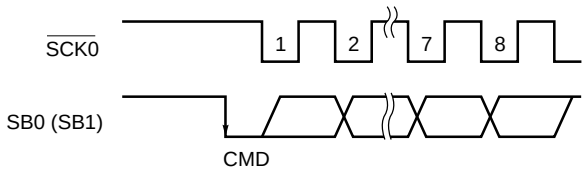
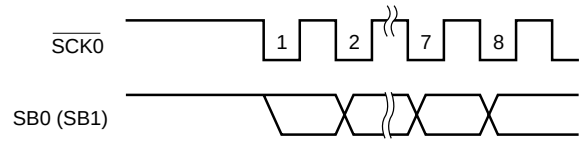
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0 (SB1) rising edge when $\overline{SCK0}$ = 1		RELT set	• RELD set • CMDD clear	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0 (SB1) falling edge when $\overline{SCK0}$ = 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 an command.
Acknowledge signal ($\overline{ACK}$)	Master/ slave	Low-level signal to be output to SB0 (SB1) during one-clock period of $\overline{SCK0}$ after completion of serial reception	[Synchronous BUSY output]	①ACKE = 1 ②ACKT set	ACKD set	Completion of reception
Busy signal ($\overline{BUSY}$)	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		①BSYE = 0 ②Execution of instruction for data write to SIO0 (transfer start instruction)	—	Serial receive enable

Table 12-4. Various Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Serial clock ($\overline{SCK0}$)	Master	Synchronous clock to output address/command/ data, $\overline{ACK}$ signal, synchronous $\overline{BUSY}$ 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 $\overline{SCK0}$)Note 1	Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK0}$ after output of REL and CMD signals				Address value of slave device on the serial bus
Commands (C7 to C0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK0}$ 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 $\overline{SCK0}$ without output of REL and CMD signals				Numeric values to be processed with slave or master device

Notes 1. When WUP = 0, CSIIF0 is set at the rising edge of the 9th clock of $\overline{SCK0}$.
When WUP = 1, an address is received. Only when the address matches the slave address register (SVA) value, CSIIF0 is set.

2. In $\overline{BUSY}$ state, transfer starts after the READY state is set.

(5) Pin configuration

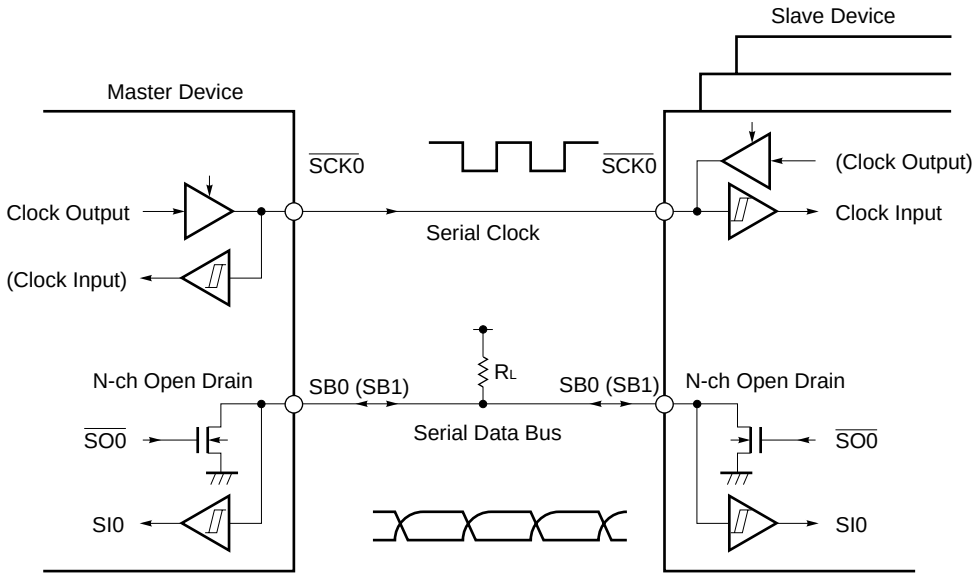
The serial clock pin ($\overline{SCK0}$) and serial data bus pin SB0 (SB1) have the following configurations.

(a) $\overline{SCK0}$ ..... 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 12-27. Pin Configuration



Caution Because the N-ch open-drain output must be made high-impedance state at time of data reception, write FFH to SIO0 in advance. The N-ch open-drain output can be made high-impedance state at any time of transfer. However, when the wake-up function specification bit (WUP) = 1, the N-ch open-drain output is always made high-impedance state. Thus, it is not necessary to write FFH to SIO0.

(6) Address match detection method

In the SBI mode, a particular slave device is selected by address communication from the master device and communication is started.

Address match detection is executed by hardware. With the slave address register (SVA), CSIIF0 is set in the wake-up state (WUP = 1) only when the address transmitted from the master device matches the value set to SVA.

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 (INTCSI0) 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 way.

(a) Method of comparing 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) Method of using 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 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 12-28 to 12-31 show data communication timing charts.

Shift operation of the serial I/O shift register 0 (SIO0) is carried out at the falling edge of 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 SIO0.

Figure 12-28. Address Transmission from Master Device to Slave Device (WUP = 1)

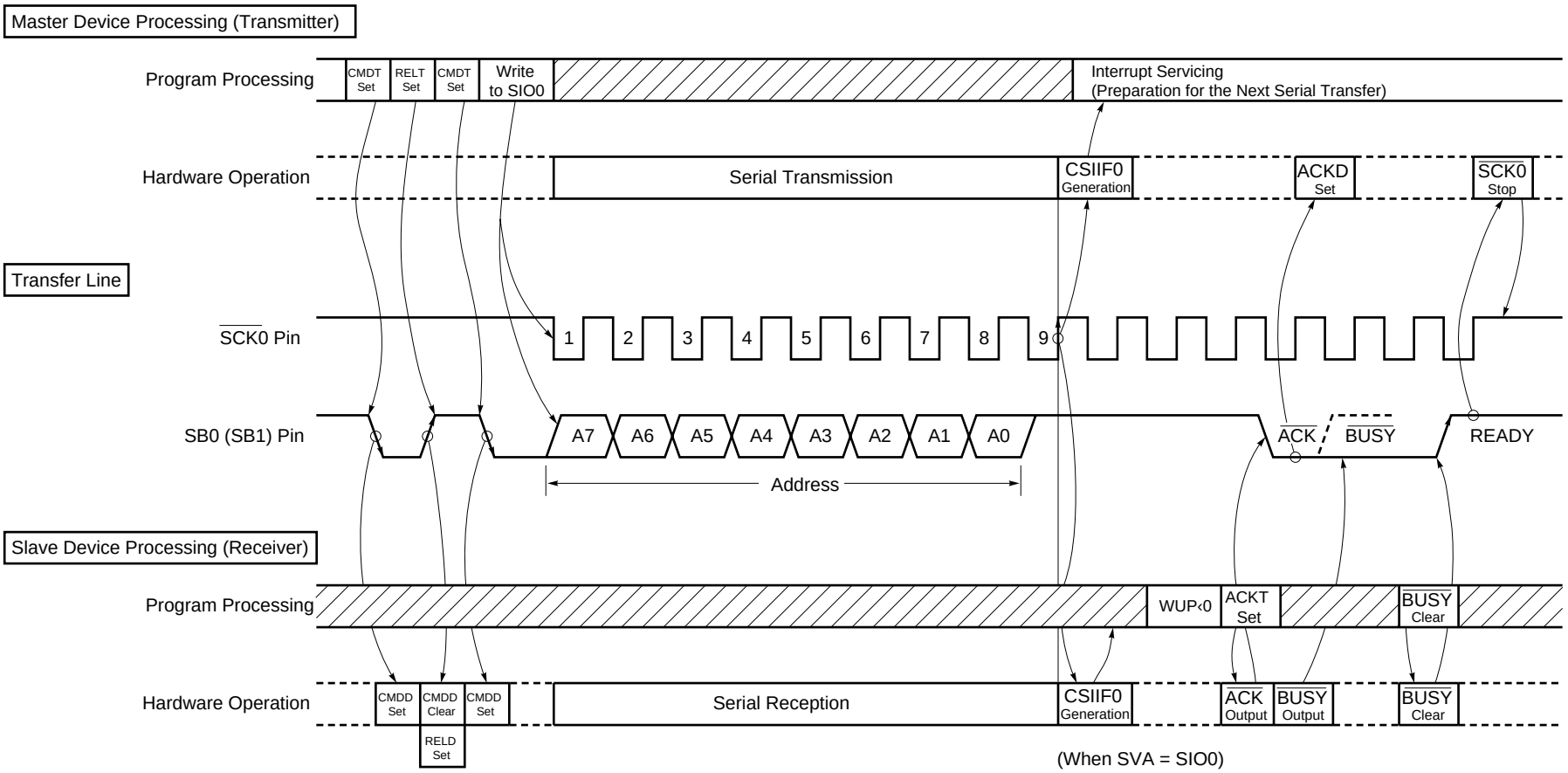


Figure 12-29. Command Transmission from Master Device to Slave Device

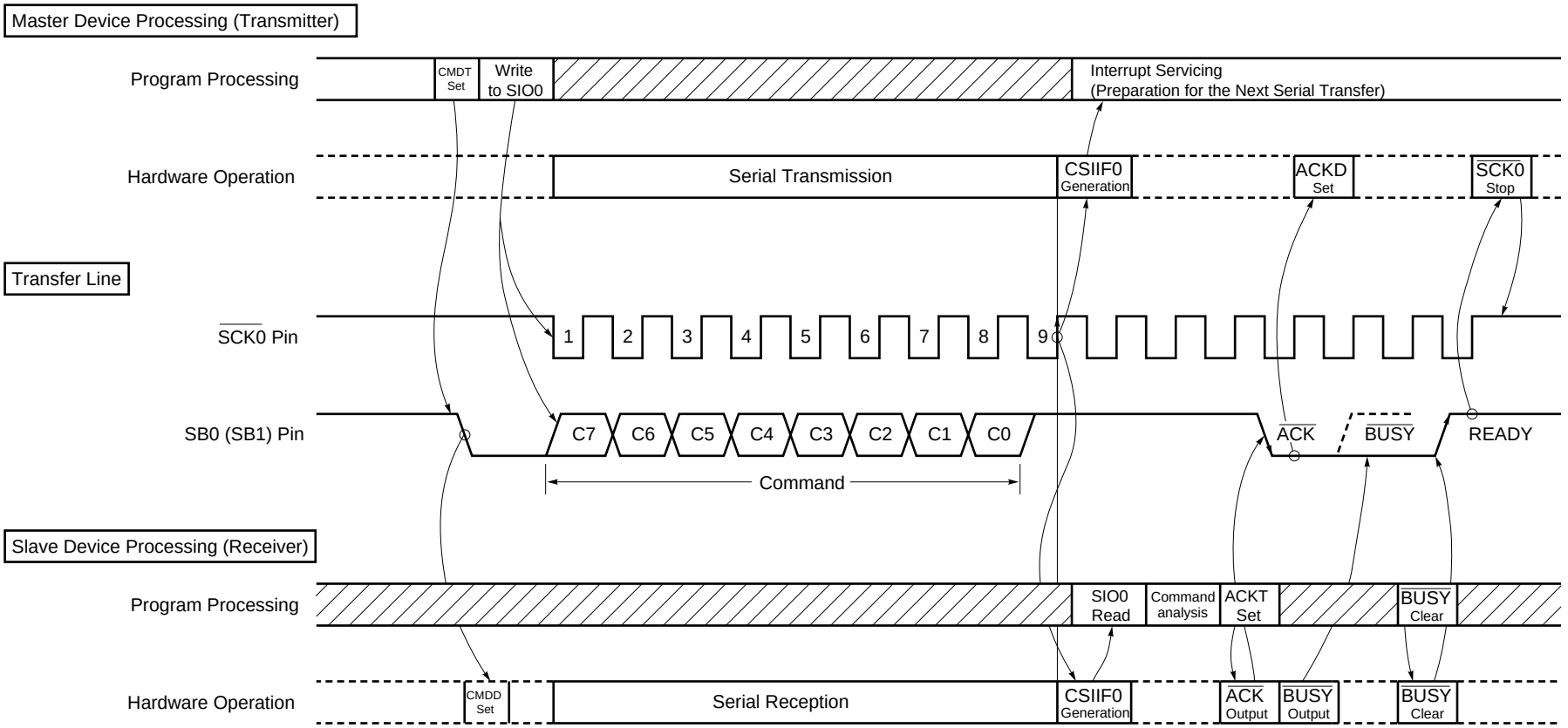


Figure 12-30. Data Transmission from Master Device to Slave Device

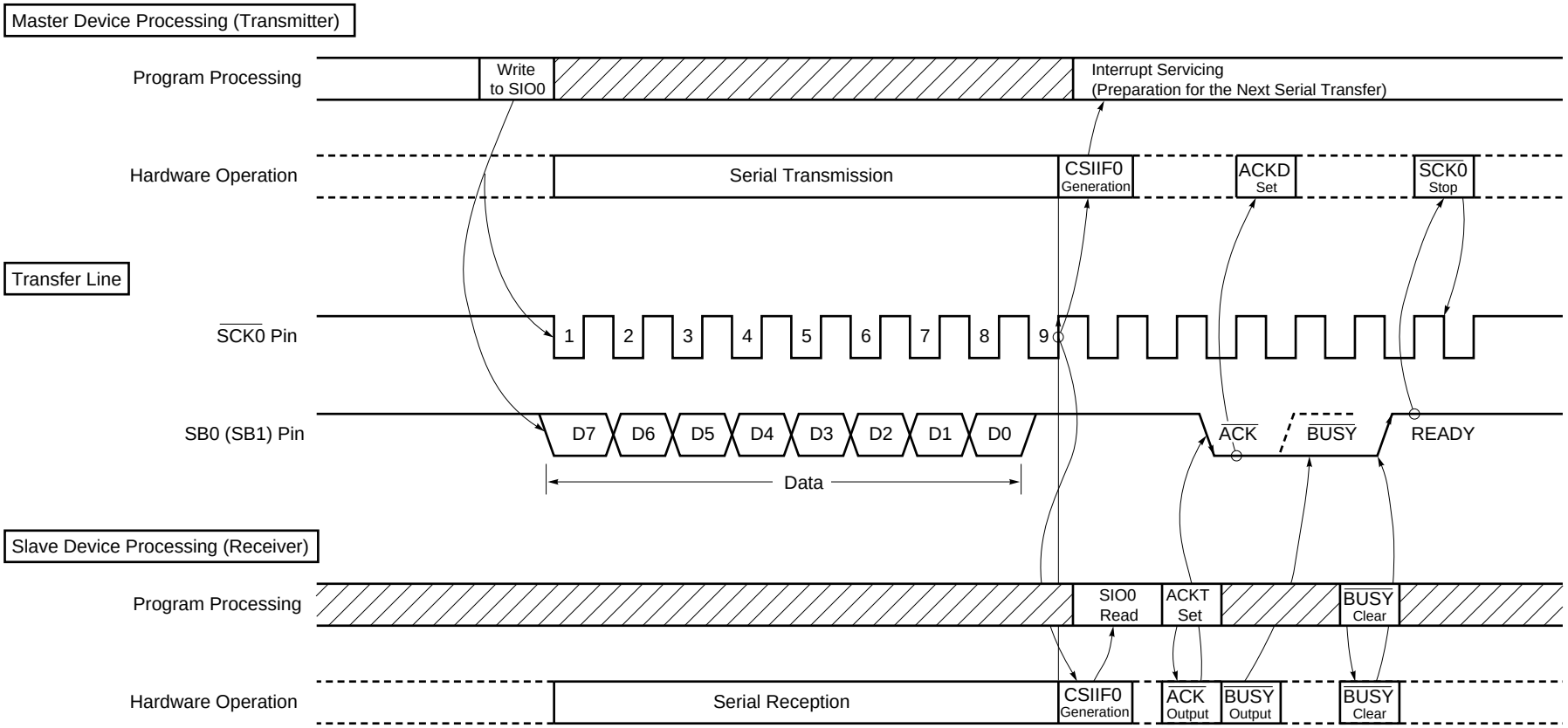
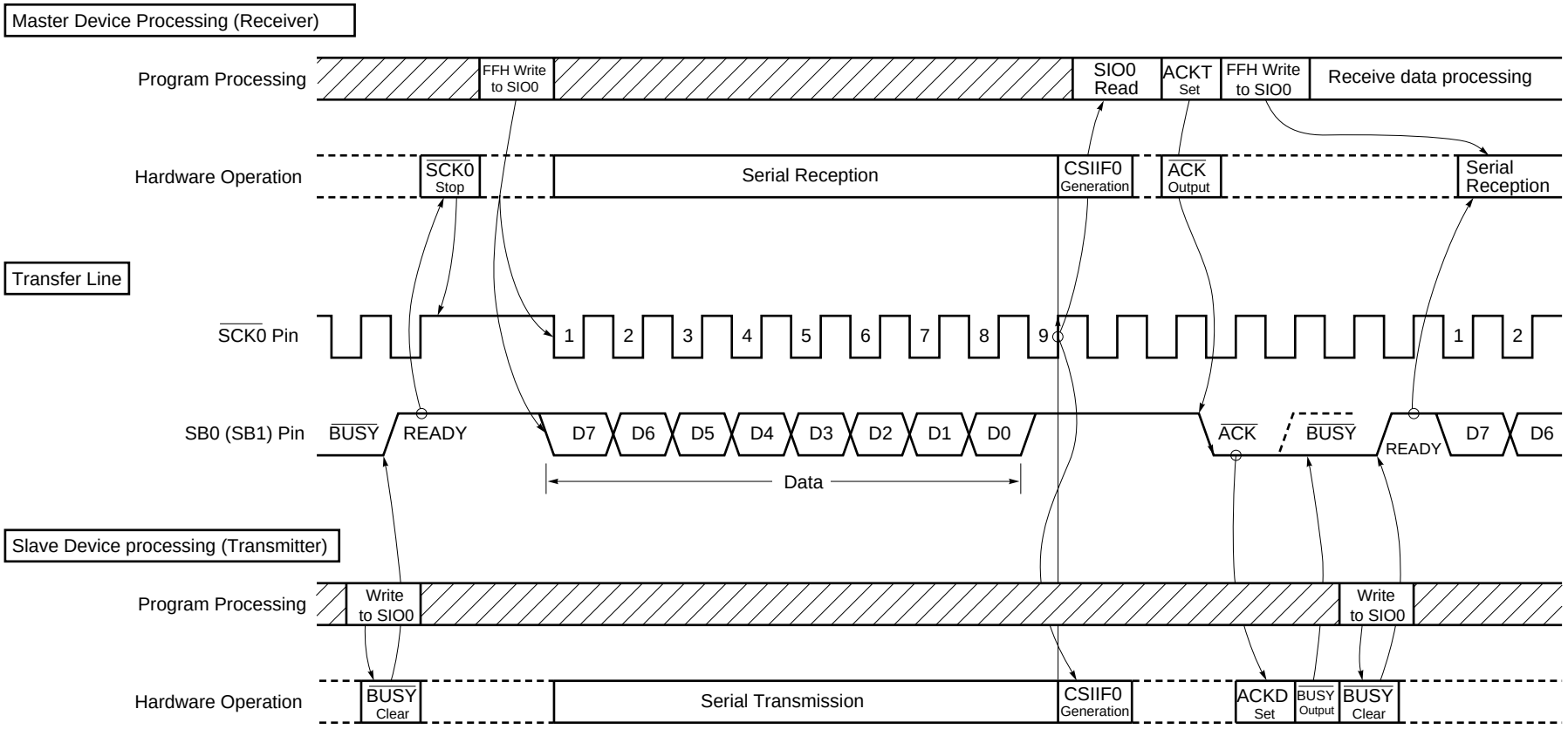


Figure 12-31. Data Transmission from Slave Device to Master Device



(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 $\overline{SCK0}$ 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 high-impedance state for data reception, write FFH to SIO0 in advance.
However, when the make-up function specification bit (WUP) = 1, the N-ch open-drain output is always made 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.

Be sure to set the pins used to input/output data after input of the RESET signal and before serial transfer of 1 byte, as follows:

<1> Set the output latches of P25 and P26 to 1.
<2> Set bit 0 (RELT) of the serial bus interface control register (SBIC) to 1.
<3> Clear the output latches of P25 and P26, which have been set to 1 above, to 0.

(10) Method to judge busy state of a slave

When device is in the master mode, follow the method below to judge whether slave device is in the busy state or not.

<1> Detect acknowledge signal ($\overline{ACK}$) 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 (INTCSI0) 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) In the SBI mode, SBI outputs the $\overline{BUSY}$ signal after the $\overline{BUSY}$ has been cleared and until the next serial clock falls. If WUP is set to 1 by mistake during this period, $\overline{BUSY}$ will not be cleared. To set WUP to 1, therefore, clear $\overline{BUSY}$, and make sure that the SB0 (SB1) pin has gone high.

(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 reset input.

<1> Set the P25 and P26 output latches to 1.
<2> Set bit 0 (RELT) of the serial bus control register to 1.
<3> Reset the P25 and P26 output latches from 1 to 0.

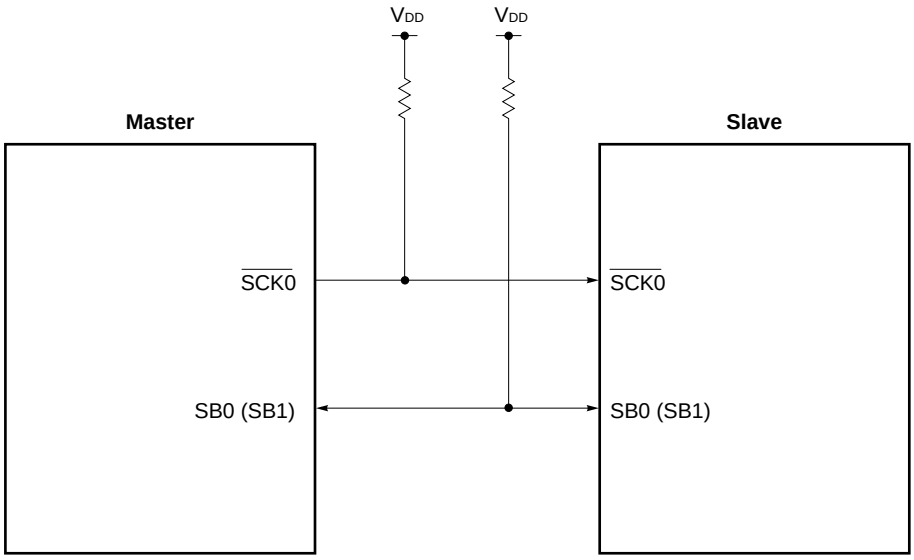
(e) A positive transition of the SB0 (SB1) pin from low to high or high to low is recognized as a bus release signal or a command signal when the $\overline{SCK0}$ line is high. If the change timing of the bus is shifted due to the influence of the board capacitance, data that is transmitted may be identified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

12.4.4 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 ($\overline{SCK0}$) and serial data input/output (SB0 or SB1).

Figure 12-32. Serial Bus Configuration Example Using 2-Wire Serial I/O Mode



(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.

Reset input sets CSIM0 to 01H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	1	FF61H	01H	R/W Note 1

R/W

CSIM01	Serial Interface Channel 0 Clock Selection
0	Input Clock to $\overline{\text{SCK0}}$/SCL/P27 pin from off-chip
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	Operation Mode	Start Bit	SIO/SB0/SDA0/P25 Pin Function	SO0/SB1/SDA1/P26 Pin Function	$\overline{\text{SCK0}}$/SCL/P27 Pin Function
0	×	3-wire Serial I/O mode (Refer to **12.4.2 3-wire serial I/O mode operation**)											
1	0	SBI mode (Refer to **12.4.3 SBI mode operation**)											
1	1	0	Note 2 ×	Note 2 ×	0	0	0	1	2-wire serial I/O mode or I²C bus mode (Refer to **12.4.5**)	MSB	P25 (CMOS I/O)	SB1/SDA1 (N-ch open-drain I/O)	$\overline{\text{SCK0}}$/SCL (N-ch open-drain I/O)
		1	0	0	Note 2 ×	Note 2 ×	0	1			SB0/SDA0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

R/W

WUP	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 data

R

COI	Slave Address Comparison Result Flag Note4
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 enabled

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

PM××: port mode register

P×× : output latch of port

(b) Serial bus interface control register (SBIC)

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

Reset input sets 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.

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.

Reset input sets 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

SIC	INTCSI0 Interrupt Source 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	$\overline{\text{SCK0}}$/SCL/P27 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 CSIIF0: Interrupt request flag corresponding to INTCSI0

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

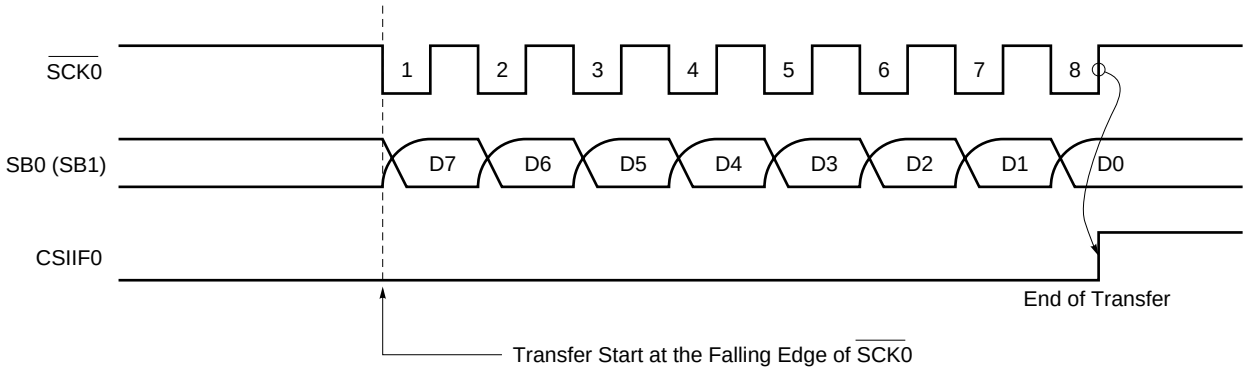
(2) Communication operation

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 ($\overline{SCK0}$). The transmit data is held in the SO0 latch and is output from the SB0/P25 (or SB1/P26) pin on an MSB-first basis. The receive data input from the SB0 (or SB1) pin is latched into the shift register at the rising edge of $\overline{SCK0}$.

Upon termination of 8-bit transfer, the shift register operation stops automatically and the interrupt request flag (CSIIF0) is set.

Figure 12-33. 2-Wire Serial I/O Mode Timings



The SB0 (or SB1) pin specified for the serial data bus is an N-ch open-drain input/output and thus it must be externally connected to a pull-up resistor. Because it is necessary to make the N-ch open-drain output high-impedance state for data reception, write FFH to SIO0 in advance.

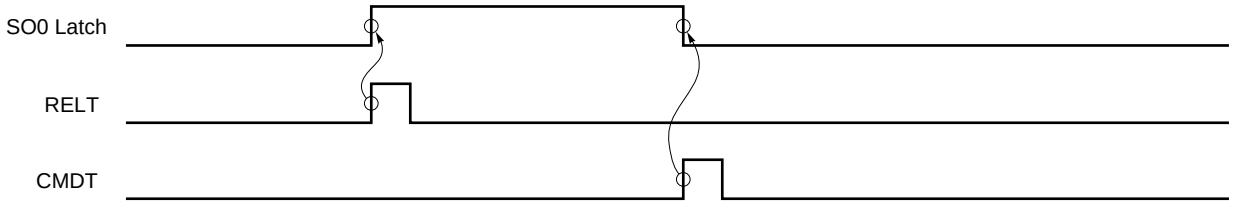
The SB0 (or SB1) pin generates the SO0 latch status and thus the SB0 (or SB1) pin output status can be manipulated by setting the bit 0 (RELT) and bit 1 (CMDT) of the serial bus interface control register (SBIC). However, do not carry out this manipulation during serial transfer.

Control the $\overline{SCK0}$ pin output level in the output mode (internal system clock mode) by manipulating the P27 output latch (refer to **12.4.8 $\overline{SCK0}$/SCL/P27 pin output manipulation**).

(3) Other signals

Figure 12-34 shows RELT and CMDT operations.

Figure 12-34. 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 $\overline{SCK0}$ 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 high-impedance state for 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, serial I/O shift register 0 (SIO0). Thus, transmit error can be detected in the following way.

(a) Method of comparing 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) Method of using 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.

12.4.5 I²C bus mode operation

The I²C bus mode is provided for when communication operations are performed between a single master device and multiple slave devices. This mode configures a serial bus that includes only a single master device, and is based on the clocked serial I/O format with the addition of bus configuration functions, which allows the master device to communicate with a number of (slave) devices using only two lines: serial clock (SCL) line and serial data bus (SDA0 or SDA1) line. Consequently, when the user plans to configure a serial bus which includes multiple microcomputers and peripheral devices, using this configuration results in reduction of the required number of port pins and on-board wires.

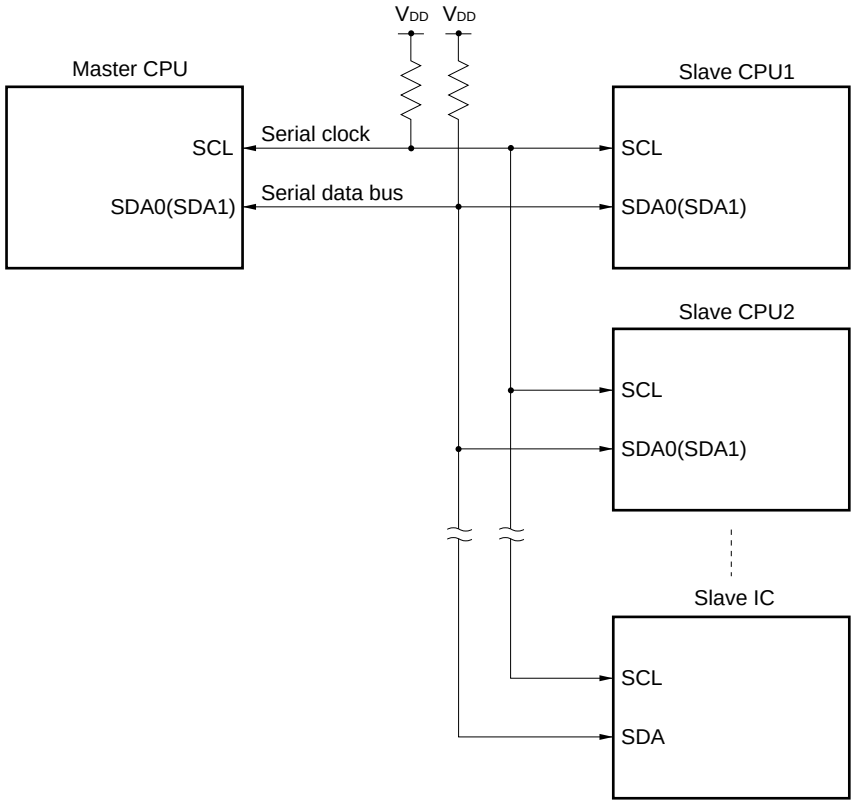
In the I²C bus specification, the master sends start condition, data, and stop condition signals to slave devices through the serial data bus, while slave devices automatically detect and distinguish the type of signals due to the signal detection function incorporated as hardware. This simplifies I²C bus control sections in the application program.

An example of a serial bus configuration is shown in Figure 12-35. This system below is composed of CPUs and peripheral ICs having serial interface hardware that complies with the I²C bus specification.

Note that pull-up resistors are required to connect to both serial clock line and serial data bus line, because N-ch open-drain outputs are used for the serial clock pin (SCL) and the serial data bus pin (SDA0 or SDA1) on the I²C bus.

The signals used in the I²C bus mode are described in Table 12-5.

Figure 12-35. Example of Serial Bus Configuration Using I²C Bus



(1) I^2C bus mode functions

In the I^2C bus mode, the following functions are available.

(a) Automatic identification of serial data

Slave devices automatically detect and identifies start condition, data, and stop condition signals sent in series through the serial data bus.

(b) Chip selection by specifying device addresses

The master device can select a specific slave device connected to the I^2C bus and communicate with it by sending in advance the address data corresponding to the destination device.

(c) Wake-up function

When received address matches with the value of the slave address register (SVA), the slave device internally generates an interrupt signal (Also generated when a stop condition is detected). Therefore, CPUs other than the selected slave device on the I^2C bus can perform independent operations during the serial communication.

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

The master device and a slave device send and receive acknowledge signals to confirm that the serial communication has been executed normally.

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

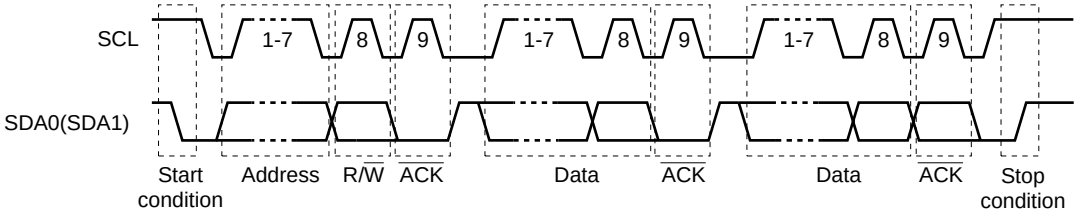
When a slave device is preparing for data transmission or reception and requires more waiting time, the slave device outputs a wait signal on the bus to inform the master device of the wait status.

(2) I^2C bus definition

This section describes the format of serial data communications and functions of the signals used in the I^2C bus mode.

First, the transfer timings of the start condition, data, and stop condition signals, which are output onto the signal data bus of the I^2C bus, are shown in Figure 12-36.

Figure 12-36. I^2C Bus Serial Data Transfer Timing



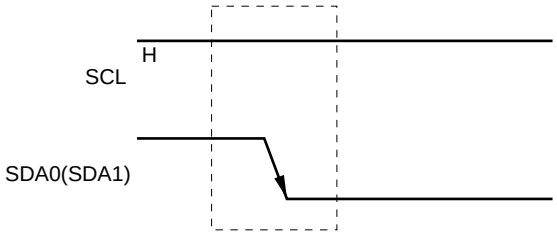
The start condition, slave address, and stop condition signals are output by the master. The acknowledge signal ($\overline{\text{ACK}}$) is output by either the master or the slave device (normally by the device which has received the 8-bit data that was sent). A serial clock (SCL) is continuously supplied from the master device.

(a) Start condition

When the SDA0 (SDA1) pin level is changed from high to low while the SCL pin is high, this transition is recognized as the start condition signal. This start condition signal, which is created using the SCL and SDA0 (or SDA1) pins, is output from the master device to slave devices to initiate a serial transfer. Refer to **12.4.6 Cautions on use of I^2C bus mode**, for details of the start condition output.

The start condition signal is detected by hardware incorporated in slave devices.

Figure 12-37. Start Condition

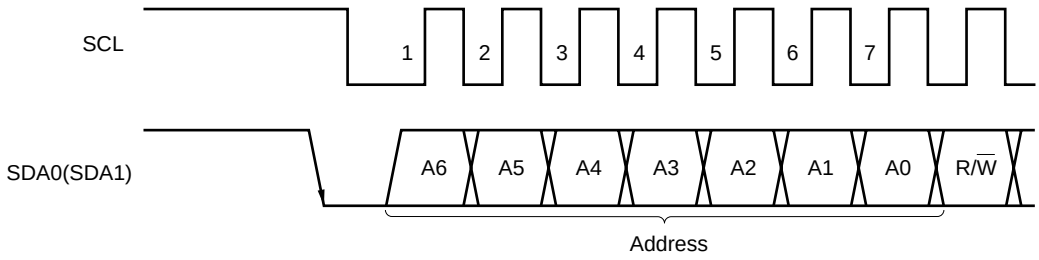


(b) Address

The 7 bits following the start condition signal are defined as an address.

The 7-bit address data is output by the master device to specify a specific slave from among those connected to the bus line. Each slave device on the bus line must therefore have a different address. Therefore, after a slave device detects the start condition, it compares the 7-bit address data received and the data of the slave address register (SVA). After the comparison, only the slave device in which the data are a match becomes the communication partner, and subsequently performs communication with the master device until the master device sends a start condition or stop condition signal.

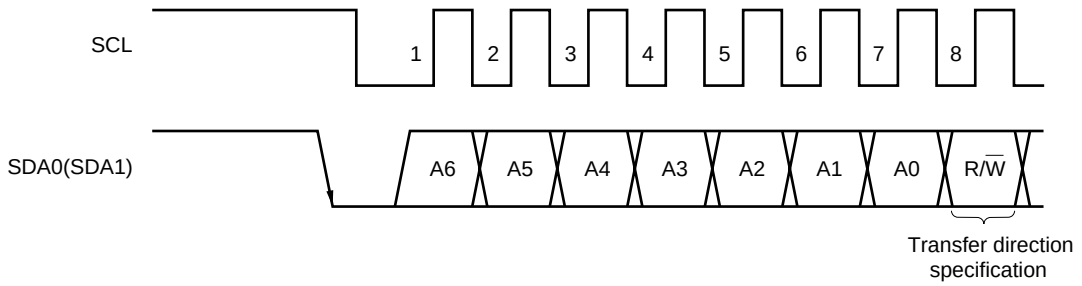
Figure 12-38. Address



(c) Transfer direction specification

The 1 bit that follows the 7-bit address data will be sent from the master device, and it is defined as the transfer direction specification bit. If this bit is 0, it is the master device which will send data to the slave. If it is 1, it is the slave device which will send data to the master.

Figure 12-39. Transfer Direction Specification

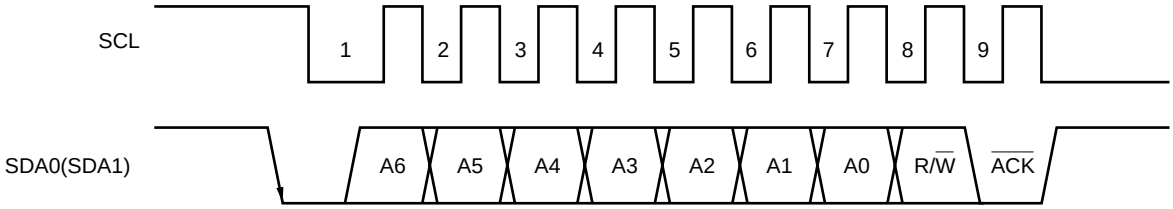


(d) Acknowledge signal ($\overline{ACK}$)

The acknowledge signal indicates that the transferred serial data has definitely been received. This signal is used between the sending side and receiving side devices for confirmation of correct data transfer. In principle, the receiving side device returns an acknowledge signal to the sending device each time it receives 8-bit data. The only exception is when the receiving side is the master device and the 8-bit data is the last transfer data; the master device outputs no acknowledge signal in this case.

The sending side that has tranferred 8-bit data waits for the acknowledge signal which will be sent from the receiving side. If the sending side device receives the acknowledge signal, which means a successful data transfer, it proceeds to the next processing. If this signal is not sent back from the slave device, this means that the data sent has not been received by the slave device, and therefore the master device outputs a stop condition signal to terminate subsequent transmissions.

Figure 12-40. Acknowledge Signal

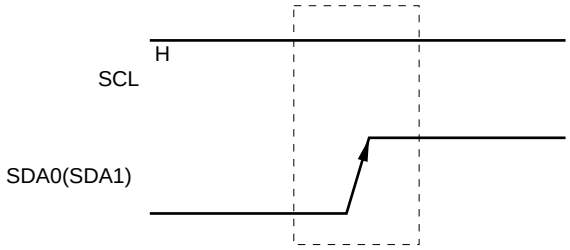


(e) Stop condition

If the SDA0 (SDA1) pin level changes from low to high while the SCL pin is high, this transition is defined as a stop condition signal.

The stop condition signal is output from the master to the slave device to terminate a serial transfer. The stop condition signal is detected by hardware incorporated in the slave device.

Figure 12-41. Stop Condition



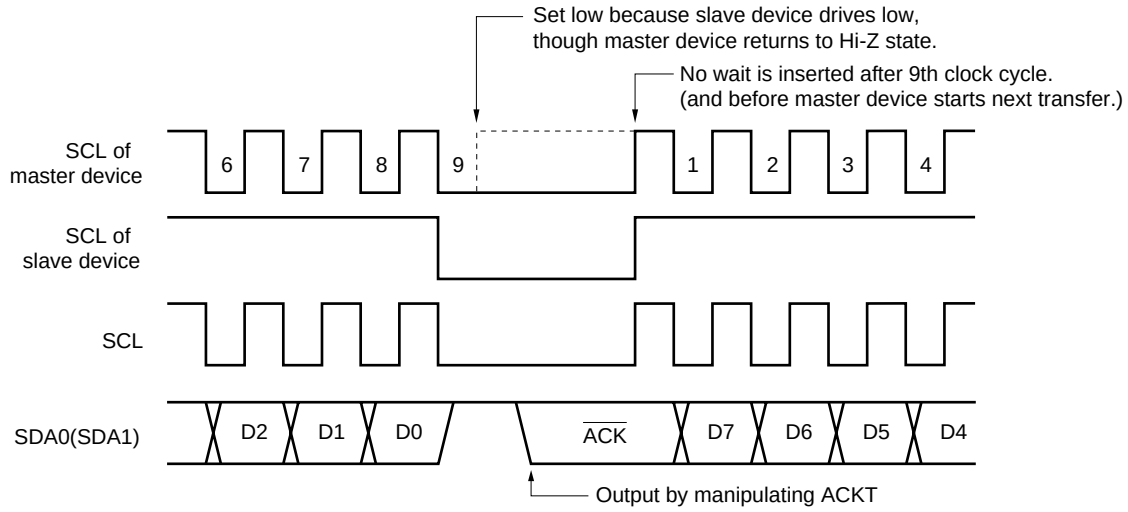
(f) Wait signal ($\overline{\text{WAIT}}$)

The wait signal is output by a slave device to inform the master device that the slave device is in wait state due to preparing for transmitting or receiving data.

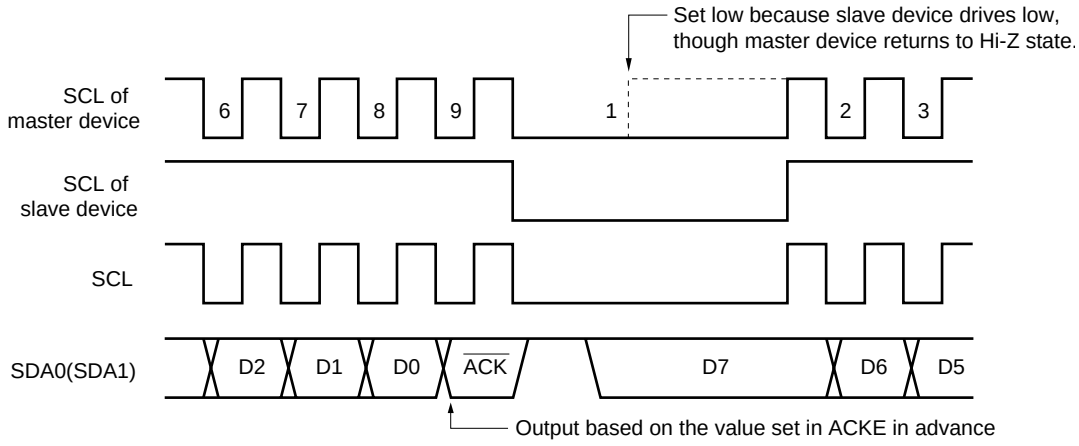
During the wait state, the slave device continues to output the wait signal by keeping the SCL pin low to delay subsequent transfers. When the wait state is released, the master device can start the next transfer. For the releasing operation of slave devices, refer to **12.4.6 Cautions on use of I^2C bus mode**.

Figure 12-42. Wait Signal

(a) Wait of 8 Clock Cycles



(b) Wait of 9 Clock Cycles



(3) Register setting

The I^2C mode setting is performed by 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 by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets 01H.

Symbol	⑦	⑥	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM0	CSIE0	COI	WUP	CSIM04	CSIM03	CSIM02	CSIM01	1	FF60H	01H	R/W Note 1

R/W

CSIM01	Serial Interface Channel 0 Clock Selection
0	Input clock from off-chip to $\overline{SCK0}$/SCL/P27 pin
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	Operation mode	Start bit	SI0/SB0/SDA0/ P25 pin function	SO0/SB1/SDA1/ P26 pin function	$\overline{SCK0}$/SCL/P27 pin function
0	x	3-wire serial I/O mode (refer to **12.4.2 3-wire serial I/O mode operation**)											
1	0	SBI mode (refer to **12.4.3 SBI mode operation**)											
1	1	0	x Note2	x Note2	0	0	0	1	2-wire serial I/O (Refer to **2.4.4**) or I^2C bus mode	MSB	P25 (CMOS I/O)	SB1/SDA1 (N-ch open-drain I/O)	$\overline{SCK0}$/SCL (N-ch open-drain I/O)
		1	0	0	x Note2	x Note2	0	1			SB0/SDA0 (N-ch open-drain I/O)	P26 (CMOS I/O)	

R/W

WUP	Wake-up Function Control Note 3
0	Interrupt request signal generation with each serial transfer in any mode
1	In I^2C bus mode, interrupt request signal is generated when the address data received after start condition detection (when CMDD = 1) matches data in slave address register.

R

COI	Slave Address Comparison Result Flag Note 4
0	Slave address register not equal to data in serial I/O shift register 0
1	Slave address register equal to data in serial I/O shift register 0

R/W

CSIE0	Serial Interface Channel 0 Operation Control
0	Stops operation.
1	Enables operation.

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

2. Can be used freely as a port.

3. When using wake-up function (WUP = 1), set bit 5 (SIC) of the interrupt timing specification register (SINT) to 1. Do not execute write instruction to the serial I/O shift register 0 (SIO0) during WUP = 1.

4. When CSIE0 = 0, COI is 0.

Remark × : don't care

PM×× : port mode register

P×× : output latch of port

(b) Serial bus interface control register (SBIC)

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

Reset input sets 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 1

R/W		
R/W	RELT	Use for stop condition 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		
R/W	CMDT	Use for start condition output. When CMDT = 1, SO latch is cleared to 0. After clearing SO latch, automatically cleared to 0. Also cleared to 0 when CSIE0 = 0.

R	RELD	Stop Condition Detection
	0	Clear Conditions • 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
	1	Setting Condition • When stop condition is detected

R	CMDD	Start Condition Detection
	0	Clear Conditions • When transfer start instruction is executed • When stop condition is detected • When CSIE0 = 0 • When reset input is applied
	1	Setting Condition • When start condition is detected

R/W		
R/W	ACKT	SDA0 (SDA1) is set to low after the set instruction (1) execution (ACKT = 1) before the next SCL falling edge. Used for generating an $\overline{ACK}$ signal by software if the 8-clock wait mode is selected. Cleared to 0 if CSIE = 0 when a transfer by the serial interface is started.

R/W	ACKE	Acknowledge Signal Automatic Output Control Note 2
	0	Disabled (with ACKT enabled). Used when receiving data in the 8-clock wait mode or when transmitting data. Note 3
	1	Enabled. After completion of transfer, acknowledge signal is output in synchronization with the 9th falling edge of SCL clock (automatically output when ACKE = 1). However, not automatically cleared to 0 after acknowledge signal output. Used for reception when the 9-clock wait mode is selected.

R	ACKD	Acknowledge Detection
	0	Clear Conditions • When transfer start instruction is executed • When CSIE0 = 0 • When reset input is applied
	1	Set Conditions • When acknowledge signal is detected at the rising edge of SCL clock after completion of transfer

R/W	BSYE Note 4	Control of N-ch Open-Drain Output for Transmission in I^2C Bus Mode Note 5
	0	Output enabled (transmission)
	1	Output disabled (reception)

Notes 1. Bits 2, 3, and 6 (RELD, CMDD, ACKD) are read-only bits.

2. This setting must be performed prior to transfer start.

3. In the 8-clock wait mode, use ACKT for output of the acknowledge signal after normal data reception.

4. The busy mode can be released by the start of a serial interface transfer or reception of an address signal. However, the BSYE flag is not cleared.

5. When using the wake-up function, be sure to set BSYE to 1.

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

(c) Interrupt timing specification register (SINT)

SINT is set by the 1-bit or 8-bit memory manipulation instruction.

Reset input sets SINT to 00H.

Symbol	7	⑥	⑤	④	③	②	1	0	Address	After Reset	R/W
SINT	0	CLD	SIC	SVAM	CLC	WREL	WAT1	WAT0	FF63H	00H	R/W Note 1

R/W	WAT1	WAT0	Interrupt control by wait Note 2
	0	0	Interrupt service request is generated on rise of 8th $\overline{SCK0}$ clock cycle (clock output is high impedance).
	0	1	Setting prohibited
	1	0	Used in I^2C bus mode (8-clock wait) Generates an interrupt service request on rise of 8th SCL clock cycle. (In case of master device, SCL pin is driven low after output of 8 clock cycles, to enter the wait state. In case of slave device, SCL pin is driven low after input of 8 clock cycles, to require the wait state.)
	1	1	Used in I^2C bus mode (9-clock wait) Generates an interrupt service request on rise of 9th SCL clock cycle. (In case of master device, SCL pin is driven low after output of 9 clock cycles, to enter the wait state. In case of slave device, SCL pin is driven low after input of 9 clock cycles, to require the wait state.)

R/W	WREL	Wait release control
	0	Indicates that the wait state has been released.
	1	Releases the wait state. Automatically cleared to 0 after releasing the wait state. This bit is used to release the wait state set by means of WAT0 and WAT1.

R/W	CLC	Clock level control
	0	Used in I^2C bus mode. In cases other than serial transfer, SCL pin output is driven low.
	1	Used in I^2C bus mode. In cases other than serial transfer, SCL pin output is set to high impedance. (Clock line is held high.) Used by master device to generate the start condition and stop condition signals.

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

R/W	SIC	INTCSI0 interrupt source selection Note 3
	0	CSIIF0 is set to 1 after end of serial interface channel 0 transfer.
	1	CSIIF0 is set to 1 after end of serial interface channel 0 transfer or when stop condition is detected.

R	CLD	$\overline{SCK0}$/SCL/P27 pin level Note 4
	0	Low level
	1	High level

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

2. When the I^2C bus mode is used, be sure to set 1 and 0, or 1 and 1 in WAT0 and WAT1, respectively.

3. When using the wake-up function in I^2C mode, be sure to set SIC to 1.

4. When CSIE0 = 0, CLD is 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

A list of signals in the I^2C bus mode is given in Table 12-5.

Table 12-5. Signals in I^2C Bus Mode

Signal Name	Description	
Start condition	Definition :	SDA0 (SDA1) falling edge when SCL is high Note 1
	Function :	Indicates that serial communication starts and subsequent data are address data.
	Signaled by :	Master
	Signaled when :	CMDT is set.
	Affected flag(s) :	CMDD (is set.)
Stop condition	Definition :	SDA0 (SDA1) rising edge when SCL is high Note 1
	Function :	Indicates end of serial transmission.
	Signaled by :	Master
	Signaled when :	RELT is set.
	Affected flag(s) :	RELD (is set) and CMDD (is cleared)
Acknowledge signal (ACK)	Definition :	Low level of SDA0(SDA1) pin during one SCL clock cycle after serial reception
	Function :	Indicates completion of reception of 1 byte.
	Signaled by :	Master or slave
	Signaled when :	ACKT is set with ACKE = 1.
	Affected flag(s) :	ACKD (is set.)
Wait ($\overline{\text{WAIT}}$)	Definition :	Low-level signal output to SCL
	Function :	Indicates state in which serial reception is not possible.
	Signaled by :	Slave
	Signaled when :	WAT1, WAT0 = 1x.
	Affected flag(s) :	None
Serial Clock (SCL)	Definition :	Synchronization clock for output of various signals
	Function :	Serial communication synchronization signal.
	Signaled by :	Master
	Signaled when :	**Note 2**
	Affected flag(s) :	CSIIF0. Note 3
Address (A6 to A0)	Definition :	7-bit data synchronized with SCL immediately after start condition signal
	Function :	Indicates address value for specification of slave on serial bus.
	Signaled by :	Master
	Signaled when :	**Note 2**
	Affected flag(s) :	CSIIF0. Note 3
Transfer direction (R/$\overline{\text{W}}$)	Definition :	1-bit data output in synchronization with SCL after address output
	Function :	Indicates whether data transmission or reception is to be performed.
	Signaled by :	Master
	Signaled when :	**Note 2**
	Affected flag(s) :	CSIIF0. Note 3
Data (D7 to D0)	Definition :	8-bit data synchronized with SCL, not immediately after start condition
	Function :	Contains data actually to be sent.
	Signaled by :	Master or slave
	Signaled when :	**Note 2**
	Affected flag(s) :	CSIIF0. Note 3

Notes 1. The level of the serial clock can be controlled by CLC of interrupt timing specification register (SINT).

2. Execution of instruction to write data to SIO0 when CSIE0 = 1 (serial transfer start directive). In the wait state, the serial transfer operation will be started after the wait state is released.

3. If the 8-clock wait is selected when WUP = 0, CSIIF0 is set at the rising edge of the 8th clock cycle of SCL. If the 9-clock wait is selected when WUP = 0, CSIIF0 is set at the rising edge of the 9th clock cycle of SCL. If WUP = 1, CSIIF0 is set when an address is received and the address matches the slave address register (SVA) value, and when a stop condition is detected.

(5) Pin configurations

The configurations of the serial clock pin (SCL) and the serial data bus pin SDA0 (SDA1) are shown below.

(a) SCL

Pin for serial clock input/output dual-function pin.

<1> Master N-ch open-drain output

<2> Slave Schmitt input

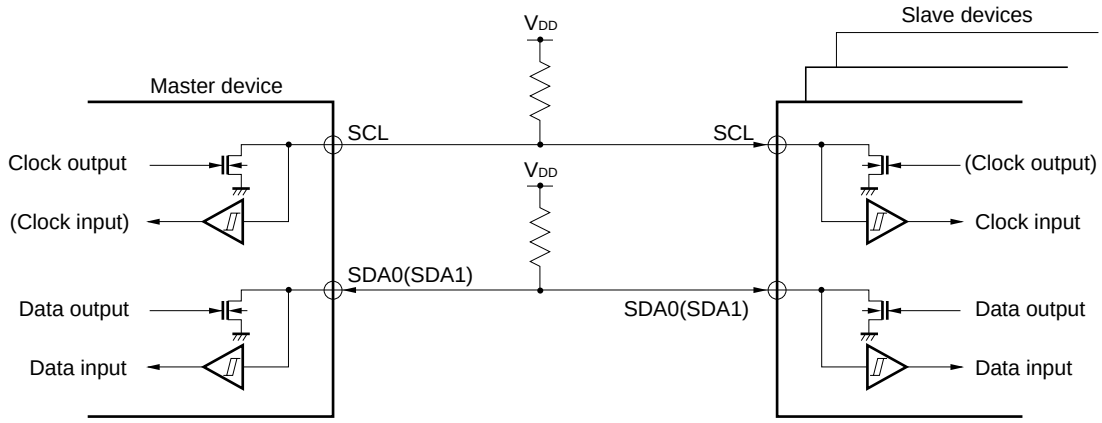
(b) SDA0 (SDA1)

Serial data input/output dual-function pin.

Uses N-ch open-drain output and Schmitt-input buffers for both master and slave devices.

Note that pull-up resistors are required to connect to both serial clock line and serial data bus line, because open-drain buffers are used for the serial clock pin (SCL) and the serial data bus pin (SDA0 or SDA1) on the I^2C bus.

Figure 12-43. Pin Configuration



Caution Because it is necessary to make N-ch open-drain output high-impedance state while data is being received, set bit 6 (BSYE) of the serial bus interface control register (SBIC) to 1 in advance and write FFH to the serial I/O shift register 0 (SIO0).
When the wake-up function is used (when bit 5 (WUP) of serial the serial operating mode register 0 (CSIM0) is set), do not write FFH to SIO0 before reception. The N-ch open-drain output is always high-impedance state even if FFH is not written to SIO0.

(6) Address match detection method

In the I^2C mode, the master can select a specific slave device by sending slave address data.

Address match detection is performed automatically by the slave device hardware. A slave device address has a slave register (SVA), and compares its contents and the slave address sent from the master device. If they match and the wake-up state (WUP) bit is then 1, interrupt request flag (CSIIF0) is set (it is also set when a stop condition is detected). When using wake-up function (WUP = 1), set SIC to 1.

Caution Whether a slave is selected or not depends on detection of coincidence of the data (address) received after the start condition.
To detect this coincidence, an address coincidence detection interrupt (INTCSI0) that occurs when WUP = 1, is normally used. Therefore, to enable detection of whether a slave is selected or not, be sure that WUP = 1.

(7) Error detection

In the I^2C bus mode, transmission error detection can be performed by the following methods because the serial bus SDA0 (SDA1) status during transmission is also taken into the serial I/O shift register 0 (SIO0) register of the transmitting device.

(a) Comparison of SIO0 data before and after transmission

In this case, a transmission error is judged to have occurred if the two data values are different.

(b) Using the slave address register (SVA)

Transmit data is set in SIO0 and SVA before transmission is performed. After transmission, the COI bit (match signal from the address comparator) of serial operating mode register 0 (CSIM0) is tested: "1" indicates normal transmission, and "0" indicates a transmission error.

(8) Communication operation

In the I^2C bus mode, the master selects the slave device to be communicated with from among multiple devices by outputting address data onto the serial bus.

After the slave address data, the master sends the R/W bit which indicates the data transfer direction, and starts serial communication with the selected slave device.

Data communication timing charts are shown in Figures 12-44 and 12-45.

In the transmitting device, the serial I/O shift register 0 (SIO0) shifts transmission data to the SO latch in synchronization with the falling edge of the serial clock (SCL), the SO0 latch outputs the data on an MSB-first basis from the SDA0 or SDA1 pin to the receiving device.

In the receiving device, the data input from the SDA0 or SDA1 pin is taken into the serial I/O shift register 0 (SIO0) in synchronization with the rising edge of SCL.

(9) Start of transfer

A serial transfer is started by setting transfer data in serial I/O shift register 0 (SIO0) if the following two conditions have been satisfied:

- The serial interface channel 0 operation control bit (CSIE0) = 1.
- After an 8-bit serial transfer, the internal serial clock is stopped or SCL is low.

Cautions 1. Be sure to set CSIE0 to 1 before writing data in SIO0. Setting CSIE0 to 1 after writing data in SIO0 does not initiate transfer operation.

2. Because the N-ch open-drain output must be made high-impedance state during data reception, set bit 7 (BSYE) of the serial bus interface control register (SBIC) to 1 before writing FFH to SIO0.

When the wake-up function is used (when bit 5 (WUP) of serial the serial operating mode register 0 (CSIM0) is set), do not write FFH to SIO0 before reception. The N-ch open-drain output is always high-impedance state even if FFH is not written to SIO0 .

3. If data is written to SIO0 while the slave is in the wait state, that data is held. The transfer is started when SCL is output after the wait state is cleared.

When an 8-bit data transfer ends, serial transfer is stopped automatically and the interrupt request flag (CSIIF0) is set.

Figure 12-44. Data Transmission from Master to Slave (Both Master and Slave Selected 9-Clock Wait) (1/3)

(a) Start Condition to Address

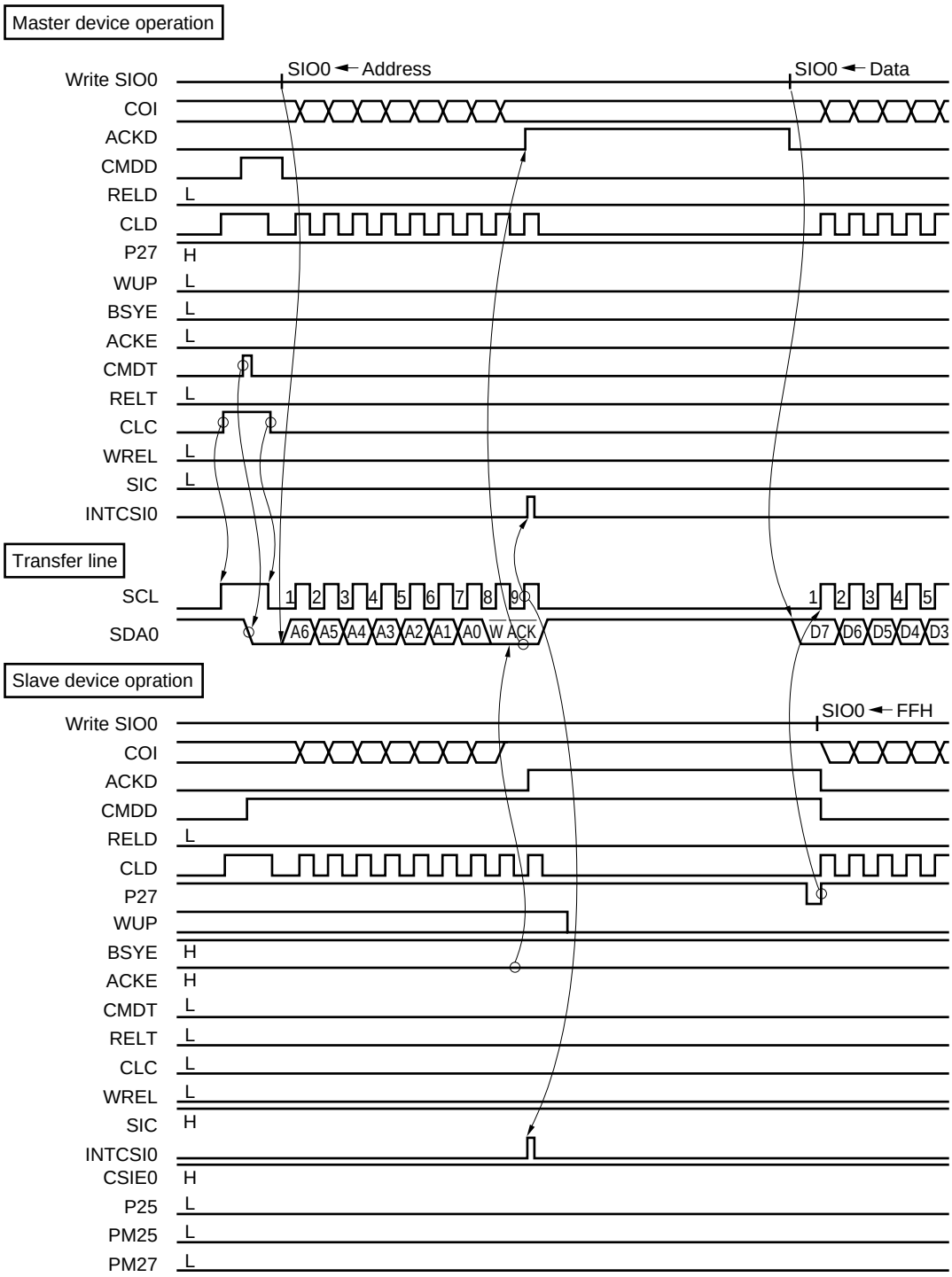


Figure 12-44. Data Transmission from Master to Slave (Both Master and Slave Selected 9-Clock Wait) (2/3)

(b) Data

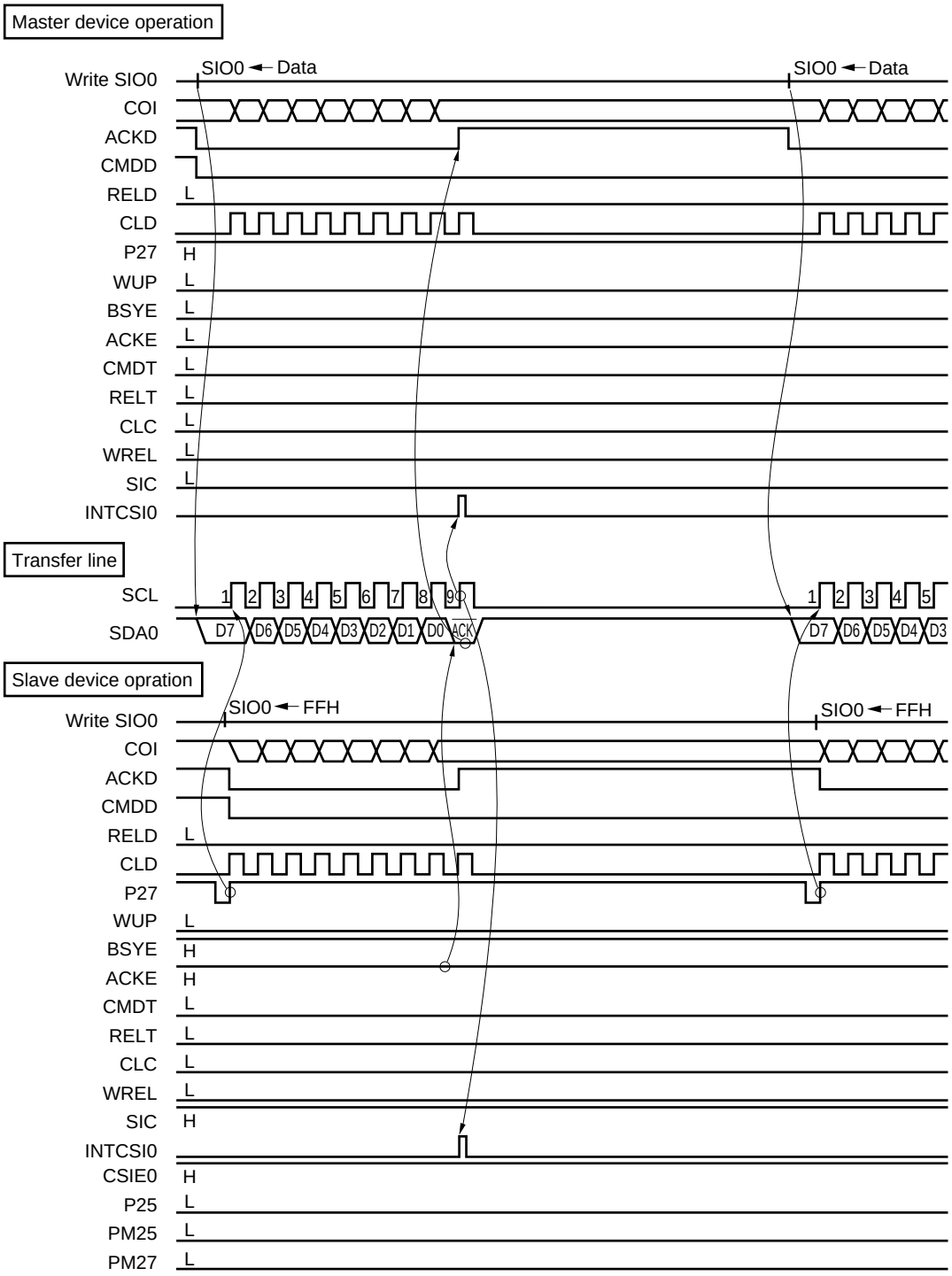


Figure 12-44. Data Transmission from Master to Slave (Both Master and Slave Selected 9-Clock Wait) (3/3)

(c) Stop Condition

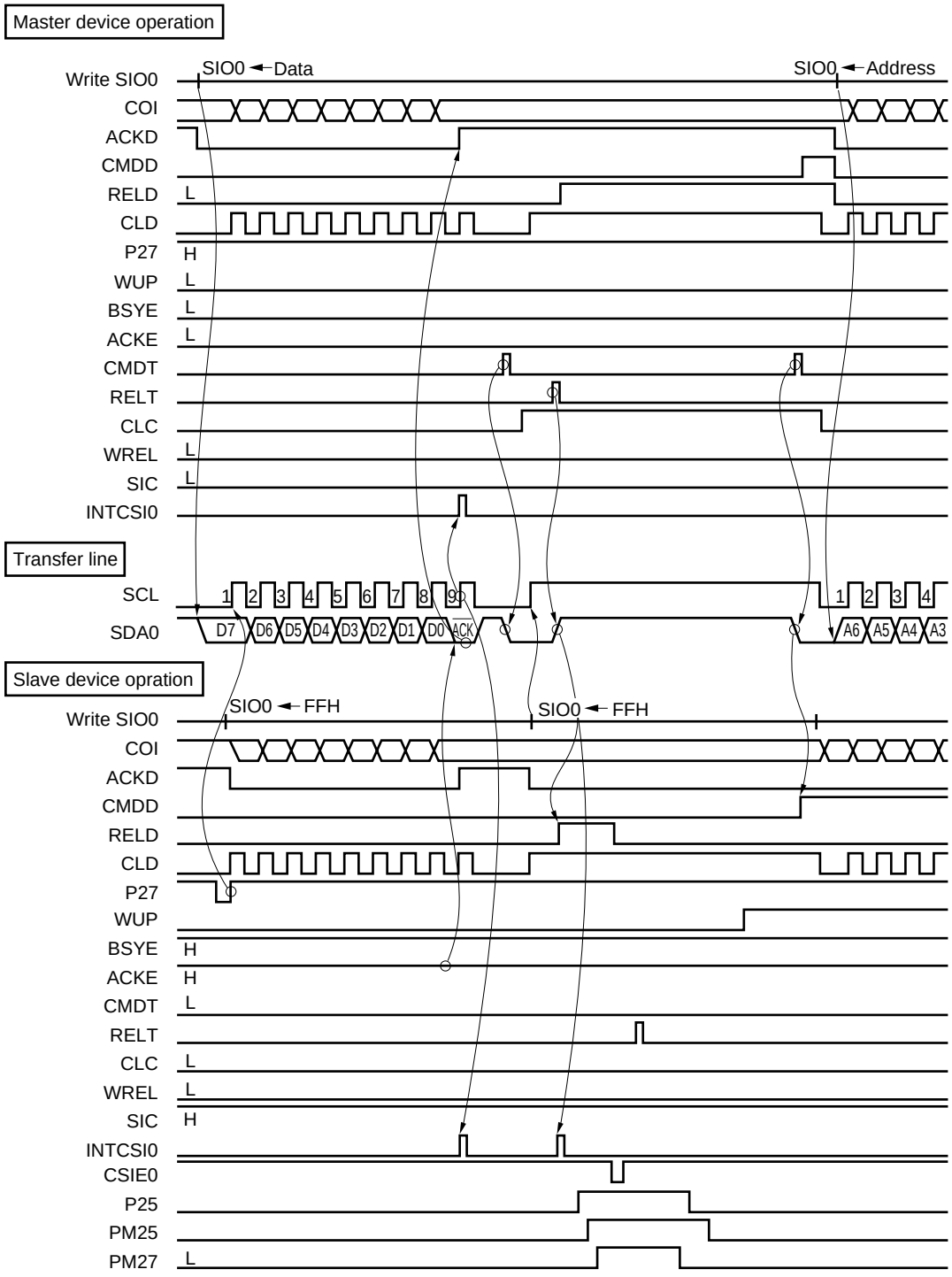


Figure 12-45. Data Transmission from Slave to Master (Both Master and Slave Selected 9-Clock Wait) (1/3)

(a) Start Condition to Address

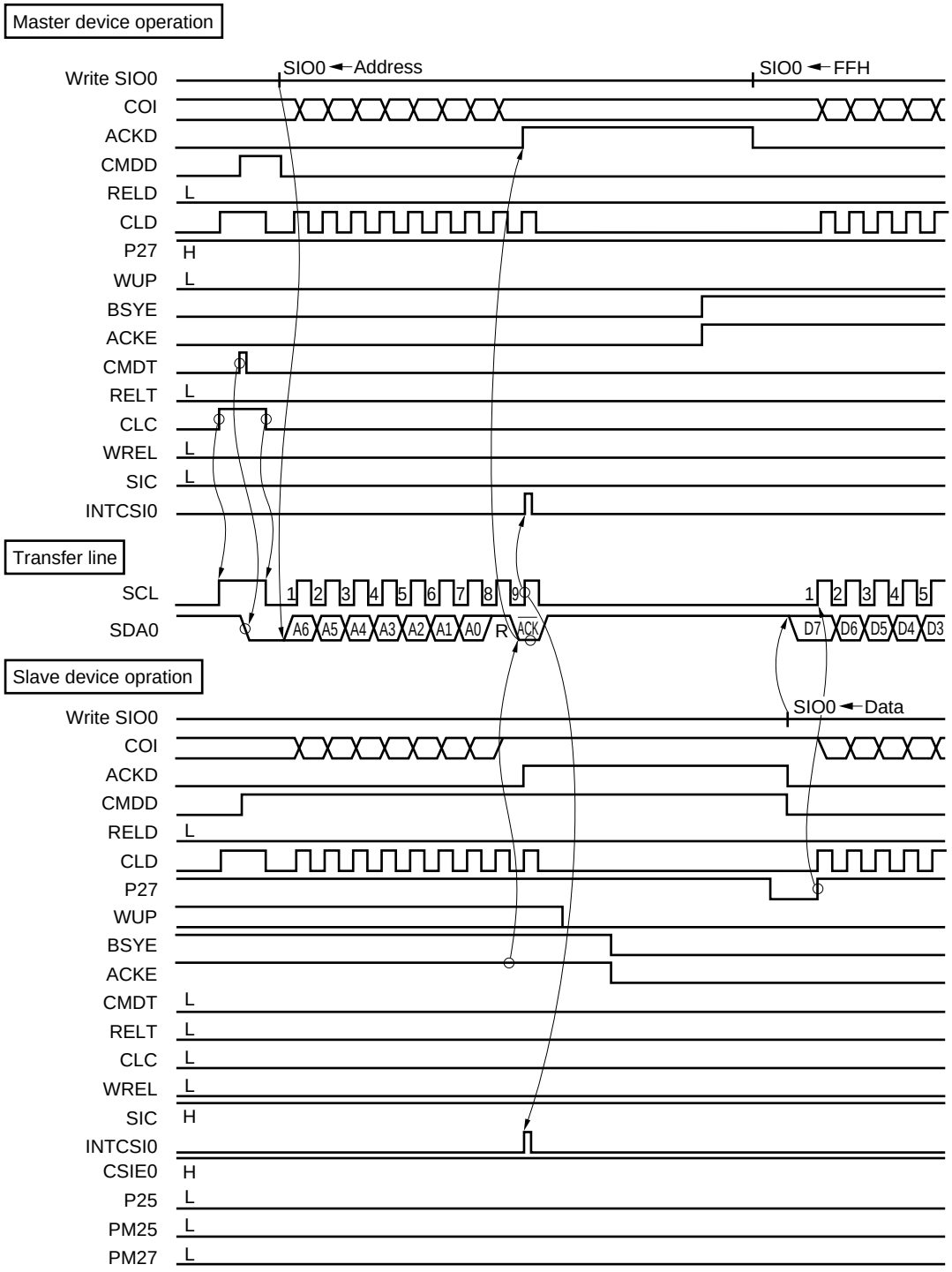


Figure 12-45. Data Transmission from Slave to Master (Both Master and Slave Selected 9-Clock Wait) (2/3)

(b) Data

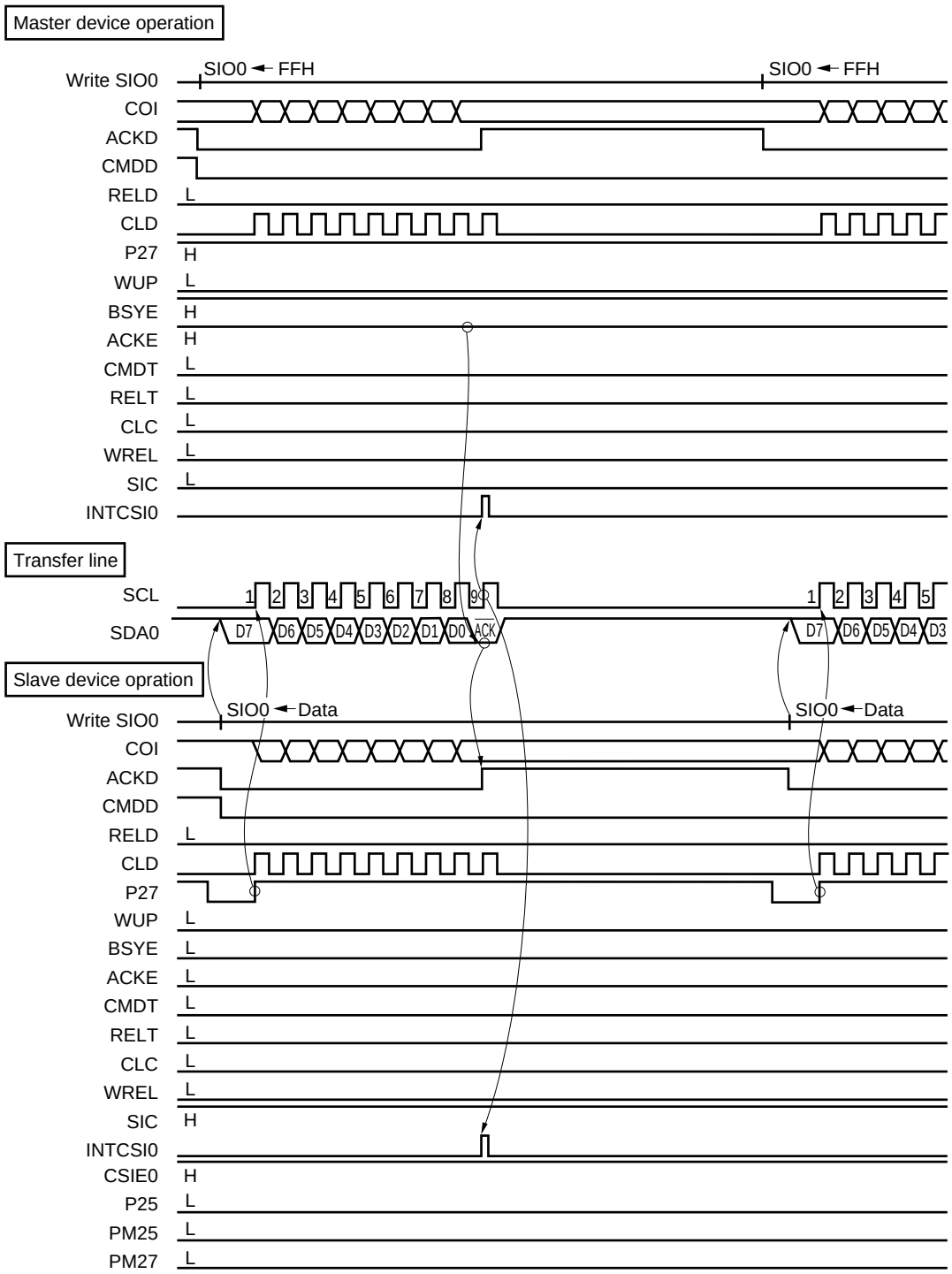
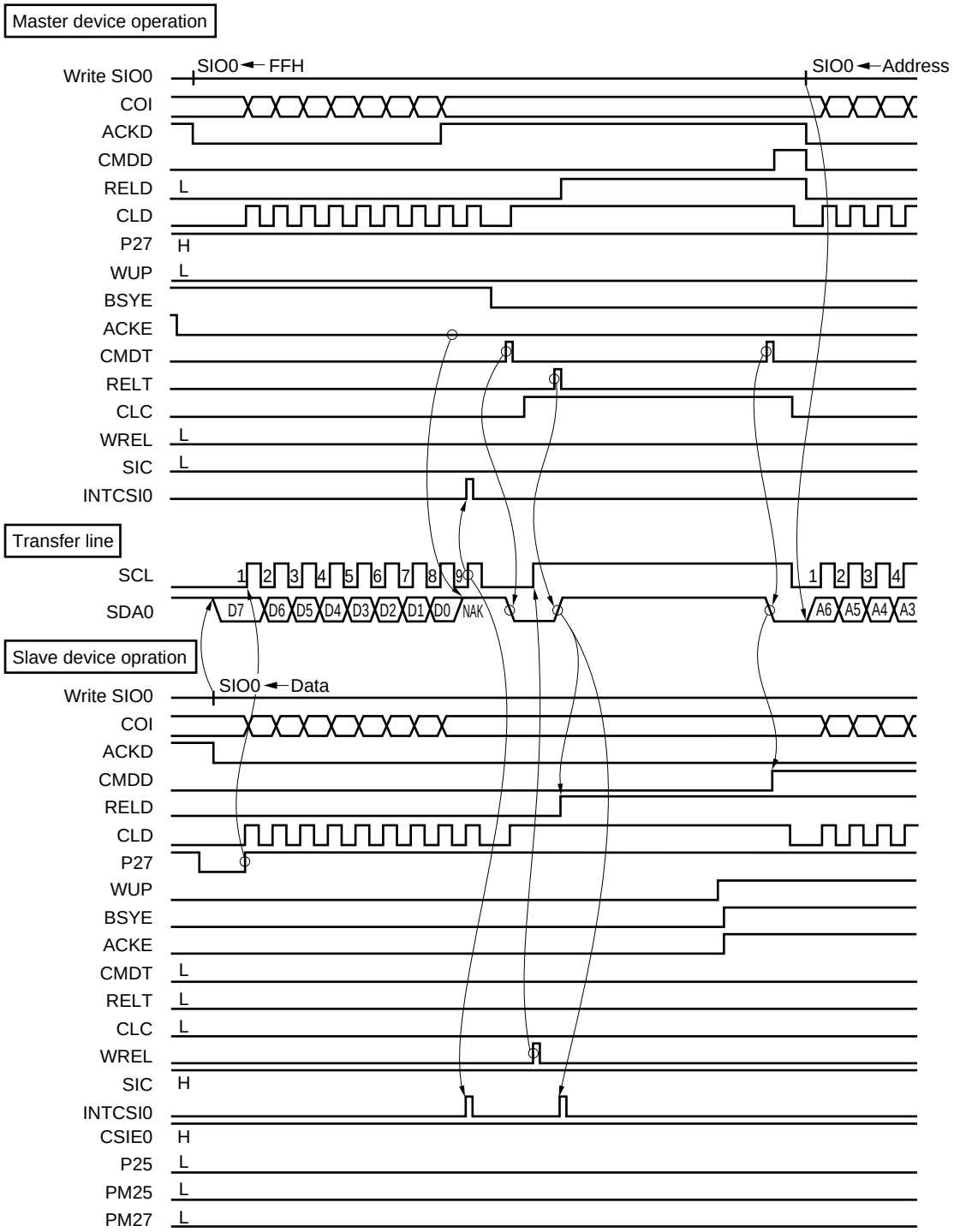


Figure 12-45. Data Transmission from Slave to Master (Both Master and Slave Selected 9-Clock Wait) (3/3)

(c) Stop Condition



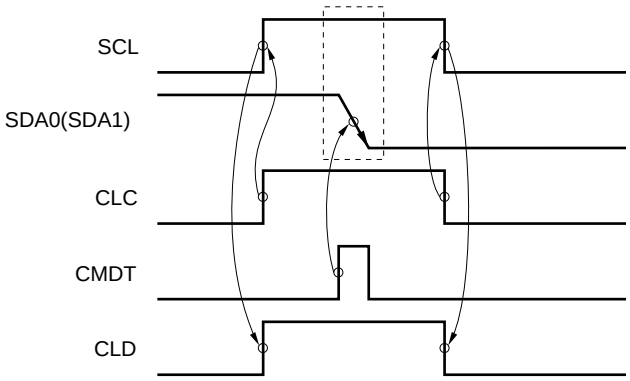
12.4.6 Cautions on use of I^2C bus mode

(1) Start condition output (master)

The SCL pin normally outputs a low-level signal when no serial clock is output. It is necessary to change the SCL pin to high in order to output a start condition signal. Set 1 in CLC of interrupt timing specification register (SINT) to drive the SCL pin high.

After setting CLC, clear CLC to 0 and return the SCL pin to low. If CLC remains 1, no serial clock is output. If it is the master device which outputs the start condition and stop condition signals, confirm that CLD is set to 1 after setting CLC to 1; a slave device may have set SCL to low (wait state).

Figure 12-46. Start Condition Output



(2) Slave wait release

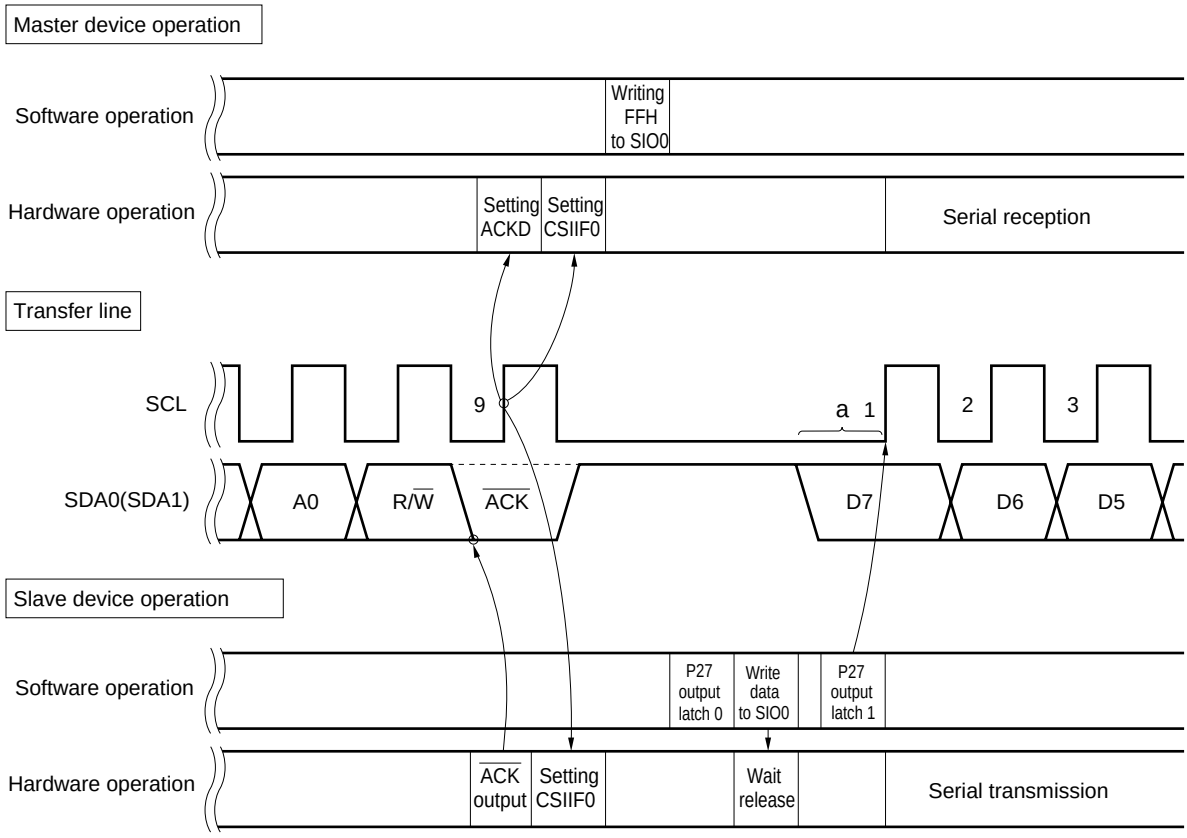
Slave wait release operation is performed by WREL flag setting or execution of an SIO0 write instruction. If the slave sends data, the wait is immediately released by execution of an SIO0 write instruction and the clock rises without the start transmission bit being output in the data line. Therefore, as shown in Figure 12-47, data should be transmitted by manipulating the P27 output latch through the program. At this time, control the low-level width ("a" in Figure 12-47) of the first serial clock at the timing used for setting the P27 output latch to 1 after execution of an SIO0 write instruction.

In addition, if the acknowledge signal from the master is not output (if data transmission from the slave is completed), set 1 in the WREL flag of SINT and release the wait.

If the slave receives data, after execution of an SIO0 write instruction, it is not necessary to manipulate the P27 output latch because the data to be received has already been output on the data line even if the wait is released.

For these timings, see Figures 12-44 and 12-45.

Figure 12-47. Slave Wait Release (Transmission)



(3) Reception completion of slave

During processing of reception completion by a slave device, confirm the statuses of CMDD and COI (if CMDD = 1). This procedure is necessary to use the wake-up function normally. If an uncertain amount of data is sent from the master device, the slave device cannot determine whether the start condition signal or the data will be sent from the master. This may disable use of the wake-up function.

12.4.7 Restrictions in using I²C bus mode

The following restrictions must be observed when using the μPD178018A subseries.

- **Restrictions when using the μPD178018A subseries as slave device in I²C bus mode**

Devices : μPD178004A, 178006A, 178016A, 178018A, 178P018A, IE-178018-R-EM

Description : If the wake-up function is executed (by setting the WUP flag (bit 5 of serial operating mode register 0 (CSIM0)) to 1) in the serial transfer status[Note], data between other slave device and the master is identified as an address. If that data matches with the slave address of the μPD178018A subseries, therefore, the μPD178018A subseries participates in communication, destroying the communicated data.

Note The serial transfer status is the status after the serial I/O shift register 0 (SIO0) has been written until the interrupt request flag (CSIIF0) is set to 1 due to the end of serial transfer.

Preventive measures : The above problem can be avoided by modifying the program.
Before executing the wake-up function, execute the program shown below that clears the serial transfer status. When executing the wake-up function, do not write an instruction that writes to SIO0. Data can be received during execution of the wake-up function even if such an instruction is not executed.
This program clears the serial transfer status. To clear the serial transfer status, serial interface channel 0 must be stopped once (by clearing the CSIE0 flag (bit 7 of serial operating mode register (CSIM0)) to 0). If serial interface channel 0 is stopped in the I²C bus mode, however, the SCL pin outputs a high level and SDA0 (SDA1) pin outputs a low level. Consequently, communication of the I²C bus may be affected. Therefore, this program makes the SCL and SDA0 (SDA1) pins go into a high-impedance state to prevent the I²C bus from being affected.
Note that, in this example, the SDA0 (/P25) pin is used as the serial data I/O pin.
If the SDA1(/P26) pin is used as the serial data I/O pin, take P2.5 and PM2.5 in the program below as P2.6 and PM2.6.
For the timing of each signal when this program is executed, refer to **Figure 12-44**.

- **Example of program clearing serial transfer status**

```
SET1   P2.5      ; (1)
SET1   PM2.5     ; (2)
SET1   PM2.7     ; (3)
CLR1   CSIE0     ; (4)
SET1   CSIE0     ; (5)
SET1   RELT      ; (6)
CLR1   PM2.7     ; (7)
CLR1   P2.5      ; (8)
CLR1   PM2.5     ; (9)
```

(1) Instruction (5) prevents the SDA0 pin from outputting a low level when the I^2C bus mode is restored. The SDA0 pin goes into a high-impedance state.
(2) Instruction (4) sets the P25(/SDA0) pin in the input mode to prevent the SDA0 line from being affected when the port mode is restored. The input mode is set when instruction (2) is executed.
(3) Instruction (4) sets the P27(/SCL) pin in the input mode to prevent the SCL line from being affected when the port mode is restored. The input mode is set when instruction (3) is executed.
(4) The mode is changed from the I^2C bus mode to the port mode.
(5) The mode is changed from the port mode to the I^2C bus mode.
(6) Instruction (8) prevents the SDA0 pin from outputting a low level.
(7) Because the P27 pin must be set in the output mode in the I^2C bus mode, the P27 pin is set in the output mode.
(8) Because the output latch of the P25 pin must be set to 0 in the I^2C bus mode, the output latch of the P25 pin is set to 0.
(9) Because the P25 pin must be set in the output mode in the I^2C bus mode, the P25 pin is set in the output mode.

Remark RELT: Bit 0 of serial bus interface control register (SBIC)

12.4.8 $\overline{\text{SCK0}}$/SCL/P27 pin output manipulation

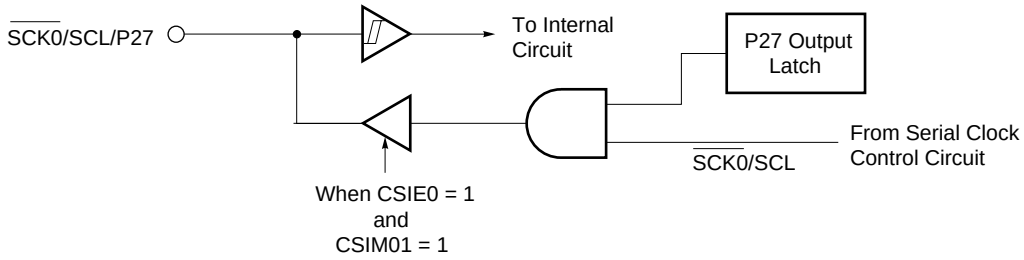
Because the $\overline{\text{SCK0}}$/SCL/P27 pin incorporates an output latch, static output is also possible by software in addition to normal serial clock output.

P27 output latch manipulation enables any value of $\overline{\text{SCK0}}$/SCL to be set by software. (SI0/SB0/SDA0 and SO0/SB1/SDA1 pin to be controlled with the RELT and CMDT bits of SBIC.)

$\overline{\text{SCK0}}$/SCL/P27 pin output manipulating procedure is described below.

(1) Set the serial operating mode register 0 (CSIM0) ($\overline{\text{SCK0}}$/SCL pin enabled for serial operation in the output mode). $\overline{\text{SCK0}}$ = 1 and SCL = 0 with serial transfer suspended.
(2) Manipulate the P27 output latch with a bit manipulation instruction.

Figure 12-48. $\overline{\text{SCK0}}$/SCL/P27 Pin Configuration



[MEMO]

CHAPTER 13 SERIAL INTERFACE CHANNEL 1

13.1 Serial Interface Channel 1 Functions

Serial interface channel 1 employs the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

(1) Operation stop mode

This mode is used when serial transfer is not carried out to reduce power consumption.

(2) 3-wire serial I/O mode (MSB/LSB first switchable)

This mode is used for 8-bit data transfer using three lines, each for serial clock ($\overline{\text{SCK1}}$), serial output (SO1) and serial input (SI1).

The 3-wire serial I/O mode enables simultaneous transmission/reception and so decreases the data transfer processing time.

Since the start bit of 8-bit data to undergo serial transfer is switchable between MSB and LSB, connection is enabled with either start bit device.

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 such as the 75X/XL, 78K and 17K series.

(3) 3-wire serial I/O mode with automatic transmit/receive function (MSB-/LSB first switchable)

This mode has an automatic transmit/receive function in addition to the functions in **(2)** above.

The automatic transmit/receive function is used to transmit/receive data with a maximum of 32 bytes. This function enables the hardware to transmit/receive data to/from the OSD (On Screen Display) device and a device with built-in display controller/driver independently of the CPU, thus the software load can be alleviated.

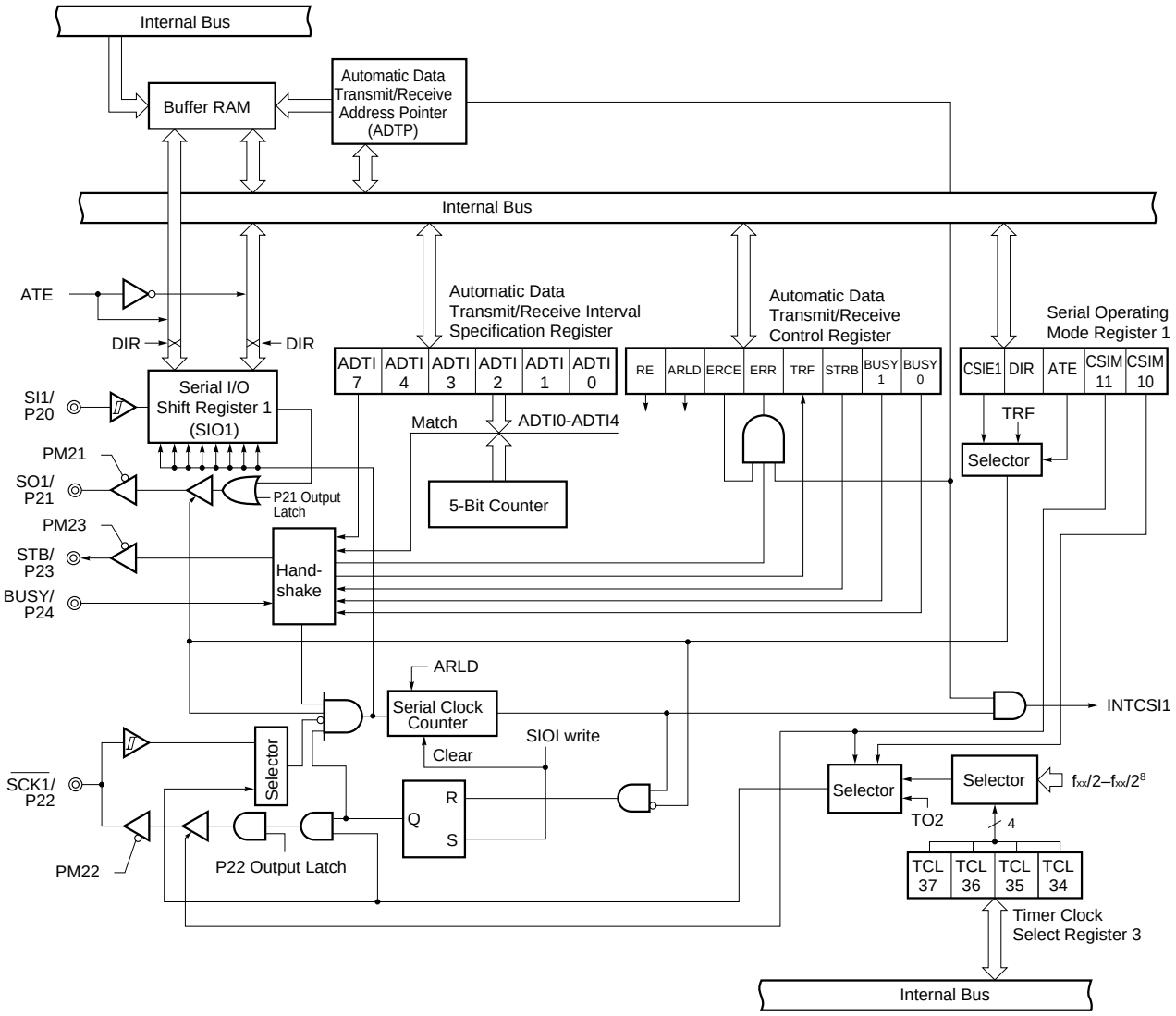
13.2 Serial Interface Channel 1 Configuration

Serial interface channel 1 consists of the following hardware.

Table 13-1. 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 specification register (ADTI) Port mode register 2 (PM2)

Figure 13-1. 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 the 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.

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 (transmit data byte –1) while the automatic transmit/receive function is activated. As data is transferred/received, it is automatically decremented.

ADTP is set with an 8-bit memory manipulation instruction. The high-order 3 bits must be set to 0.

Reset input sets 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 to check whether 8-bit data has been transmitted/received.

13.3 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 specification 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.

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 13-2. 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

TCL37	TCL36	TCL35	TCL34	Serial Interface Channel 1 Serial Clock Selection		
					MCS = 1	MCS = 0
0	1	1	0	$f_{XX}/2$	Setting prohibited	$f_X/2^2$ (1.13 MHz)
0	1	1	1	$f_{XX}/2^2$	$f_X/2^2$ (1.13 MHz)	$f_X/2^3$ (563 kHz)
1	0	0	0	$f_{XX}/2^3$	$f_X/2^3$ (563 kHz)	$f_X/2^4$ (281 kHz)
1	0	0	1	$f_{XX}/2^4$	$f_X/2^4$ (281 kHz)	$f_X/2^5$ (141 kHz)
1	0	1	0	$f_{XX}/2^5$	$f_X/2^5$ (141 kHz)	$f_X/2^6$ (70.3 kHz)
1	0	1	1	$f_{XX}/2^6$	$f_X/2^6$ (70.3 kHz)	$f_X/2^7$ (35.2 kHz)
1	1	0	0	$f_{XX}/2^7$	$f_X/2^7$ (35.2 kHz)	$f_X/2^8$ (17.6 kHz)
1	1	0	1	$f_{XX}/2^8$	$f_X/2^8$ (17.6 kHz)	$f_X/2^9$ (8.8 kHz)
Other than above				Setting prohibited		

Caution When rewriting other data to TCL3 , stop the serial transfer operation beforehand.

Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. MCS : Oscillation mode selection register (OSMS) bit 0

4. Figures in parentheses apply to operation with f_X = 4.5 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.

Reset input sets CSIM1 to 01H.

Figure 13-3. 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	CSIM11	1	FF68H	01H	R/W

CSIM11	Serial Interface Channel 1 Clock Selection
0	Clock externally input to $\overline{\text{SCK1}}$ pin Note 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		

CSIE 1	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 I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
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 bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. P20 (CMOS input/output) when only transmitter is used. (Set bit 7 (RE) of the ADTC to 0).

Remark × : don't care

PM×× : port mode register

P×× : output latch of port

(3) Automatic data transmit/receive control register (ADTC)

This register sets automatic receive enable/disable, the operating mode, strobe output enable/disable, busy input enable/disable and displays automatic transmit/receive execution.

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

Reset input sets ADTC to 00H.

Figure 13-4. 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 checked by using TRF, not CSIIF1 of 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 specification 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.

Reset input sets ADTI to 00H.

Figure 13-5. Automatic Data Transmit/Receive Interval Specification 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_{XX} = 4.5 MHz Operation)	
					Minimum Note 2	Maximum Note 2
0	0	0	0	0	20.4 μs + 0.5/f_{SCK}	22.2 μs + 1.5/f_{SCK}
0	0	0	0	1	34.7 μs + 0.5/f_{SCK}	36.4 μs + 1.5/f_{SCK}
0	0	0	1	0	48.9 μs + 0.5/f_{SCK}	50.7 μs + 1.5/f_{SCK}
0	0	0	1	1	63.1 μs + 0.5/f_{SCK}	64.9 μs + 1.5/f_{SCK}
0	0	1	0	0	77.3 μs + 0.5/f_{SCK}	79.1 μs + 1.5/f_{SCK}
0	0	1	0	1	91.5 μs + 0.5/f_{SCK}	93.3 μs + 1.5/f_{SCK}
0	0	1	1	0	105.8 μs + 0.5/f_{SCK}	107.5 μs + 1.5/f_{SCK}
0	0	1	1	1	120.0 μs + 0.5/f_{SCK}	121.8 μs + 1.5/f_{SCK}
0	1	0	0	0	134.2 μs + 0.5/f_{SCK}	136.0 μs + 1.5/f_{SCK}
0	1	0	0	1	148.4 μs + 0.5/f_{SCK}	150.2 μs + 1.5/f_{SCK}
0	1	0	1	0	162.6 μs + 0.5/f_{SCK}	164.4 μs + 1.5/f_{SCK}
0	1	0	1	1	176.9 μs + 0.5/f_{SCK}	178.6 μs + 1.5/f_{SCK}
0	1	1	0	0	191.1 μs + 0.5/f_{SCK}	192.9 μs + 1.5/f_{SCK}
0	1	1	0	1	205.3 μs + 0.5/f_{SCK}	207.1 μs + 1.5/f_{SCK}
0	1	1	1	0	219.5 μs + 0.5/f_{SCK}	221.3 μs + 1.5/f_{SCK}
0	1	1	1	1	233.7 μs + 0.5/f_{SCK}	235.5 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 $2/f_{SCK}$.

$$\text{Minimum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}, \quad \text{Maximum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Be sure to set bits 5 and 6 to 0.

3. When controlling the interval time of automatic transmit/receive data transfer by using ADTI, busy control is invalid. (Refer to 13.4.3 (4) (a) Busy control option.)

Remark f_{XX} : System clock frequency (f_X or $f_X/2$)

f_X : System clock oscillation frequency

f_{SCK} : Serial clock frequency

Figure 13-5. Automatic Data Transmit/Receive Interval Specification 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_{XX} = 4.5 MHz Operation)	
					Minimum[Note]	Maximum[Note]
1	0	0	0	0	248.0 μs + 0.5/f_{SCK}	250.0 μs + 1.5/f_{SCK}
1	0	0	0	1	262.2 μs + 0.5/f_{SCK}	264.0 μs + 1.5/f_{SCK}
1	0	0	1	0	276.4 μs + 0.5/f_{SCK}	278.2 μs + 1.5/f_{SCK}
1	0	0	1	1	290.6 μs + 0.5/f_{SCK}	292.4 μs + 1.5/f_{SCK}
1	0	1	0	0	304.8 μs + 0.5/f_{SCK}	306.6 μs + 1.5/f_{SCK}
1	0	1	0	1	319.0 μs + 0.5/f_{SCK}	320.8 μs + 1.5/f_{SCK}
1	0	1	1	0	333.2 μs + 0.5/f_{SCK}	335.1 μs + 1.5/f_{SCK}
1	0	1	1	1	347.5 μs + 0.5/f_{SCK}	349.3 μs + 1.5/f_{SCK}
1	1	0	0	0	361.7 μs + 0.5/f_{SCK}	363.5 μs + 1.5/f_{SCK}
1	1	0	0	1	375.9 μs + 0.5/f_{SCK}	377.7 μs + 1.5/f_{SCK}
1	1	0	1	0	390.2 μs + 0.5/f_{SCK}	391.9 μs + 1.5/f_{SCK}
1	1	0	1	1	404.4 μs + 0.5/f_{SCK}	406.2 μs + 1.5/f_{SCK}
1	1	1	0	0	418.6 μs + 0.5/f_{SCK}	420.4 μs + 1.5/f_{SCK}
1	1	1	0	1	432.8 μs + 0.5/f_{SCK}	434.6 μs + 1.5/f_{SCK}
1	1	1	1	0	447.0 μs + 0.5/f_{SCK}	448.8 μs + 1.5/f_{SCK}
1	1	1	1	1	461.3 μs + 0.5/f_{SCK}	463.0 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}.

$$\text{Minimum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions
1. **Do not write ADTI during operation of automatic data transmit/receive function.**
2. **Be sure to set bits 5 and 6.**
3. **When controlling the interval time of automatic transmit/receive data transfer by using ADTI, busy control is invalid. (Refer to 13.4.3 (4) (a) Busy control option.)**

Remark f_{XX} : System clock frequency (f_X or f_X/2)
f_X : System clock oscillation frequency
f_{SCK} : Serial clock frequency

13.4 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

13.4.1 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.

Reset input sets CSIM1 to 01H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	1	FF68H	01H	R/W

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 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Note 1 ×	Operation stop	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
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. 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

PM×× : port mode register

P×× : output latch of port

13.4.2 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 such as the 75X/XL, 78K and 17K series.

Communication is carried out with three lines of serial clock ($\overline{SCK1}$), 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.

Reset input sets CSIM1 to 01H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	1	FF68H	01H	R/W

CSIM11	Serial Interface Channel 1 Clock Selection
0	Clock externally input to $\overline{SCK1}$ pin Note 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	SO1 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	$\overline{SCK1}$/P22 Pin Function
0	×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Note 2 ×	Operation stop	Clear	P20 (CMOS I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
1	0	Note 3 1	Note 3 ×	0	0	1	×	Operation enable	Count operation	SI1 Note 3 (Input)	SO1 (CMOS output)	$\overline{SCK1}$ (Input)
	1					0	1					$\overline{SCK1}$ (CMOS output)

Notes 1. If the external clock input has been selected with CSIM11 set to 0, set bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. 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

PM×× : port mode register

P×× : output latch of port

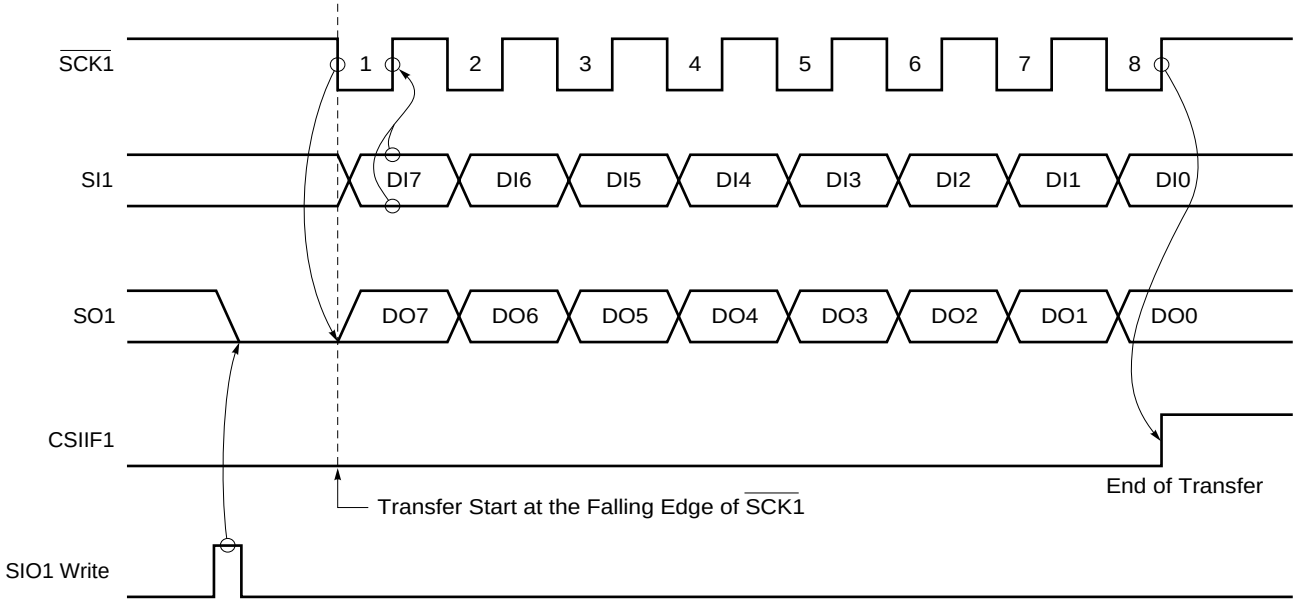
(2) Communication operation

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 ($\overline{SCK1}$). 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 $\overline{SCK1}$.

Upon termination of 8-bit transfer, the SIO1 operation stops automatically and the interrupt request flag (CSIIF1) is set.

Figure 13-6. 3-Wire Serial I/O Mode Timings



Caution SO1 pin becomes low level by SIO1 write.

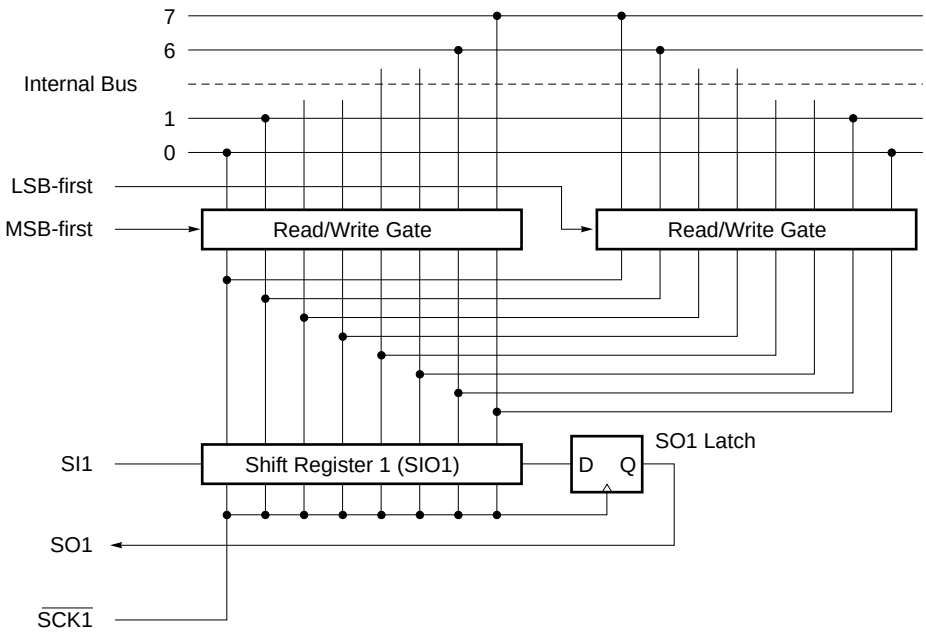
(3) MSB/LSB switching as the start bit

The 3-wire serial I/O mode enables to select transfer to start from MSB or LSB.

Figure 13-7 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 13-7. 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 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.

13.4.3 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 32-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 specification register (ADTI).

(a) Serial operating mode register 1 (CSIM1)

CSIM1 is set with a 1-bit or 8-bit memory manipulation instruction.

Reset input sets CSIM1 to 01H.

Symbol	⑦	6	⑤	4	3	2	1	0	Address	After Reset	R/W
CSIM1	CSIE1	DIR	ATE	0	0	0	CSIM11	1	FF68H	01H	R/W

CSIM11	Serial Interface Channel 1 Clock Selection
0	Clock externally input to $\overline{\text{SCK1}}$ pin Note 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 I/O)	P21 (CMOS I/O)	P22 (CMOS I/O)
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 bit 1 (BUSY1) and bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) to 0, 0.

2. Can be used freely as port function.

3. P20 (CMOS input/output) when only transmitter is used. (Set bit 7 (RE) of ADTC to 0).

Remark × : don't care
PM×× : port mode register
P×× : output latch of port

(b) Automatic data transmit/receive control register (ADTC)

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

Reset input sets 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	×	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 checked by using TRF, not CSIIF1 of 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 (when an external clock is input, hand-shake control cannot be performed).

Remark ×: don't care

(c) Automatic data transmit/receive interval specification register (ADTI)

This register sets the automatic data transmit/receive function data transfer interval.

ADTI is set with a 1-bit or 8-bit memory manipulation instruction. Reset input sets 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_{XX} = 4.5 MHz Operation)	
					Minimum Note 2	Maximum Note 2
0	0	0	0	0	20.4 μs + 0.5/f_{SCK}	22.2 μs + 1.5/f_{SCK}
0	0	0	0	1	34.7 μs + 0.5/f_{SCK}	36.4 μs + 1.5/f_{SCK}
0	0	0	1	0	48.9 μs + 0.5/f_{SCK}	50.7 μs + 1.5/f_{SCK}
0	0	0	1	1	63.1 μs + 0.5/f_{SCK}	64.9 μs + 1.5/f_{SCK}
0	0	1	0	0	77.3 μs + 0.5/f_{SCK}	79.1 μs + 1.5/f_{SCK}
0	0	1	0	1	91.5 μs + 0.5/f_{SCK}	93.3 μs + 1.5/f_{SCK}
0	0	1	1	0	105.8 μs + 0.5/f_{SCK}	107.5 μs + 1.5/f_{SCK}
0	0	1	1	1	120.0 μs + 0.5/f_{SCK}	121.8 μs + 1.5/f_{SCK}
0	1	0	0	0	134.2 μs + 0.5/f_{SCK}	136.0 μs + 1.5/f_{SCK}
0	1	0	0	1	148.4 μs + 0.5/f_{SCK}	150.2 μs + 1.5/f_{SCK}
0	1	0	1	0	162.6 μs + 0.5/f_{SCK}	164.4 μs + 1.5/f_{SCK}
0	1	0	1	1	176.9 μs + 0.5/f_{SCK}	178.6 μs + 1.5/f_{SCK}
0	1	1	0	0	191.1 μs + 0.5/f_{SCK}	192.9 μs + 1.5/f_{SCK}
0	1	1	0	1	205.3 μs + 0.5/f_{SCK}	207.1 μs + 1.5/f_{SCK}
0	1	1	1	0	219.5 μs + 0.5/f_{SCK}	221.3 μs + 1.5/f_{SCK}
0	1	1	1	1	233.7 μs + 0.5/f_{SCK}	235.5 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}.

$$\text{Minimum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}, \quad \text{Maximum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Be sure to set bits 5 and 6 to 0.

3. When controlling the interval time of automatic transmit/receive data transfer by using ADTI, busy control is invalid. (Refer to 13.4.3 (4) (a) Busy control option.)

Remark f_{XX} : System clock frequency (f_X or f_X/2)

f_X : System clock oscillation frequency

f_{SCK} : Serial clock frequency

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
ADTI	ADTI7	0	0	ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	FF6BH	00H	R/W

ADTI4	ADTI3	ADTI2	ADTI1	ADTI0	Data Transfer Interval Specification (f_{XX} = 4.5 MHz Operation)	
					Minimum Note	Maximum Note
1	0	0	0	0	248.0 μs + 0.5/f_{SCK}	250.0 μs + 1.5/f_{SCK}
1	0	0	0	1	262.2 μs + 0.5/f_{SCK}	264.0 μs + 1.5/f_{SCK}
1	0	0	1	0	276.4 μs + 0.5/f_{SCK}	278.2 μs + 1.5/f_{SCK}
1	0	0	1	1	290.6 μs + 0.5/f_{SCK}	292.4 μs + 1.5/f_{SCK}
1	0	1	0	0	304.8 μs + 0.5/f_{SCK}	306.6 μs + 1.5/f_{SCK}
1	0	1	0	1	319.0 μs + 0.5/f_{SCK}	320.8 μs + 1.5/f_{SCK}
1	0	1	1	0	333.2 μs + 0.5/f_{SCK}	335.1 μs + 1.5/f_{SCK}
1	0	1	1	1	347.5 μs + 0.5/f_{SCK}	349.3 μs + 1.5/f_{SCK}
1	1	0	0	0	361.7 μs + 0.5/f_{SCK}	363.5 μs + 1.5/f_{SCK}
1	1	0	0	1	375.9 μs + 0.5/f_{SCK}	377.7 μs + 1.5/f_{SCK}
1	1	0	1	0	390.2 μs + 0.5/f_{SCK}	391.9 μs + 1.5/f_{SCK}
1	1	0	1	1	404.4 μs + 0.5/f_{SCK}	406.2 μs + 1.5/f_{SCK}
1	1	1	0	0	418.6 μs + 0.5/f_{SCK}	420.4 μs + 1.5/f_{SCK}
1	1	1	0	1	432.8 μs + 0.5/f_{SCK}	434.6 μs + 1.5/f_{SCK}
1	1	1	1	0	447.0 μs + 0.5/f_{SCK}	448.8 μs + 1.5/f_{SCK}
1	1	1	1	1	461.3 μs + 0.5/f_{SCK}	463.0 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}.

$$\text{Minimum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{28}{f_{XX}} + \frac{0.5}{f_{SCK}}$$

$$\text{Maximum} = (n+1) \times \frac{2^6}{f_{XX}} + \frac{36}{f_{XX}} + \frac{1.5}{f_{SCK}}$$

Cautions 1. Do not write ADTI during operation of automatic data transmit/receive function.

2. Be sure to set bits 5 and 6 to 0.

3. When controlling the interval time of automatic transmit/receive data transfer by using ADTI, busy control is invalid. (Refer to 13.4.3 (4) (a) Busy control option.)

Remark f_{XX} : System clock frequency (f_X or f_X/2)

f_X : System clock oscillation frequency

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 FADFH 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 bit 7 (CSIE1) and bit 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 specification register (ADTI).
<4> Write any value to the serial I/O shift register 1 (SIO1) (transfer start trigger).

Caution Writing any value to SIO1 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, TRF 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.

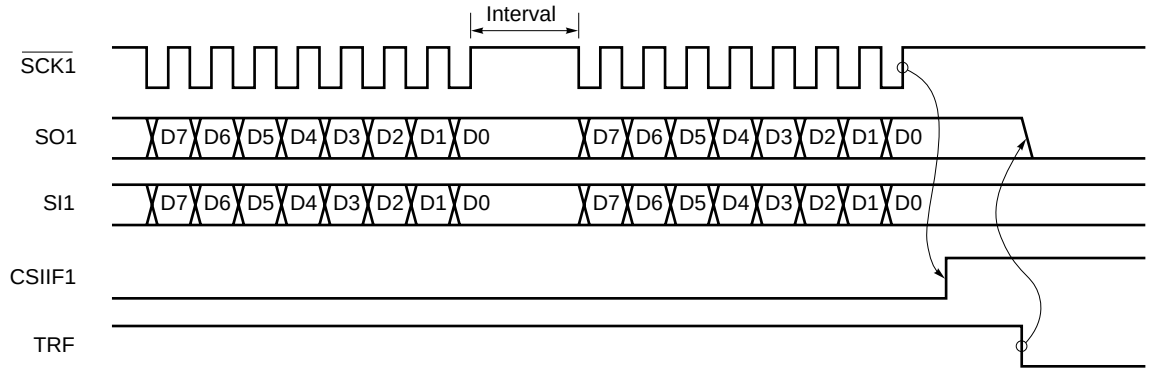
Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. The termination of automatic transmission/reception should be checked by using bit 3 (TRF) of the automatic data transmit/receive control register (ADTC), not by CSIIF1.

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 13-8 shows the basic transmission/reception mode operation timings, and Figure 13-9 shows the operation flowchart.

Figure 13-10 shows buffer RAM operation at 6-byte transmission.

Figure 13-8. Basic Transmission/Reception Mode Operation Timings

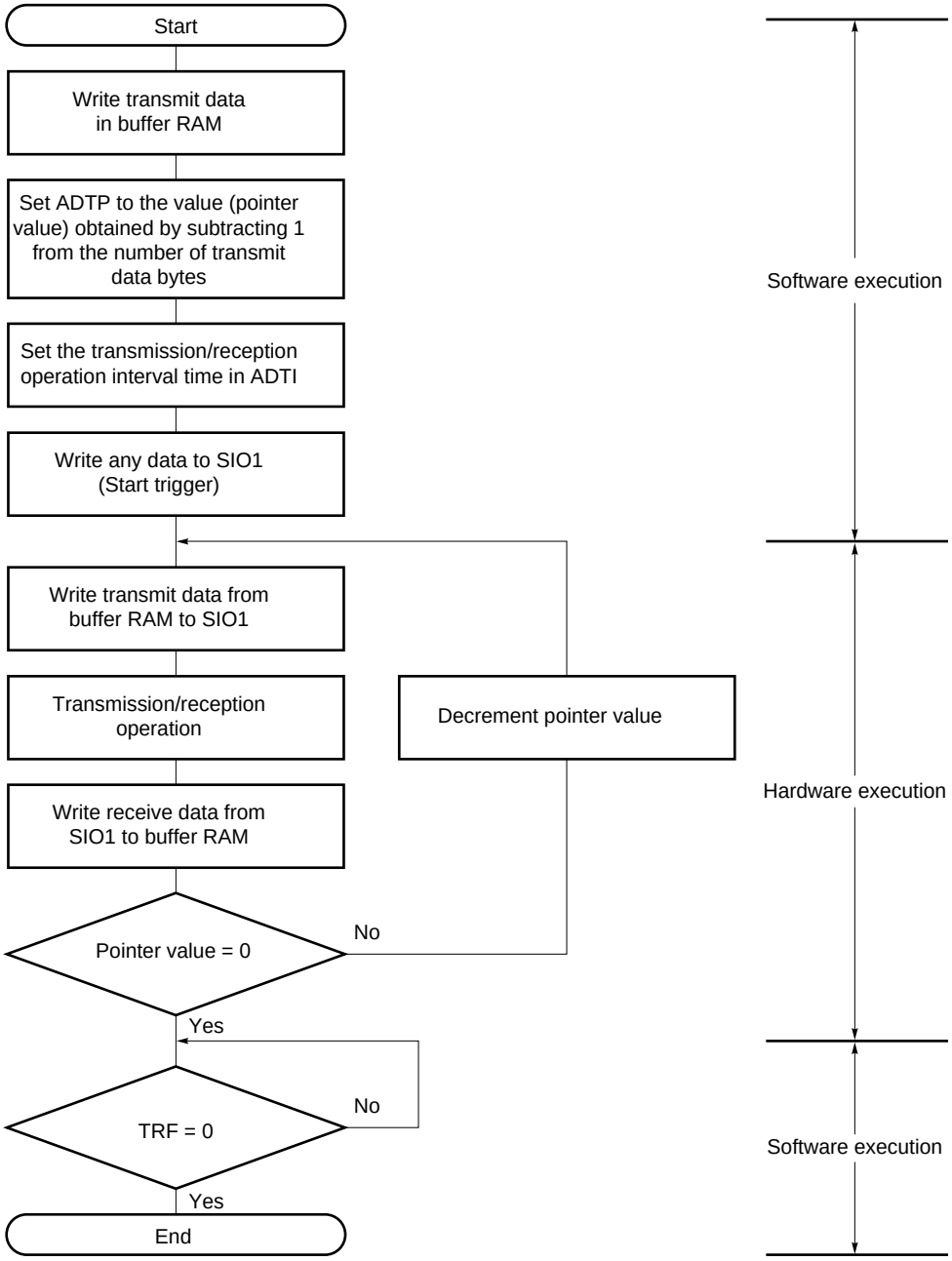


Cautions 1. Because, in the basic transmission/reception mode, the automatic transmit/receive function writes/reads data to/from the buffer RAM after 1-byte transmission/reception, an interval is inserted till the next transmission/reception. As the buffer RAM write/read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specification register (ADTI) (refer to (5) Automatic data transmit/receive interval).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF: Interrupt request flag
TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

Figure 13-9 Basic Transmission/Reception Mode Flowchart



ADTP: Automatic data transmit/receive address pointer
ADTI: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1
TRF: Bit 3 of automatic data transmit/receive control register (ADTC)

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 13-10 (a))

After any data has been written to 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 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 13-10 (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 13-10 (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 13-10. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (1/2)

(a) Before transmission/reception

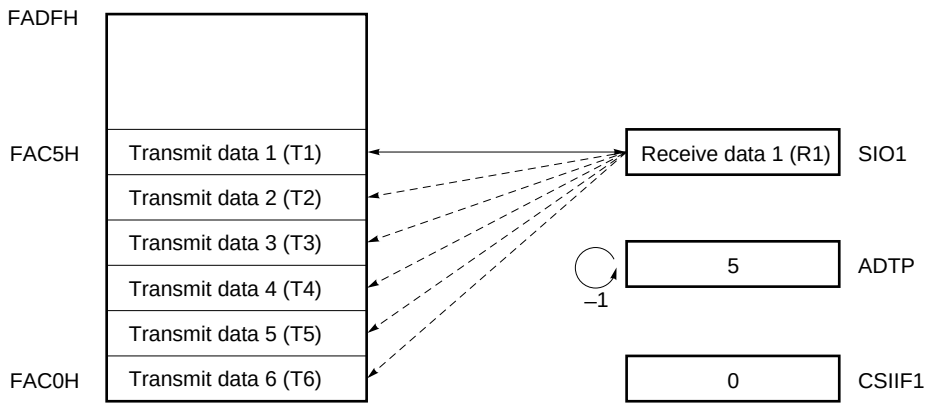
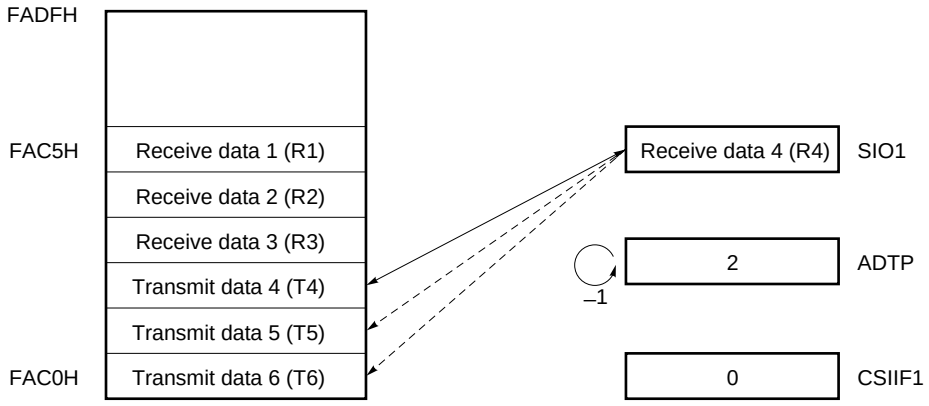
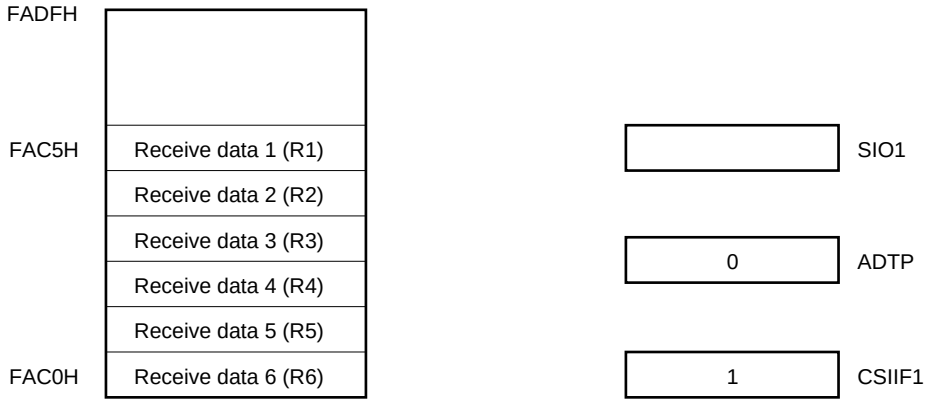


Figure 13-10. Buffer RAM Operation in 6-Byte Transmission/Reception (in Basic Transmit/Receive Mode) (2/2)

(b) 4th byte transmission/reception



(c) Completion of transmission/reception



(b) Basic transmission mode

In this mode, the 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.

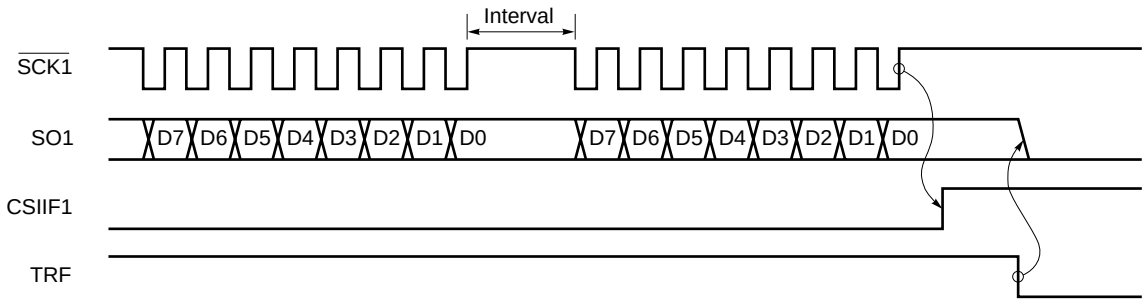
Upon completion of transmission of the last byte, the interrupt request flag (CSIIF1) is set. The termination of automatic transmission/reception should be checked by using bit 3 (TRF) of the automatic data transmit/receive control register (ADTC), not by CSIIF1.

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/ports.

Figure 13-11 shows the basic transmission mode operation timings, and Figure 13-12 shows the operation flowchart.

Figure 13-13 shows buffer RAM operation when repeatedly transmit 6 bytes.

Figure 13-11. Basic Transmission Mode Operation Timings



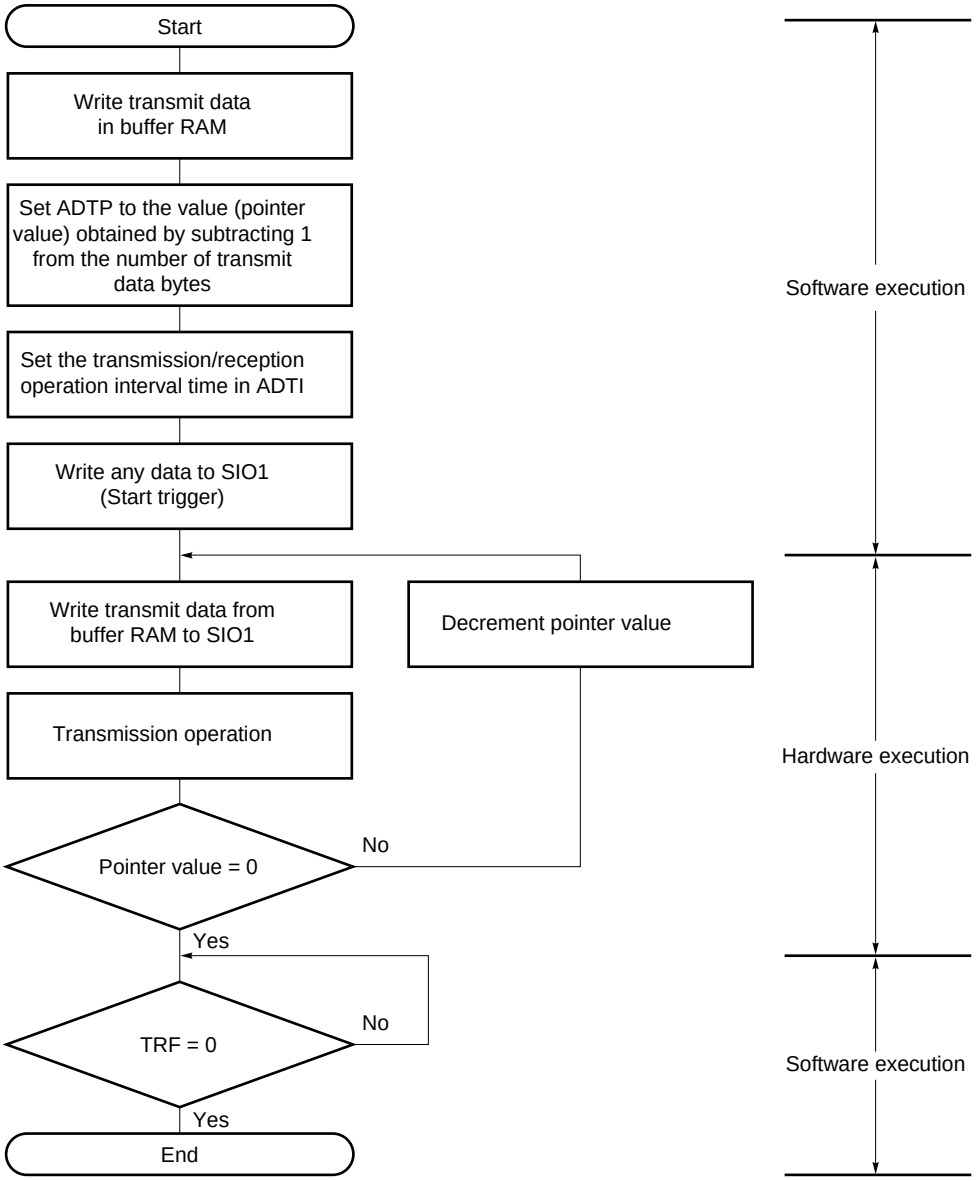
Cautions 1. Because, in the basic transmission mode, the automatic transmit/receive function reads data from the buffer RAM after 1-byte transmission, an interval is inserted till the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the maximum interval is dependent upon CPU processing and the value of the automatic data transmit/receive interval specification register (ADTI) (refer to (5) Automatic data transmit/receive interval).

2. When TRF is cleared, the SO1 pin becomes low level.

Remark CSIIF: Interrupt request flag

TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

Figure 13-12. Basic Transmission Mode Flowchart



ADTP: Automatic data transmit/receive address pointer
ADTI: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1
TRF: Bit 3 of automatic data transmit/receive control register (ADTC)

In 6-byte transmission (ARLD = 0, RE = 0) in basic transmit mode, buffer RAM operates as follows.

(i) Before transmission (refer to Figuree 13-13 (a))

After any data has been written to 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, ADTP is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) 4th byte transmission point (refer to Figure 13-13 (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/reception (refer to Figure 13-13 (c))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is set (INTCSI1 generation).

Figure 13-13. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (1/2)

(a) Before transmission

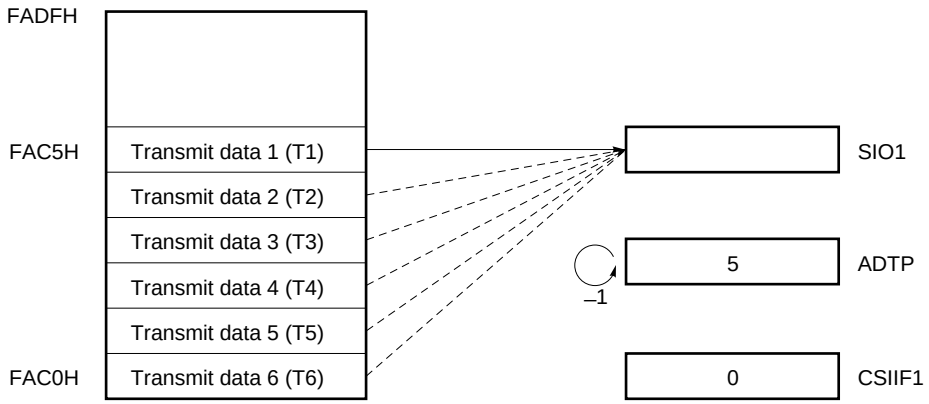
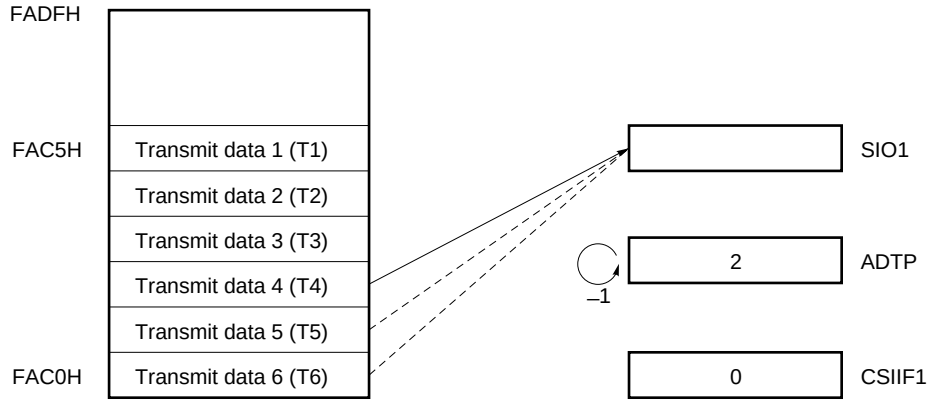
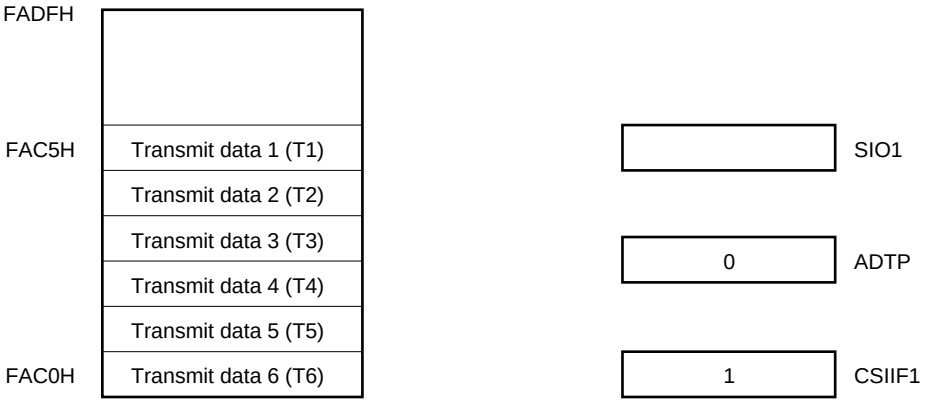


Figure 13-13. Buffer RAM Operation in 6-Byte Transmission (in Basic Transmit Mode) (2/2)

(b) 4th byte transmission point



(c) Completion of transmission/reception



(c) Repeat transmission mode

In this mode, data stored in the buffer RAM is transmitted repeatedly.

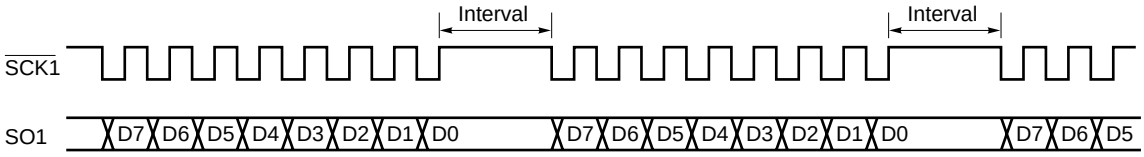
Serial transfer is started by writing any data to serial I/O shift register 1 (SIO1) when 1 is set in bit 7 (CSIE1) of the serial operating mode register 1 (CSIM1).

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 transmission 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 ordinary input/output ports.

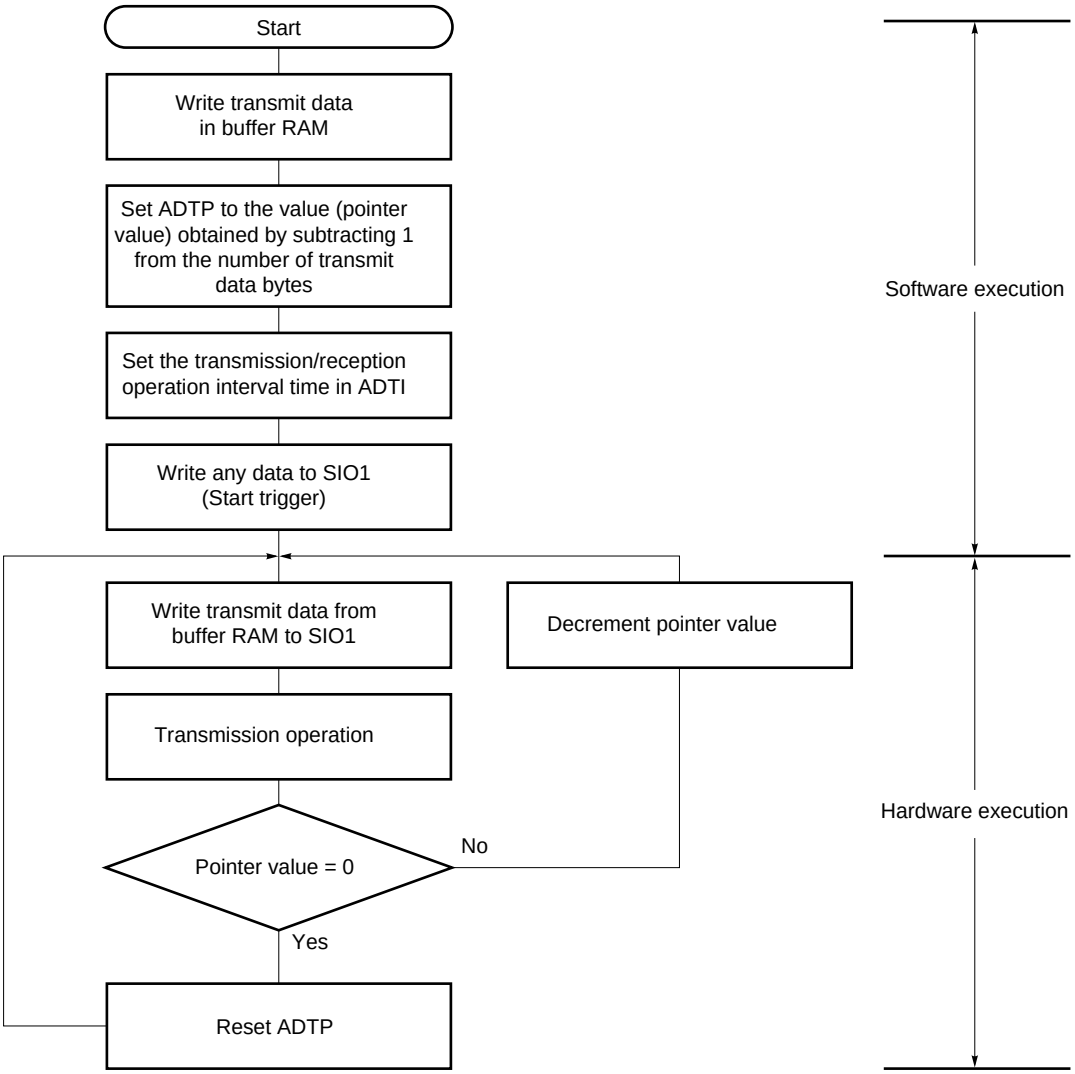
The repeat transmission mode operation timing is shown in Figure 13-14, and the operation flowchart in Figure 13-15.

Figure 13-14. Repeat Transmission Mode Operation Timing



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 specification register (ADTI) (refer to (5) Automatic data transmit/receive interval).

Figure 13-15. Repeat Transmission Mode Flowchart



ADTP: Automatic data transmit/receive address pointer
ADTI: Automatic data transmit/receive interval specification register
SIO1: Serial I/O shift register 1

In 6-byte transmission (ARLD = 1, RE = 0) in repeat transmit mode, buffer RAM operates as follows.

(i) Before transmission (refer to Figure 13-16 (a))

After any data has been written to 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, ADTP is decremented. Then transmit data 2 (T2) is transferred from the buffer RAM to SIO1.

(ii) Upon completion of transmission of 6 bytes (refer to Figure 13-16 (b))

When transmission of the sixth byte is completed, the interrupt request flag (CSIIF1) is not set. The previous pointer value is assigned to the ADTP.

(iii) 7th byte transmission point (refer to Figure 13-16 (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 13-16. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (1/2)

(a) Before transmission

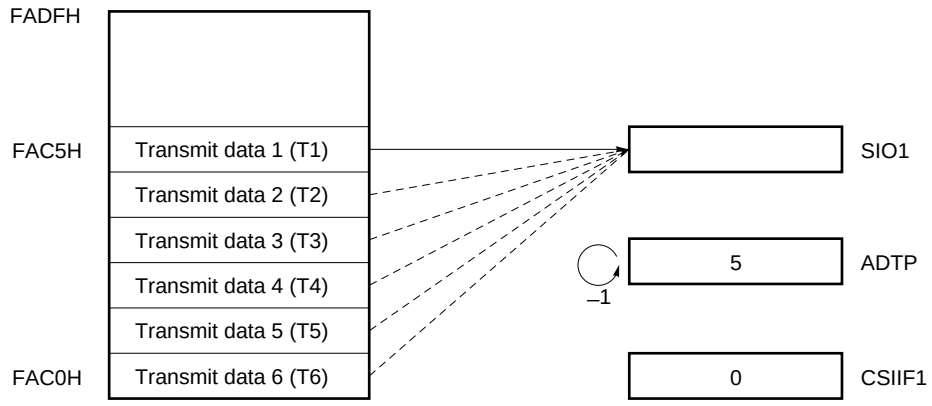
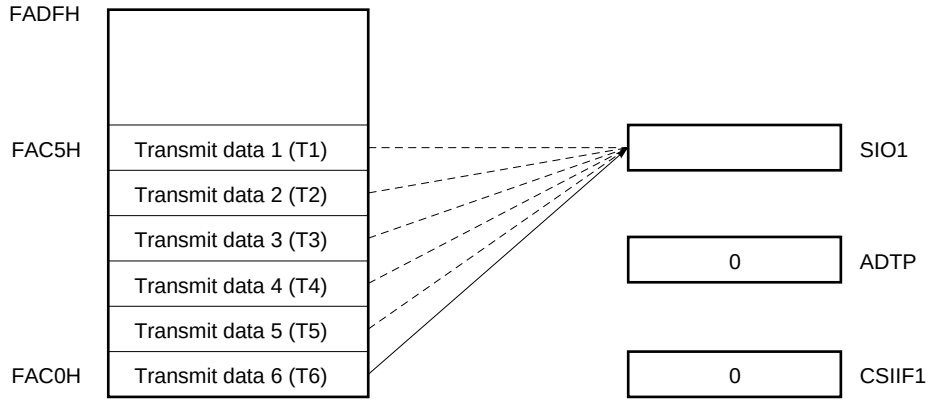
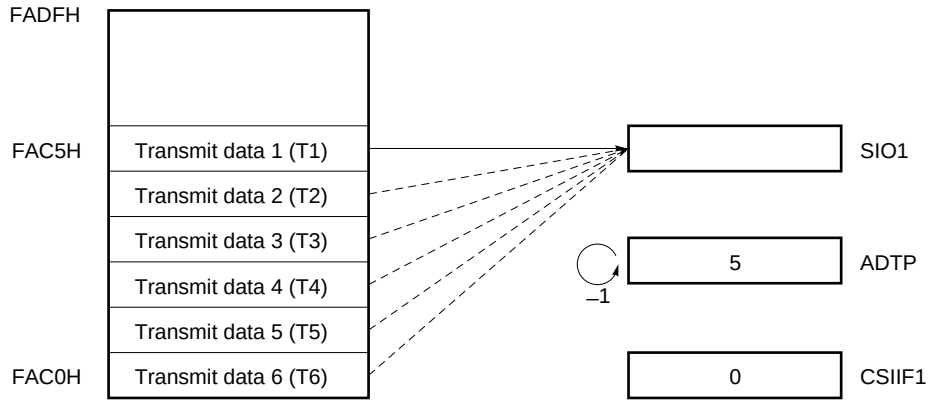


Figure 13-16. Buffer RAM Operation in 6-Byte Transmission (in Repeat Transmit Mode) (2/2)

(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.

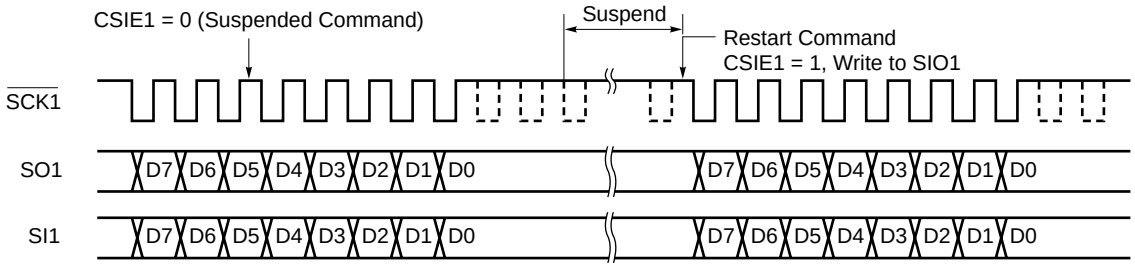
When suspended, bit 3 (TRF) of the automatic data transmit/receive control register (ADTC) is set to 0 after transfer of the 8th bit, and all the port pins used with the serial interface pins for dual function (P20/SI1, P21/SO1, P22/$\overline{\text{SCK1}}$, P23/STB and P24/BUSY) are set to the port mode.

During restart of transmission/reception, 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 from 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 13-17. Automatic Transmission/Reception Suspension and Restart



CSIE1: Bit 7 of serial operating mode register 1 (CSIM1)

(4) Synchronization control

Busy control and strobe control are functions to synchronize transmission/reception between the master device and a slave device.

By using these functions, a shift in bits being transmitted or received can be detected.

(a) Busy control option

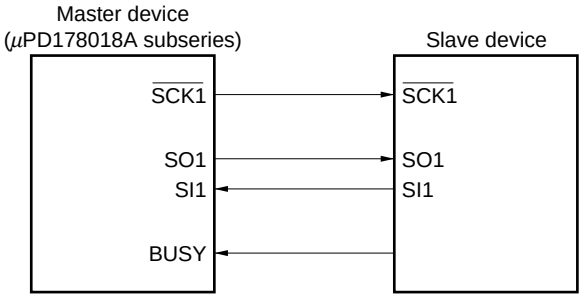
Busy control is a function to keep the serial transmission/reception by the master device waiting while the busy signal output by a slave device to the master is active.

When using this busy control option, the following conditions must be satisfied.

- Bit 5 (ATE) of the serial operating mode register 1 (CSIM1) is set to 1.
- Bit 1 (BUSY1) of the automatic data transmit/receive control register (ADTC) is set to 1.

Figure 13-18 shows the system configuration of the master device and a slave device when the busy control option is used.

Figure 13-18. System Configuration when Busy Control Option Is Used



The master device inputs the busy signal output by the slave device to the BUSY/P24 pin. The master device samples the input busy signal in synchronization with the falling of the serial clock. Even if the busy signal becomes active while 8-bit data is being transmitted or received, transmission/reception by the master is not kept waiting. If the busy signal is active at the rising edge of the serial clock 2 clocks after completion of transmission/reception of the 8-bit data, the busy input becomes valid. After that, the master transmission/reception is kept waiting while the busy signal is active.

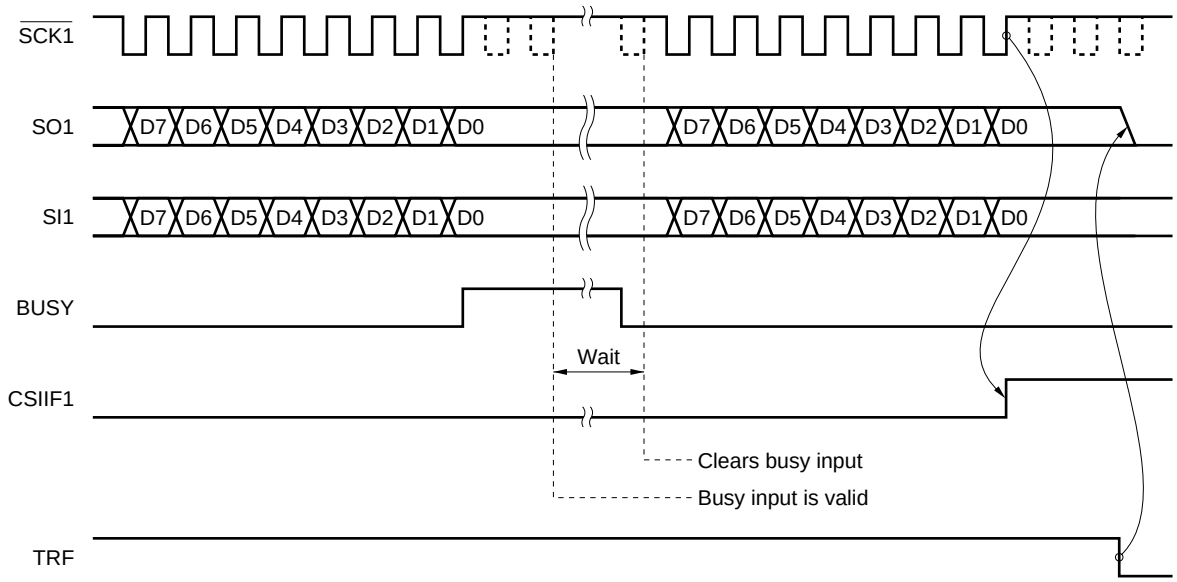
The active level of the busy signal is set by bit 0 (BUSY0) of ADTC.

BUSY0 = 0: Active high
BUSY1 = 1: Active low

When using the busy control option, select the internal clock as the serial clock. Control with the busy signal cannot be implemented with the external clock.
Figure 13-19 shows the operation timing when the busy control option is used.

Caution Busy control cannot be used simultaneously with the interval time control function of the automatic data transmit/receive interval specification register (ADTI). If used, busy control is invalid.

Figure 13-19. Operation Timing When Busy Control Option Is Used (when BUSY0 = 0)

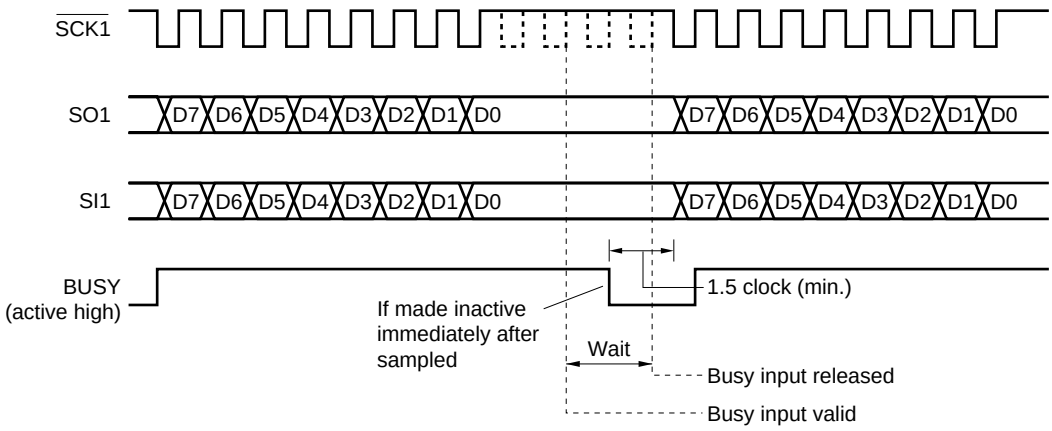


Caution If the TRF is cleared, the SO1 pin goes low.

Remark CSIIF1: Interrupt request flag
TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

When the busy signal becomes inactive, waiting is released. If the sampled busy signal is inactive, transmission/reception of the next 8-bit data is started at the falling edge of the next clock.
Because the busy signal is asynchronous with the serial clock, it takes up to 1 clock until the busy signal, even if made inactive by the slave, is sampled. It takes 0.5 clock until data transfer is started after the busy signal was sampled.
To accurately release waiting, the slave must keep the busy signal inactive at least for the duration of 1.5 clock.
Figure 13-20 shows the timing of the busy signal and releasing the waiting. This figure shows an example where the busy signal is active as soon as transmission/reception has been started.

Figure 13-20. Busy Signal and Wait Release (when BUSY0 = 0)



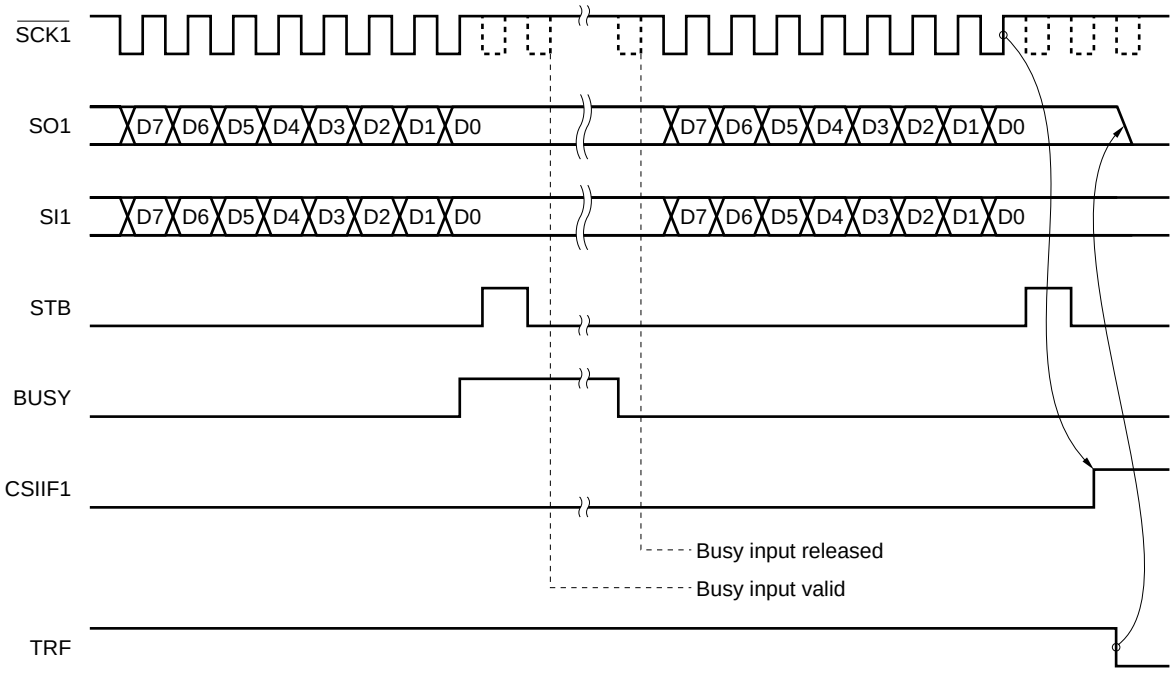
(b) Busy & strobe control option

Strobe control is a function to synchronize data transmission/reception between the master and slave devices. The master device outputs the strobe signal from the STB/P23 pin when 8-bit transmission/reception has been completed. By this signal, the slave device can determine the timing of the end of data transmission. Therefore, synchronization is established even if a bit shift occurs because noise is superimposed on the serial clock, and transmission of the next byte is not affected by the bit shift.
To use the strobe control option, the following conditions must be satisfied:

- Bit 5 (ATE) of the serial operating mode register 1 (CSIM1) is set to 1.
- Bit 2 (STRB) of the automatic data transmit/receive control register (ADTC) is set to 1.

Usually, the busy control and strobe control options are simultaneously used as handshake signals. In this case, the strobe signal is output from the STB/P23 pin, and the BUSY/P24 pin is sampled, and transmission/reception can be kept waiting while the busy signal is input.
When the strobe control option is not used, the P23/STB pin can be used as a normal I/O port pin.
Figure 13-21 shows the operation timing when the busy & strobe control options are used.
When the strobe control option is used, the interrupt request flag (CSIIF1) that is set on completion of transmission/reception is set after the strobe signal is output.

Figure 13-21. Operation Timing when Busy & Strobe Control Options Are Used (when BUSY0 = 0)



Caution When TRF is cleared, the SO1 pin goes low.

Remark CSIIF1: Interrupt request flag
TRF : Bit 3 of automatic data transmit/receive control register (ADTC)

(c) Bit shift detection by busy signal

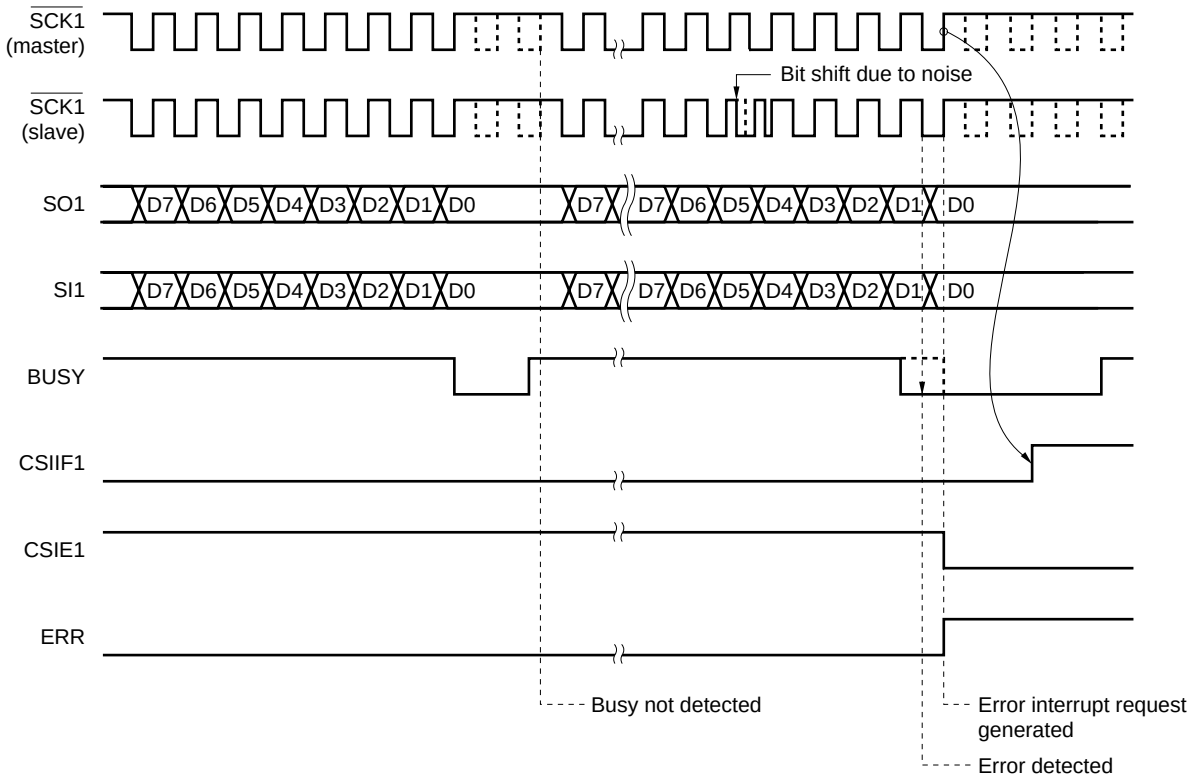
During automatic transmission/reception, a bit shift of the serial clock of the slave device may occur because noise is superimposed on the serial clock signal output by the master device. Unless the strobe control option is used at this time, the bit shift affects transmission of the next byte. In this case, the master can detect the bit shift by checking the busy signal during transmission by using the busy control option. A bit shift is detected by using the busy signal as follows:

The slave outputs the busy signal after the rising of the eighth serial clock during data transmission/reception (to not keep transmission/reception waiting by the busy sigpLL at this time, make the busy signal inactive within 2 clocks).

The master samples the busy signal in synchronization of the falling of the leading side of the serial clock. If a bit shift does not occur, all the eight serial clocks that have been sampled are inactive. If the sampled serial clocks are active, it is assumed that a bit shift has occurred, and error processing is executed (by setting bit 4 (ERR) of the automatic transmit/receive control register (ADTC) to 1).

Figure 13-22 shows the operation timing of the bit shift detection function by the busy signal.

Figure 13-22. Operation Timing of Bit Shift Detection Function by Busy Signal (when BUSY0 = 1)



CSIIF1 : Interrupt request flag
CSIE1 : Bit 7 of serial operating mode register1 (CSIM1)
ERR : Bit 4 of automatic data transmit/receive control register (ADTC)

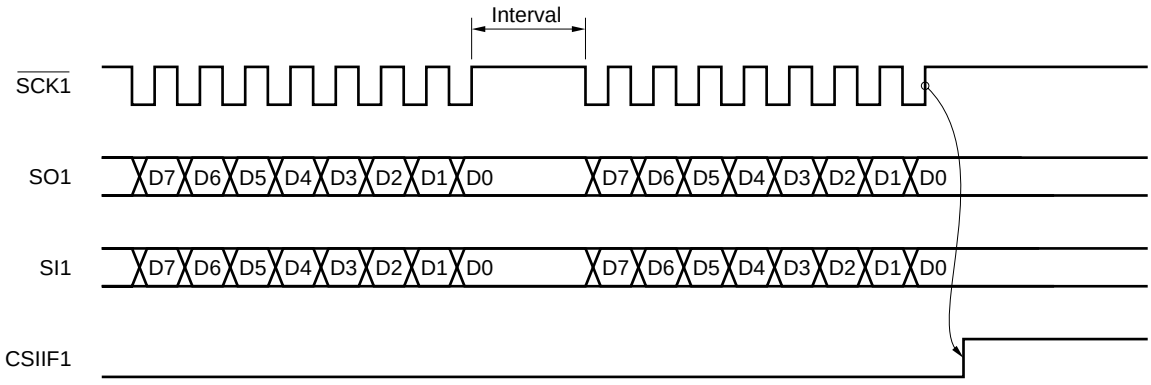
(5) Interval time of automatic transmission/reception

When using the automatic transmit/receive function, an interval time elapses after 1 byte has been transmitted or received, until the next transmission/reception is executed because data is written to or read from the buffer RAM.

To use the automatic transmit/receive function with the internal clock, the interval time is dependent on the CPU processing of the timing of the eighth rising of the serial clock and the set value of the automatic data transmit/receive interval specification register (ADTI). Whether the interval time is dependent on ADIT is selected by setting of the bit 7 (ADTI7) of ADTI. If ADTI7 is reset to 0, the interval time is dependent on the CPU processing only. If ADTI7 is set to 1, the interval time determined by the set contents of ADTI or interval time by the CPU processing is selected whichever greater.

To use the automatic transmit/receive function with the external clock, the external clock must be input in the manner that the interval time is equal to or greater than the time described in (b) below.

Figure 13-23. Interval Time of Automatic Transmission/Reception



CSIIF1: Interrupt request flag

(a) When using automatic transmit/receive function with internal clock

The internal clock is used when bit 1 (CSIM11) of the serial operating mode register 1 (CSIM1) is set to 1.

To use the automatic transmit/receive function with the internal clock, the interval time by the CPU processing is as follows:

When bit 7 (ADTI7) of the automatic data transmit/receive interval specification register (ADTI) is reset to 0, the interval time is that by the CPU processing. When ADTI7 is set to 1, the interval time is that determined by the set contents of ADTI or that by the CPU processing, whichever greater.

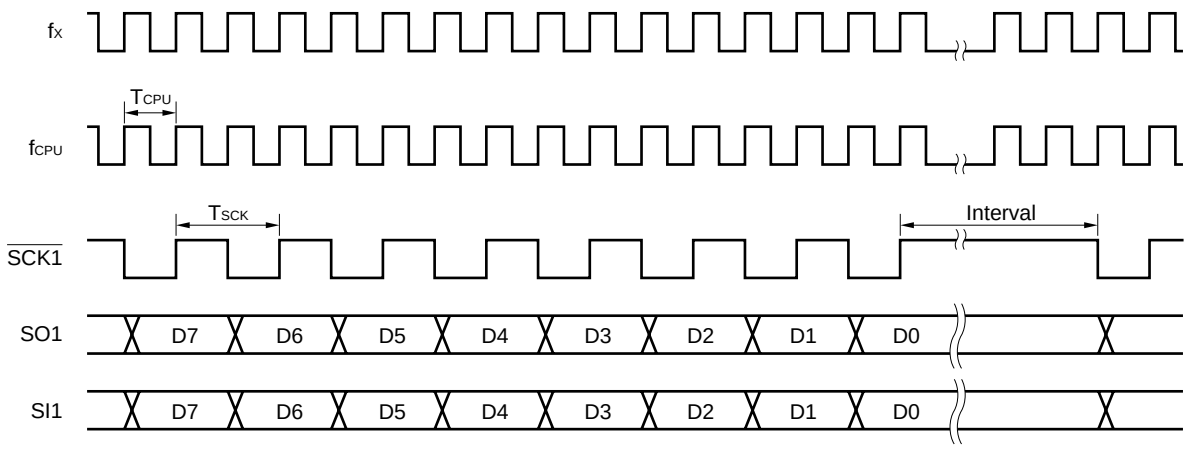
For the interval time by ADTI, refer to **Figure 13-5 Format of Automatic Data Transmit/Receive Interval Time Specification Register**.

Table 13-2. Interval Time by CPU Processing (with internal clock)

CPU Processing	Interval Time
With multiplication instruction used	MAX. (2.5 T_{SCK}, 13T_{CPU})
With division instruction used	MAX. (2.5 T_{SCK}, 20T_{CPU})
External access 1 wait mode	MAX. (2.5 T_{SCK}, 9T_{CPU})
Others	MAX. (2.5 T_{SCK}, 7T_{CPU})

T_{SCK} : $1/f_{SCK}$
f_{SCK} : Serial clock frequency
T_{CPU} : $1/f_{CPU}$
f_{CPU} : CPU clock (set by bits 0 through 2 (PCC0 through PCC2) of processor clock control register (PCC))
MAX. (a, b) : Value of a or b whichever greater

Figure 13-24. Operation Timing when Automatic Transmit/Receive Function Is Used with Internal Clock



f_X : System clock oscillation frequency
f_{CPU} : CPU clock (set by bits 0 through 2 (PCC0 through PCC2) of processor clock control register (PCC))
T_{CPU} : $1/f_{CPU}$
T_{SCK} : $1/f_{SCK}$
f_{SCK} : Serial clock frequency

(b) When using automatic transmit/receive function with external clock

The external clock is used when bit 1 (CSIM11) of the serial operation mode register 1 (CSIM1) is cleared to 0.

To use the automatic transmit/receive function with the external clock, the external clock must be input such that the interval time is as follows:

Table 13-3. Interval Time by CPU Processing (with external clock)

CPU Processing	Interval Time
With multiplication instruction used	$13T_{CPU}$ MIN.
With division instruction used	$20T_{CPU}$ MIN.
External access 1 wait mode	$9T_{CPU}$ MIN.
Others	$7T_{CPU}$ MIN.

T_{CPU} : $1/f_{CPU}$
f_{CPU} : CPU clock (set by bits 0 through 2 (PCC0 through PCC2) of processor clock control register (PCC))

CHAPTER 14 INTERRUPT AND TEST FUNCTIONS

14.1 Interrupt Function Types

The following three types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally even if interrupt is disabled. 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 specification flag register (PR0L, PR0L).
Multiple high priority interrupts can be applied to low priority interrupts. If two or more interrupts with the same priority are simultaneously generated, each interrupts has a predetermined priority (refer to **Table 14-1**).
A standby release signal is generated.
Seven external interrupt sources and fifteen 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 a disabled state. The software interrupt does not undergo interrupt priority control.

14.2 Interrupt Sources and Configuration

A total of 17 non-maskable, maskable, and software interrupt sources are incorporated in the interrupt sources (refer to **Table 14-1**).

Table 14-1. Interrupt Source List

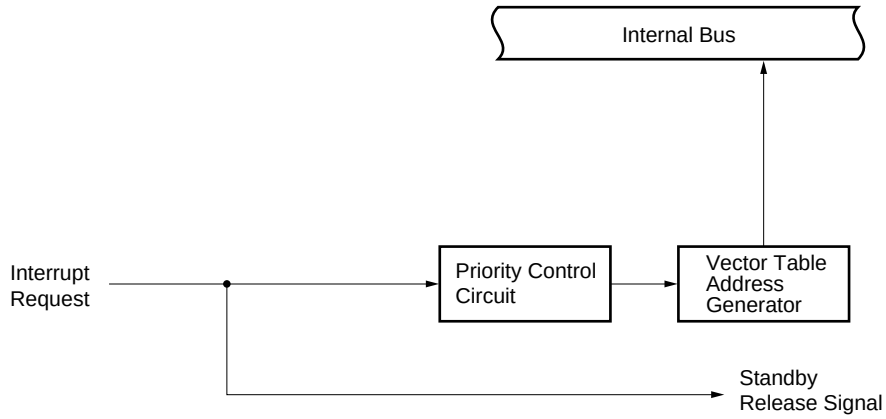
Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Non-maskable	–	INTWDT	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0004H	(A)
Maskable	0	INTWDT	Watchdog timer overflow (with interval timer mode selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	(D)
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTP6			0012H	
	8	INTCSI0	End of serial interface channel 0 transfer	Internal	0014H	(B)
	9	INTCSI1	End of serial interface channel 1 transfer		0016H	
	10	INTTMC	Generation of basic timer match signal		0018H	
	11	INTPWM	Generation of 8-bit timer match signal		001AH	
	12	INTTM1	Generation of 8-bit timer/event counter 1 match signal		001CH	
	13	INTTM2	Generation of 8 bit timer/event counter 2 match signal		001EH	
	14	INTAD	End of A/D converter conversion		0020H	
Software	–	BRK	BRK instruction execution	Internal	003EH	(E)

Notes 1. Default priorities are intended for two or more simultaneously generated maskable interrupt requests. 0 is the highest priority and 14 is the lowest priority.

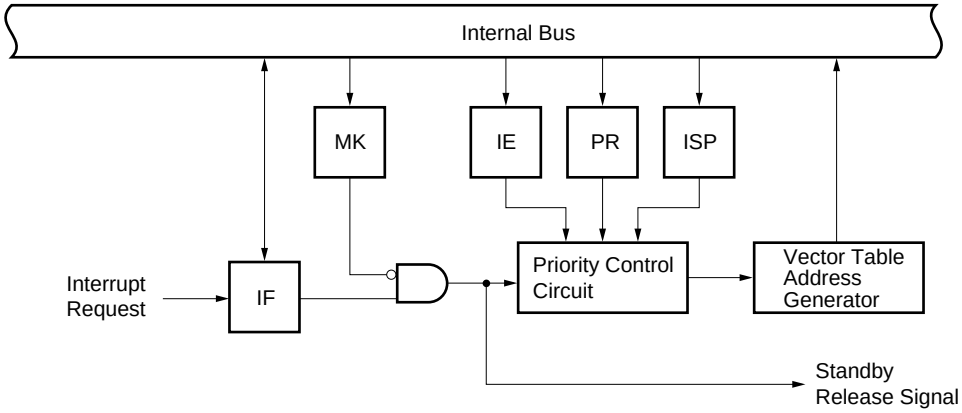
2. Basic configuration types (A) to (E) correspond to (A) to (E) of Figure 14-1.

Figure 14-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal non-maskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0)

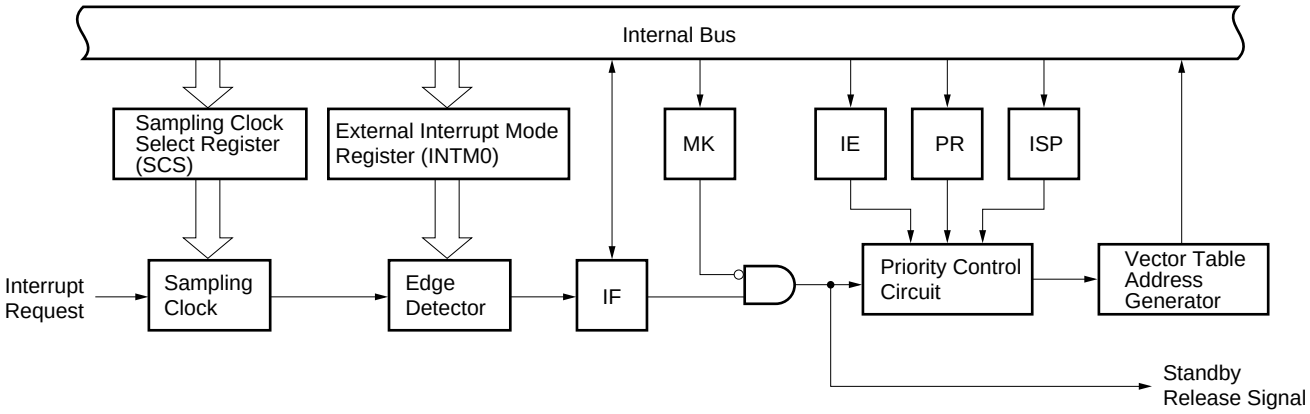
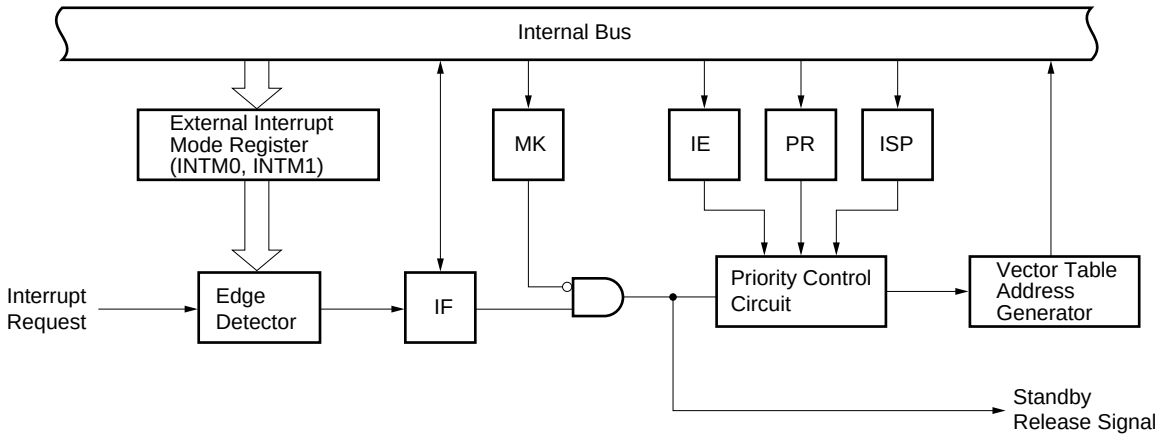
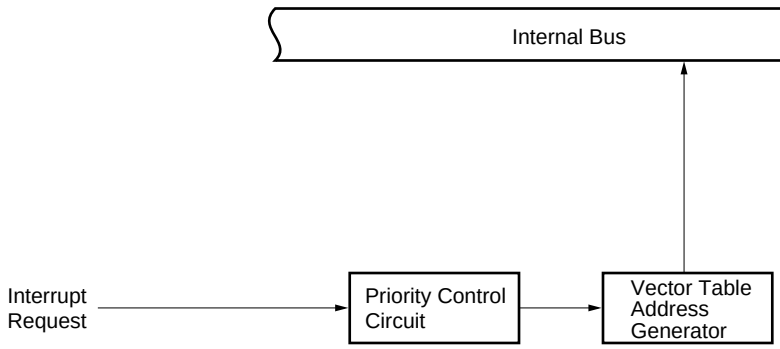


Figure 14-1. Basic Configuration of Interrupt Function (2/2)

(D) External maskable interrupt (except INTP0)



(E) Software interrupt



Remark IF : Interrupt request flag
IE : Interrupt enable flag
ISP : Inservice priority flag
MK : Interrupt mask flag
PR : Priority specification flag

14.3 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 specification flag register (PR0L, PR0H)
- External interrupt mode register (INTM0, INTM1)
- Sampling clock select register (SCS)
- Program status word (PSW)

Table 14-2 gives a listing of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 14-2. Various Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTP0	PIF0	IF0L	PMK0	MK0L	PPR0	PR0L
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTP6	PIF6		PMK6		PPR6	
INTWDT	TMIF4		TMMK4		TMPR4	
INTTM1	TMIF1	IF0H	TMMK1	MK0H	TMPR1	PR0H
INTTM2	TMIF2		TMMK2		TMPR2	
INTCSI0	CSIIF0		CSIMK0		CSIPR0	
INTCSI1	CSIIF1		CSIMK1		CSIPR1	
INTAD	ADIF		ADMK		ADPR	
INTTMC	TMCIF		TMCMK		TMCPR	
INTPWM	PWMIF		PWMMK		PWMPR	

(1) Interrupt request flag registers (IF0L and 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 reset input.

IF0L and IF0H are set with a 1-bit or 8-bit memory manipulation instruction. If IF0L and IF0H are used as a 16-bit register IF0 use a 16-bit memory manipulation instruction for the setting.

Reset input sets these registers to 00H.

Figure 14-2. Interrupt Request Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
IF0L	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W
	7	⑥	⑤	④	③	②	①	⓪			
IF0H	0	ADIF	TMIF2	TMIF1	PWMIF	TMCIF	CSIIF1	CSIIF0	FFE1H	00H	R/W

××IF×	Interrupt Request Flag
0	No interrupt request signal
1	Interrupt request signal is generated; Interrupt request state

Cautions 1. TMIF4 flag is R/W enabled only when a watchdog timer is used as an interval timer. If a watchdog timer is used in watchdog timer mode 1, set TMIF4 flag to 0.

2. Be sure to set bit 7 of IF0H to 0.

(2) Interrupt mask flag registers (MK0L and MK0H)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt service and to set standby clear enable/disable.

MK0L and MK0H are set with a 1-bit or 8-bit memory manipulation instruction. If MK0L and MK0H are used as a 16-bit register MK0, use a 16-bit memory manipulation instruction for the setting.

Reset input sets these registers to FFH.

Figure 14-3. Interrupt Mask Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
MK0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK	PMK	TMMK4	FFE4H	FFH	R/W
	7	⑥	⑤	④	③	②	①	⓪			
MK0H	1	ADMK	TMMK2	TMMK1	PWMMK	TMCMK	CSIMK1	CSIMK0	FFE5H	FFH	R/W

××MK×	Interrupt Service Control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If TMMK4 flag is read when a watchdog timer is used in watchdog timer mode 1, MK0 value becomes undefined.

2. Because port 0 has a dual 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 bit 7 of MK0H to 1.

(3) Priority specification flag registers (PR0L and PR0H)

The priority specification flag is used to set the corresponding maskable interrupt priority orders.

PR0L and PR0H are set with a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are used as a 16-bit register PR0, use a 16-bit memory manipulation instruction for the setting.

Reset input sets these registers to FFH.

Figure 14-4. Priority Specification Flag Register Format

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	After Reset	R/W
PR0L	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	TMPR4	FFE8H	FFH	R/W
	7	⑥	⑤	④	③	②	①	⓪			
PR0H	1	ADPR	TMPR2	TMPR1	PWMPR	TMCPR	CSIPR1	CSIPR0	FFE9H	FFH	R/W

××PR×	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 TMPR4 flag to 1.

2. Be sure to set bit 7 of PR0H to 1.

(4) External interrupt mode register (INTM0, INTM1)

These registers set the valid edge for INTP0 to INTP6.

INTM0 and INTM1 are set by 8-bit memory manipulation instructions.

Reset input sets these registers to 00H.

Figure 14-5. External Interrupt Mode Register 0 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

Figure 14-6. External Interrupt Mode Register 1 Format

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
INTM1	ES71	ES70	ES61	ES60	ES51	ES50	ES41	ES40	FFEDH	00H	R/W

ES41	ES40	INTP3 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES51	ES50	INTP4 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES61	ES60	INTP5 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES71	ES70	INTP6 Valid Edge Selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

(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.

Reset input sets SCS to 00H.

Figure 14-7. 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		
			MCS = 1	MCS = 0
0	0	$f_{XX}/2^N$		
0	1	$f_{XX}/2^7$	$f_X/2^7$(35.2 kHz)	$f_X/2^8$(17.6 kHz)
1	0	$f_{XX}/2^5$	$f_X/2^5$(140.6 kHz)	$f_X/2^6$(70.3 kHz)
1	1	$f_{XX}/2^6$	$f_X/2^6$(70.3 kHz)	$f_X/2^7$(35.2 kHz)

Caution $f_{XX}/2^N$ is a clock to be supplied to the CPU and $f_{XX}/2^5$, $f_{XX}/2^6$ and $f_{XX}/2^7$ are clocks to be supplied to the peripheral hardware. $f_{XX}/2^N$ stops in the HALT mode.

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

2. f_{XX} : System clock frequency (f_X or $f_X/2$)

3. f_X : System clock oscillation frequency

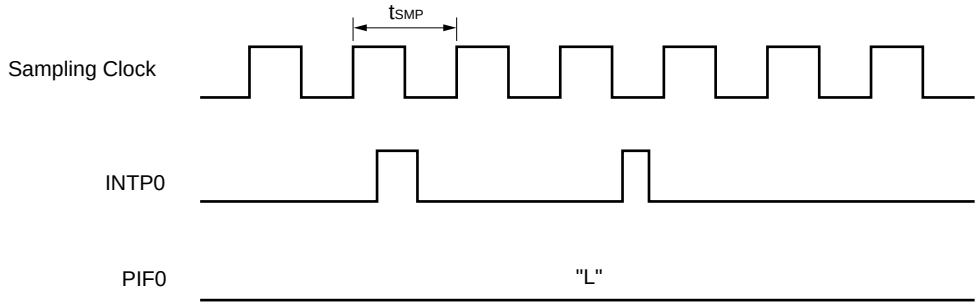
4. MCS : Oscillation mode selection register (OSMS) bit 0

5. Values in parentheses when operated with f_X = 4.5 MHz.

When the sampled INTP0 input level is active twice in succession, the noise eliminator sets interrupt request flag (PIF0) to 1.

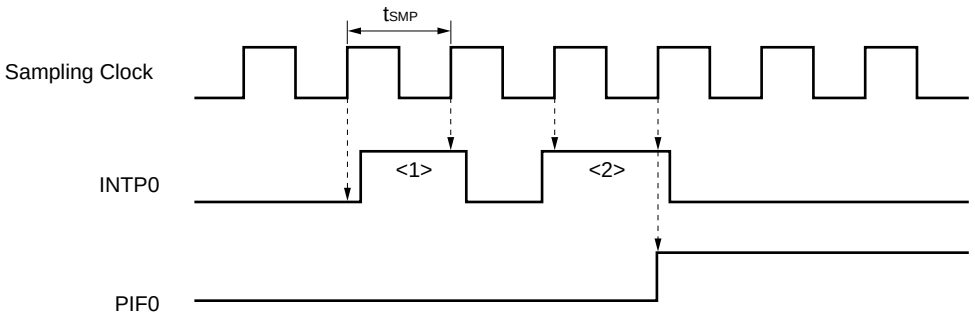
Figure 14-8. Noise Eliminator Input/Output Timing (during rising edge detection)

(a) When input is less than the sampling cycle (t_{SMP})



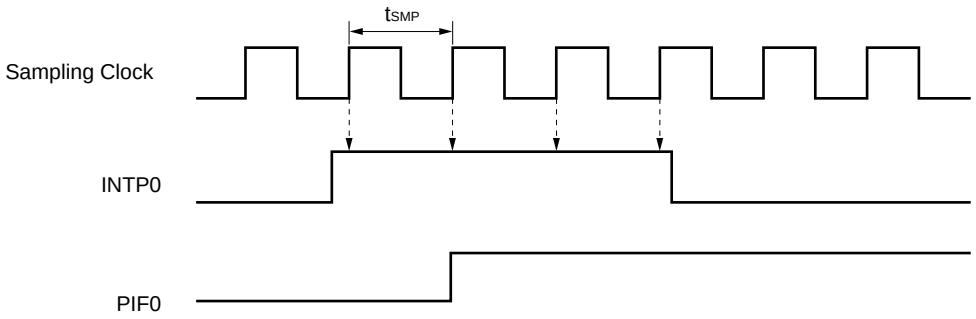
Because INTP0 level is not high when sampled,
PIF0 output remains at low level.

(b) When input is equal to or twice the sampling cycle (t_{SMP})



Because the sampled INTP0 level is high twice in succession in <2>,
PIF0 flag is set to 1.

(c) When input is twice or more than the cycle frequency (t_{SMP})



At the point when INTP0 level is high two times in succession,
PIF0 flag is set to 1.

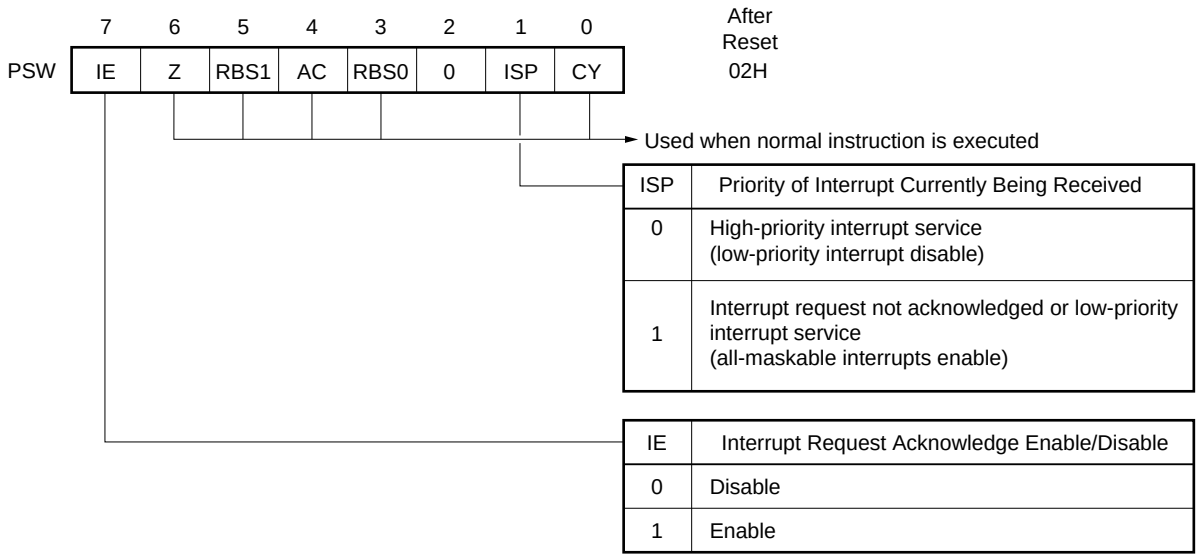
(6) Program status word (PSW)

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 nesting interrupt 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, if the BRK instruction is executed, the contents of PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The acknowledged interrupt is also saved into the stack with the PUSH PSW instruction. It is restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset input sets PSW to 02H.

Figure 14-9. Program Status Word Configuration



ISP	Priority of Interrupt Currently Being Received
0	High-priority interrupt service (low-priority interrupt disable)
1	Interrupt request not acknowledged or low-priority interrupt service (all-maskable interrupts enable)

IE	Interrupt Request Acknowledge Enable/Disable
0	Disable
1	Enable

14.4 Interrupt Service Operations

14.4.1 Non-maskable interrupt request acknowledge operation

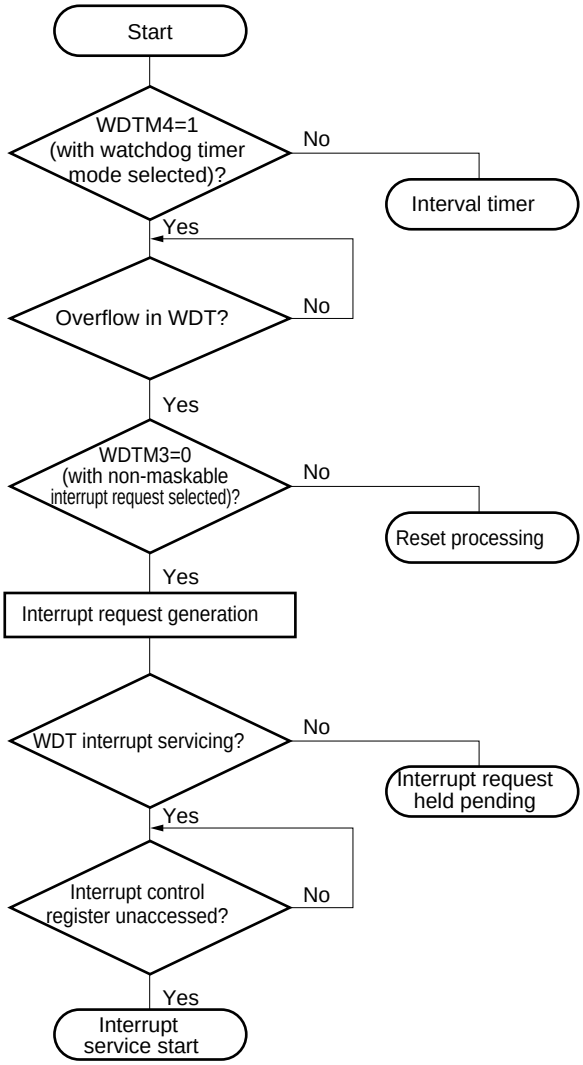
A non-maskable interrupt request is unconditionally acknowledged even if in an interrupt request acknowledge disable state. It does not undergo interrupt priority control and has highest priority over all other interrupts.

If a non-maskable interrupt request is acknowledged, the acknowledged interrupt is saved in the stacks, the program status word (PSW) and the program counter (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.

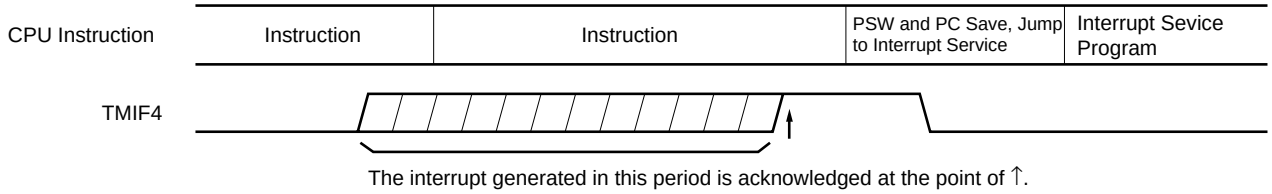
Figure 14-10 shows the flowchart from generation of the non-maskable interrupt request to acknowledging it. Figure 14-11 shows the timing of acknowledging the non-maskable interrupt request, and Figure 14-12 shows the operation performed if a more than one non-maskable interrupt request occurs.

Figure 14-10. Flowchart from Generation of Non-Maskable Interrupt Request to Acknowledge



WDTM : Watchdog timer mode register
WDT : Watchdog timer

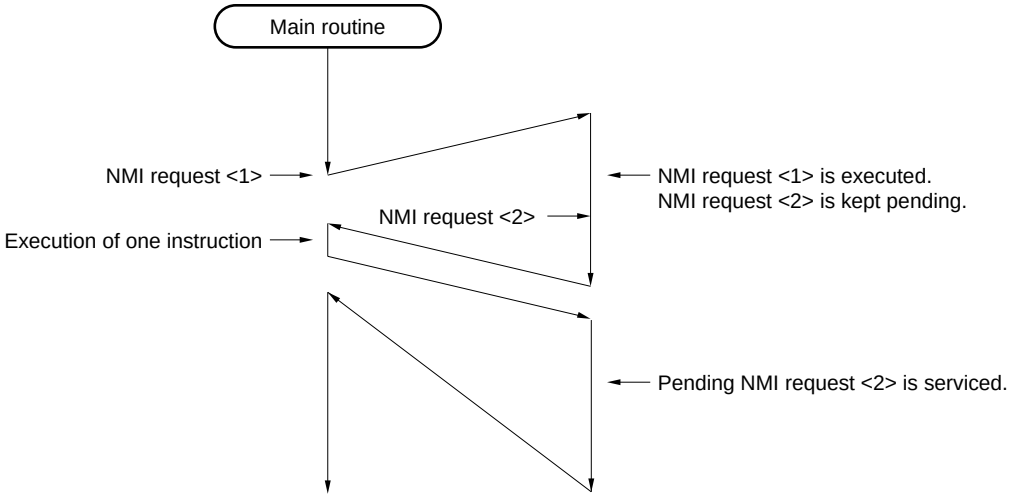
Figure 14-11. Non-Maskable Interrupt Request Acknowledge Timing



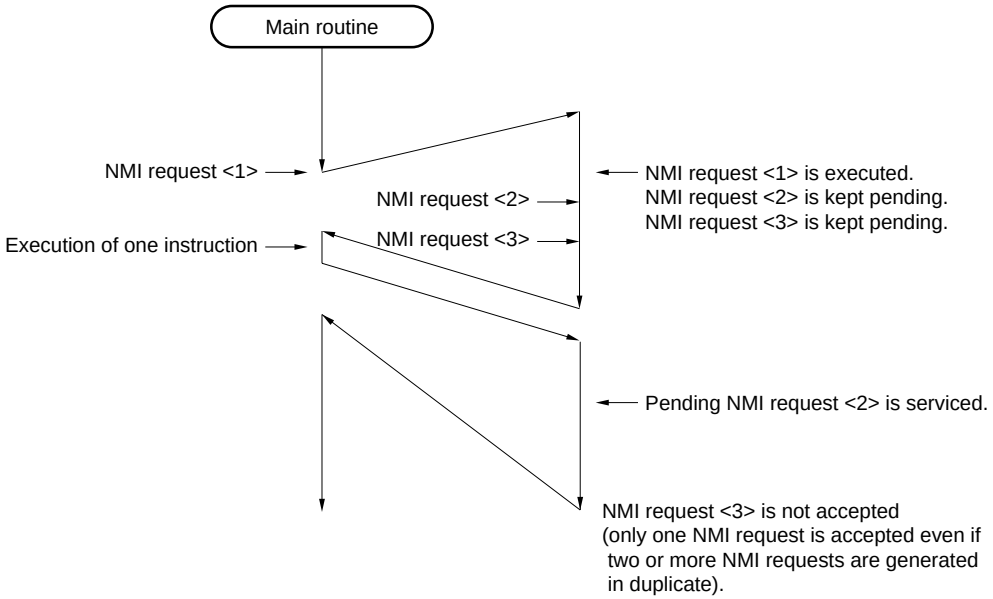
TMIF4 : Watchdog timer interrupt request flag

Figure 14-12. Non-Maskable Interrupt Request Acknowledge Operation

(a) If a new non-maskable interrupt request is generated during non-maskable interrupt service program execution



(b) If two non-maskable interrupt requests are generated during non-maskable interrupt service program execution



14.4.2 Maskable interrupt request acknowledge operation

A maskable interrupt request becomes acknowledgeable when an interrupt request flag is set to 1 and the mask (MK) flag of the interrupt request 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 request service (with ISP flag reset to 0).

Wait times maskable interrupt request generation to interrupt request service are as follows.

Table 14-3. Times from Maskable Interrupt Request Generation to Interrupt Service

	Minimum Time	Maximum Time **Note**
When ××PR× = 0	7 clocks	32 clocks
When ××PR× = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time is maximized.

Remark 1 clock : $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request specified for higher priority with the priority specification flag is acknowledged first. Two or more requests specified for the same priority by the priority specification flag, the default priorities apply.

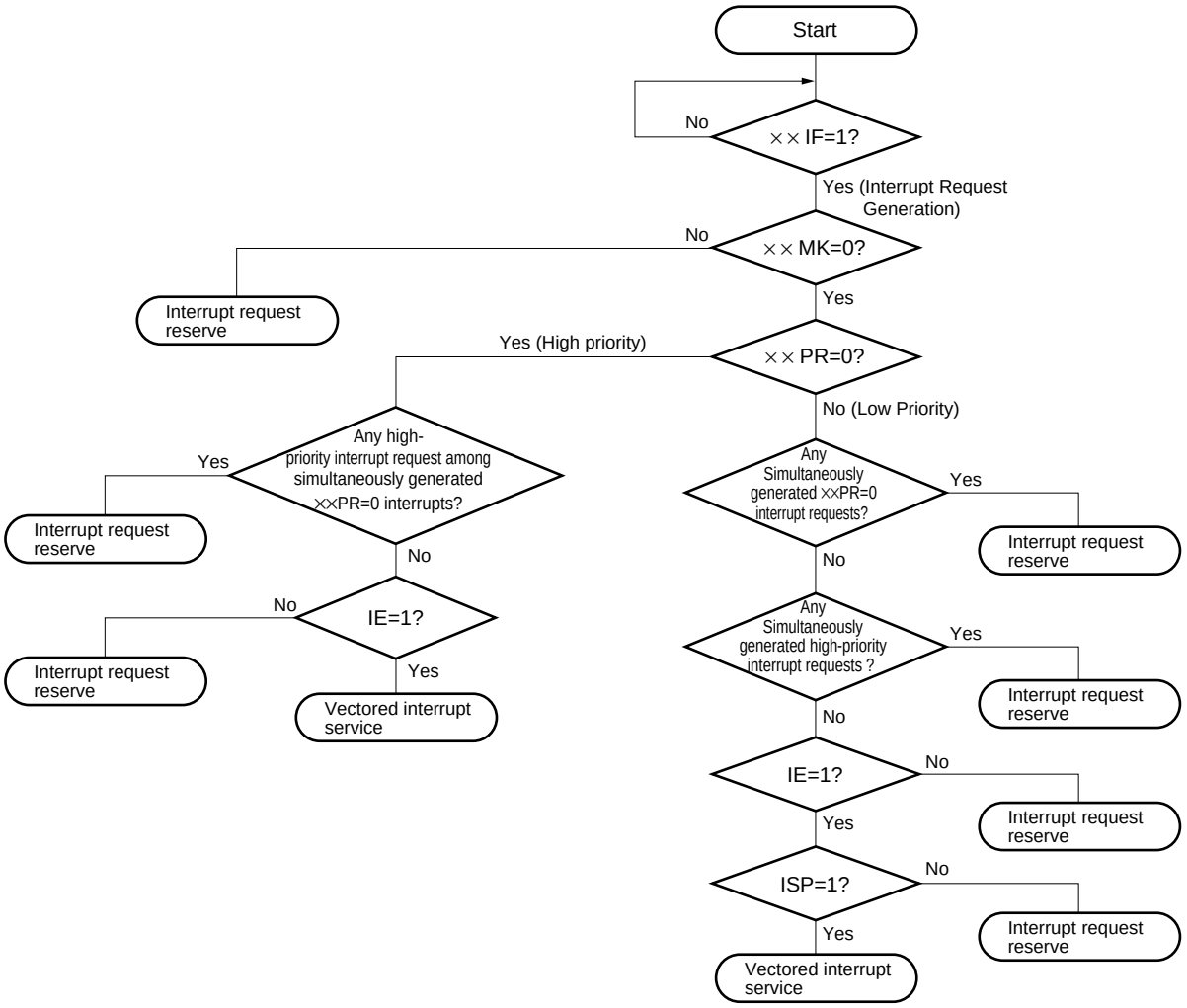
Any reserved interrupt requests are acknowledged when they become acknowledgeable.

Figure 14-13 shows interrupt request acknowledge algorithms.

If a maskable interrupt request is acknowledged, the acknowledged interrupt request is saved in the stacks, the program status word (PSW) and the program counter (PC), in that order, the IE flag is reset to 0, and the acknowledged interrupt priority specification flag contents are transferred to the ISP flag. Further, the vector table data determined for each interrupt request is loaded into PC and branched.

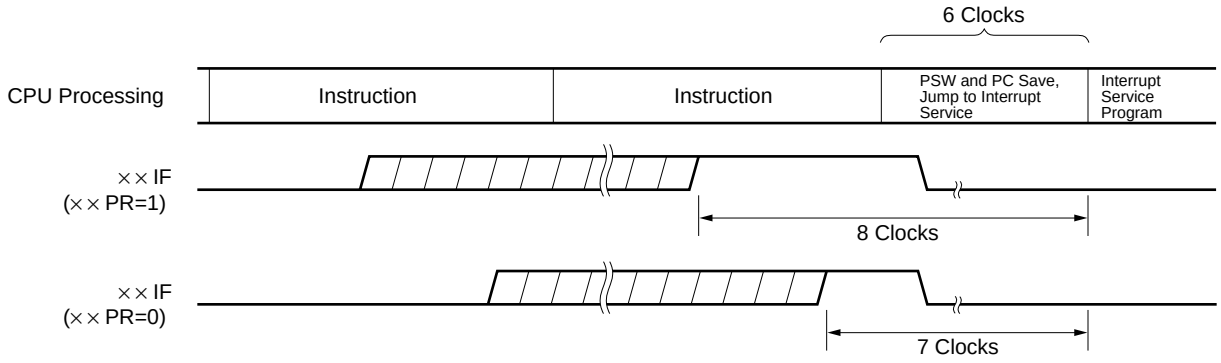
Return from the interrupt is possible with the RETI instruction.

Figure 14-13. Interrupt Request Acknowledge Processing Algorithm



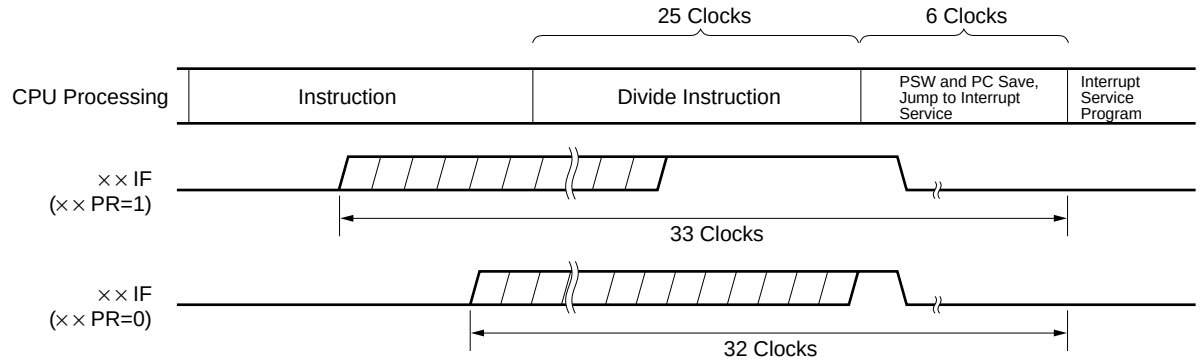
××IF : Interrupt request flag
××MK : Interrupt mask flag
××PR : Priority specification flag
IE : Flag controlling acknowledging maskable interrupt request (1 = enable, 0 = disable)
ISP : Flag indicating priority of interrupt currently serviced (0 = interrupt with high priority serviced, 1 = interrupt request is not acknowledged, or interrupt with low priority serviced)

Figure 14-14. Interrupt Request Acknowledge Timing (Minimum Time)



Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

Figure 14-15. Interrupt Request Acknowledge Timing (Maximum Time)



Remark 1 clock: $\frac{1}{f_{CPU}}$ (f_{CPU}: CPU clock)

14.4.3 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, PSW and 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.

14.4.4 Nesting interrupt

Accepting another interrupt request while an interrupt is being serviced is called nesting interrupts.

Nesting does not take place unless the interrupts (except the non-maskable interrupt) are enabled to be acknowledged (IE = 1). Accepting another interrupt request is disabled (IE = 0) when one interrupt has been acknowledged. Therefore, to enable nesting, the EI flag must be set to 1 during interrupt servicing, to enable the another interrupt.

Nesting interrupts may not occur even when the interrupts are enabled. This is controlled by the priorities of the interrupts. Although two types of priorities, default priority and programmable priority, may be assigned to an interrupt, nesting is controlled by using the programmable priority.

If an interrupt with the same level of priority as or the higher priority than the interrupt currently serviced occurs, that interrupt can be acknowledged and nested. If an interrupt with a priority lower than that of the currently serviced interrupt occurs, that interrupt cannot be acknowledged and nested.

An interrupt that is not acknowledged and nested because it is disabled or it has a low priority is kept pending. This interrupt is acknowledged after servicing of the current interrupt has been completed and one instruction of the main routine has been executed.

Nesting is not enabled while the non-maskable interrupt is being serviced.

Table 14-4 shows the interrupts that can be nested, and Figure 14-16 shows an example of nesting.

Table 14-4. Interrupt Request Enabled for Nesting Interrupt during Interrupt Service

Interrupt being serviced \ Nesting Interrupt Request		Non-maskable Interrupt Request	Maskable Interrupt Request			
			××PR = 0		××PR = 1	
			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 servicing		E	E	D	E	D

Remarks 1. E : Nesting interrupt enable

2. D : Nesting interrupt disable

3. ISP and IE are the flags contained in PSW

ISP = 0 : An interrupt with higher priority is being serviced

ISP = 1 : An interrupt request is not accepted or an interrupt with lower priority is being serviced

IE = 0 : Interrupt request acknowledge is disabled

IE = 1 : Interrupt request acknowledge is enabled

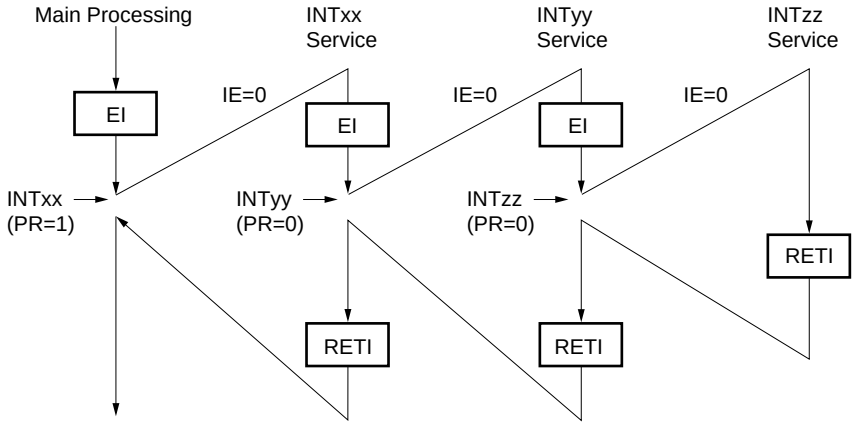
4. ××PR is a flag contained in PR0L and PR0R.

××PR = 0 : Higher priority level

××PR = 1 : Lower priority level

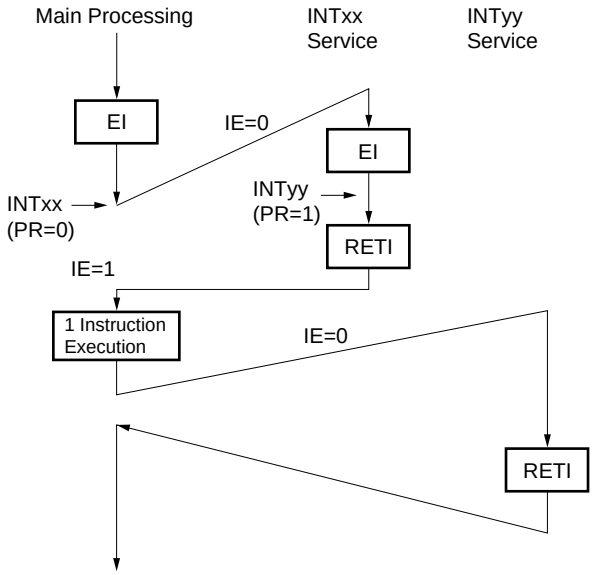
Figure 14-16. Nesting Interrupt Example (1/2)

Example 1. Example where nesting takes place two times



Two interrupt requests, INTyy and INTzz, are acknowledged while interrupt INTxx is serviced, and nesting takes place. Before each interrupt request is acknowledged, the EI instruction is always executed, and the interrupt is enabled.

Example 2. Example where nesting does not take place because of priority control



Interrupt request INTyy that is generated while interrupt INTxx is being serviced is not acknowledged because its priority is lower than that of INTxx, and therefore, nesting does not take place. INTyy request is kept pending, and is acknowledged after one instruction of the main routine has been executed.

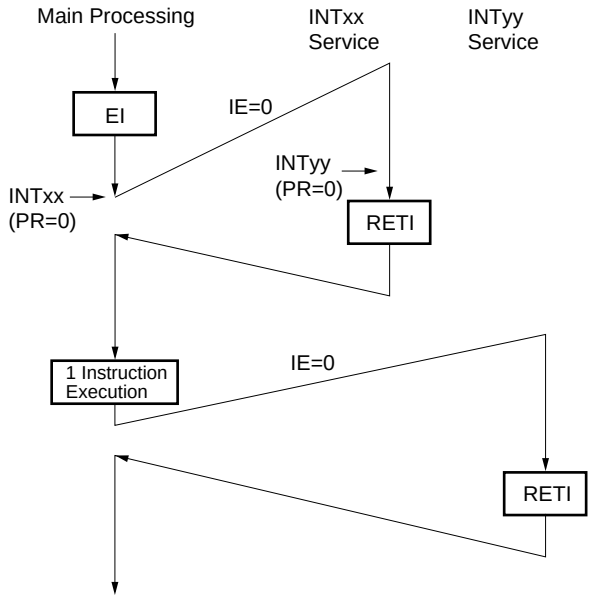
PR = 0 : High-priority level

PR = 1 : Low-priority level

IE = 0 : Acknowledging interrupt request is disabled.

Figure 14-16. Nesting Interrupt Example (2/2)

Example 3. Example where nesting does not take place because interrupts are not enabled



Because interrupts are not enabled (EI instruction is not issued) in interrupt processing INTxx, interrupt request INTyy is not acknowledged, and nesting does not take place. INTyy request is kept pending, and is acknowledged after one instruction of the main routine has been executed.

PR = 0 : High priority level
IE = 0 : Acknowledging interrupts is disabled.

14.4.5 Pending interrupt requests

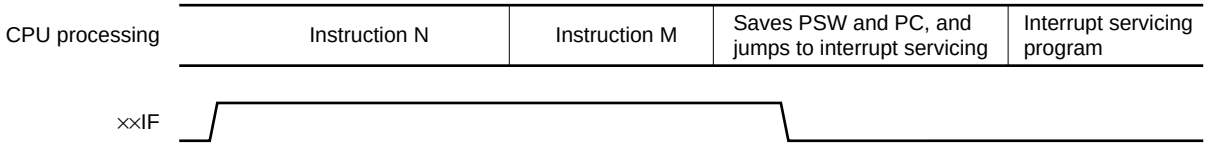
Even if an interrupt request is generated, the following instructions keep it pending.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Instructions manipulating IF0L, IF0H, MK0L, MK0H, PR0L, PR0H, and INTM0 registers

Caution The BRK instruction is not one of the above instructions that keep an interrupt request pending. However, the software interrupt that is started by execution of the BRK instruction clears the IE flag to 0. Therefore, even if a maskable interrupt request is generated while the BRK instruction is being executed, it is not accepted. However, the non-maskable interrupt is accepted.

Figure 14-17 shows the timing at which an interrupt request is accepted.

Figure 14-17. Pending Interrupt Request



Remarks 1. Instruction N: Instruction that keeps interrupt request pending

2. Instruction M: Instruction that does not keep interrupt request pending

3. Operation of ××IF (interrupt request) is not affected by value of ××PR (priority level).

14.5 Test Functions

The test function sets the corresponding test input flag and generates a standby release signal when a falling edge is detected at port 4.

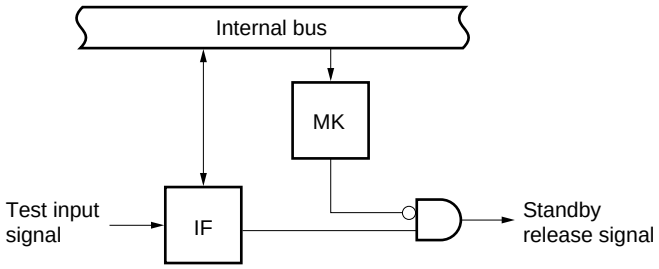
Unlike the interrupt function, this function does not perform vector processing.

There is one test input factor as shown in Table 14-5. The basic configuration is shown in Figure 14-18.

Table 14-5. Test Input Factor

Test Input Factor		Internal/ External
Name	Trigger	
INTPT4	Falling edge detection at port 4	External

Figure 14-18. Basic Configuration of Test Function



Remark IF: test input flag
MK: test mask flag

14.5.1 Registers controlling the test function

The test function is controlled by the following register.

- Key return mode register (KRM)

The name of the test input flag and test mask flag corresponding to the test input signal are listed in Table 14-6.

Table 14-6. Flag Corresponding to Test Input Signal

Test Input Signal Name	Test Input Flag	Test Mask Flag
INTPT4	KRIF	KRMK

(1) Key return mode register (KRM)

This register is used to set enable/disable of standby function clear by key return signal (port 4 falling edge detection).

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

Reset input sets KRM to 02H.

Figure 14-19. Key Return Mode Register Format

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

KRIF	Key Return Signal
0	Not detected
1	Detected (port 4 falling edge detection)

KRMK	Standby Mode Control by Key Return Signal
0	Standby mode release enabled
1	Standby mode release disabled

Caution When port 4 falling edge detection is used, clear KRIF to 0 in program (not cleared to 0 by hardware)

14.5.2 Test input signal acknowledge operation

- **External test signal (INTPT4)**

  When a falling edge is input to the port 4 (P40 to P47) pins, an external test input signal (INTPT4) is generated, setting the KRIF flag. At this time, the standby release signal is generated if it is not masked by the KRMK flag. By using port 4 as key matrix return signal input, whether or not a key input has been applied can be checked from the KRIF status.

[MEMO]

CHAPTER 15 PLL FREQUENCY SYNTHESIZER

15.1 Function of PLL Frequency Synthesizer

The PLL (Phase Locked Loop) frequency synthesizer is used to lock the frequency in the MF (Middle Frequency), HF (High Frequency), and VHF (Very High Frequency) ranges to a specific frequency by means of phase difference comparison.

The PLL frequency synthesizer divides the frequency of the signal input from the VCOL or VCOH pin by using a programmable divider, and outputs the phase difference between the frequency of this signal and reference frequency from the EO0 and EO1 pin.

The following two types of input pins and four frequency division modes are used.

(1) Direct division (MF) mode

The VCOL pin is used.

The VCOH pin goes into a high-impedance state.

(2) Pulse swallow (HF) mode

The VCOL pin is used.

The VCOH pin goes into a high-impedance state.

(3) Pulse swallow (VHF) mode

The VCOH pin is used.

The VCOL pin goes into a high-impedance state.

(4) VCOL, VCOH pin disable

The VCOL and VCOH pin go into a high-impedance state.

However, the phase comparator, reference frequency generator, and charge pump operate.

Therefore, the operation is differ from the PLL frequency synthesizer disabled status which is described later.

These division modes are selected by using the PLL mode select register (PLLMD).

The division value (N value) is set to the programmable divider by using the PLL data register. Frequency division in each of the above modes is carried out according to the value (N value) set to the programmable divider.

Table 15-1 shows the division modes, input pins used (VCOL pin or VCOH pin), and the value that can be set to the programmable divider.

Table 15-1. Division Mode, Input Pin, and Division Value

Division Mode	Pin Used	Value That Can Be Set
Direct division (MF)	VCOL	32 to 2^{12}–1
Pulse swallow (HF)	VCOL	1024 to 2^{17}–1
Pulse swallow (VHF)	VCOH	1024 to 2^{17}–1

Caution For the frequencies that can be actually input, and input amplitude, refer to Electrical Characteristics in Data Sheet.

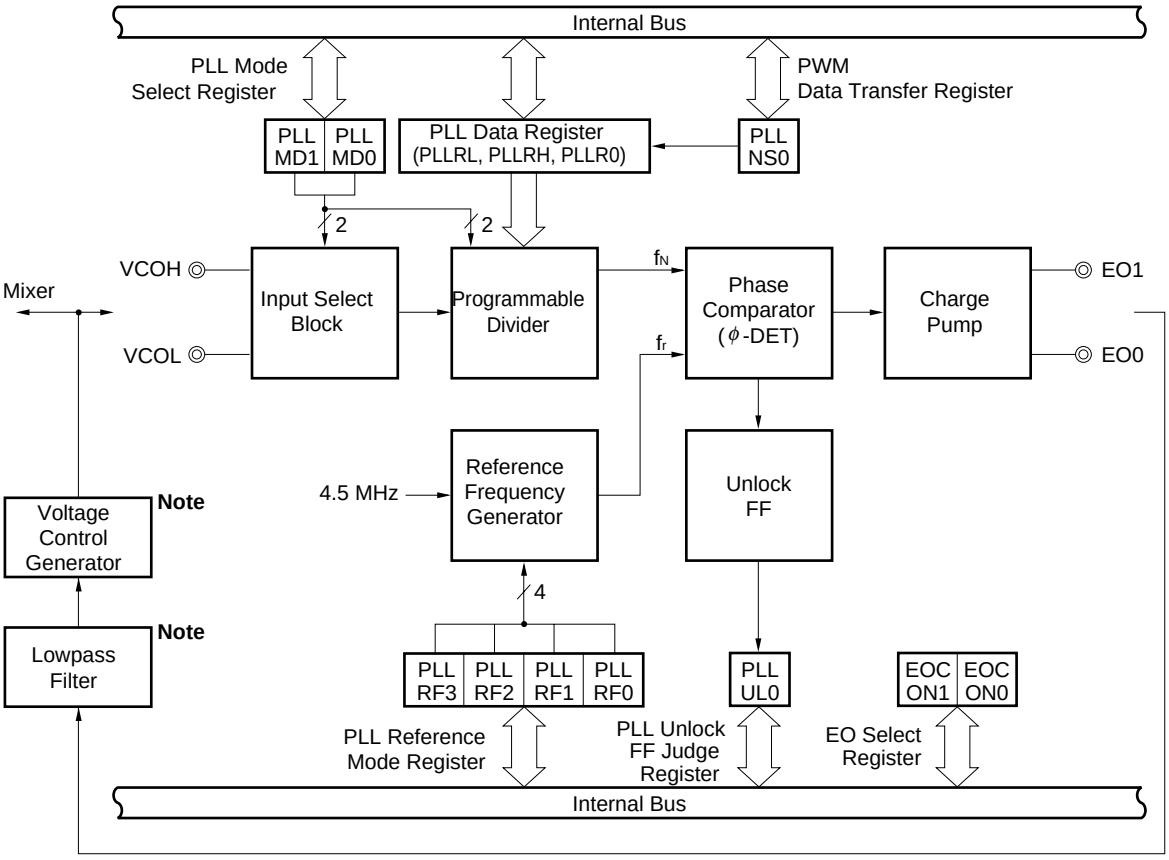
15.2 Configuration of PLL Frequency Synthesizer

The PLL frequency synthesizer consists of the following hardware units.

Table 15-2. PLL Frequency Synthesizer Configuration

Item	Configuration
Data register	PLL data register L (PLLRL) PLL data register H (PLLRH) PLL data register 0 (PLLR0)
Control register	PLL mode select register (PLLMD) PLL reference mode register (PLLRF) PLL unlock FF judge register (PLLUL) PLL data transfer register (PLLNS) EO select register (EOCON)

Figure 15-1. PLL Frequency Synthesizer Block Diagram



Note External circuit

Cautions 1. Be sure to set EOCON0 to 0.

2. With the μPD178004A and 178006A, be sure to set EOCON1 to 0.

(1) PLL data register L (PLLRL), PLL data register H (PLLRH), and PLL data register 0 (PLLR0)

These registers set the division value of the PLL frequency synthesizer. The division value of the PLL frequency synthesizer is made up of 17 bits. The high-order 16 bits of this value are set by the PLL data register L (PLLRL) and PLL data register H (PLLRH). The high-order 16 bits can also be set by the PLL data register (PLLR).

The least significant bit is set by bit 7 (PLLSCN) of the PLL data register 0 (PLLR0).

The contents of these registers are undefined at reset. These registers hold the current values in the STOP and HALT modes.

(2) Input select block

The input select block consists of the VCOL and VCOH pins, and input amplifiers of the respective pins.

(3) Programmable divider

The programmable divider consists of two modulus prescalers, a programmable counter (12 bits), a swallow counter (5 bits), and a division mode select switch.

(4) Reference frequency generator

The reference frequency generator consists of a divider that generates the reference frequency f_r of the PLL frequency synthesizer, and a multiplexer.

(5) Phase comparator

The phase comparator (ϕ-DET) compares the phase of the divided frequency output f_N of the programmable divider with that of the reference frequency output f_r of the reference frequency generator, and outputs an up request signal ($\overline{UP}$) and down request signal ($\overline{DW}$).

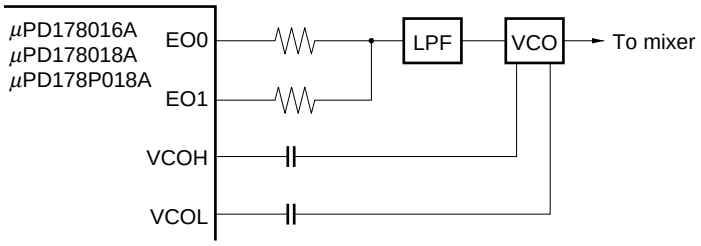
(6) Unlock FF

The unlock FF detects the unlock status of the PLL frequency synthesizer from the up request signal ($\overline{UP}$) and down request signal ($\overline{DW}$) of the phase comparator (ϕ-DET).

(7) Charge pump

The charge pump outputs the result of the output of the phase comparator from the error out pins (EO0 and EO1 pins). With the μPD178016A, 178018A, and 178P018A, EO1 pin can be set to high-impedance by setting bit 1 (EOCON1) of the EO select register (EOCON).

An application example is as follows:



LPF: Low pass filter VCO: Voltage control generator

- **To lock target frequency at high-speed**
  By setting EO0 and EO1 pins to error out output, the output current capability and voltage control capability of LPF are improved.
- **Steady state**
  By setting only EO0 pin to error out output, the stability of LPF is maintained.

15.3 Registers Controlling PLL Frequency Synthesizer

The PLL frequency synthesizer is controlled by the following five registers.

- PLL mode select register (PLLMD)
- PLL reference mode register (PLLRF)
- PLL unlock FF judge register (PLLUL)
- PLL data transfer register (PLLNS)
- EO select register (EOCON)

(1) PLL mode select register (PLLMD)

This register selects the input pin and division mode of the PLL frequency synthesizer.

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

The value of this register is set to 00H at reset and in the STOP mode.

In the HALT mode, it holds the value immediately before the HALT mode was set.

Figure 15-2. PLL Mode Select Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
PLLMD	0	0	0	0	0	0	PLLMD1	PLLMD0	FFA0H	00H	R/W

PLLMD1	PLLMD0	Selects Division Mode of PLL Frequency Synthesizer and VCO Input Pin
0	0	Disables VCOL and VCOH pins[Note]
0	1	Direct division (VCOL pin and MF mode)
1	0	Pulse swallow (VCOH pin and VHF mode)
1	1	Pulse swallow (VCOL pin and HF mode)

Note This does not mean that the PLL is disabled. The VCOL and VCOH pins go into a high-impedance state. The EO0 and EO1 pin go low.

Remark Bits 2 through 7 are fixed to 0 by hardware.

(2) PLL reference mode register (PLLRF)

This register selects the reference frequency f_r of the PLL frequency synthesizer and sets the disabled status of the PLL frequency synthesizer.

PLLRF is set by using 1-bit or 8-bit memory manipulation instruction.

The value of this register is set to 0FH at reset and in the STOP mode.

In the HALT mode, it holds the value immediately before the HALT mode was set.

Figure 15-3. PLL Reference Mode Register Format

Symbol	7	6	5	4	③	②	①	⓪	Address	After Reset	R/W
PLLRF	0	0	0	0	PLLRF3	PLLRF2	PLLRF1	PLLRF0	FFA1H	0FH	R/W

PLLRF3	PLLRF2	PLLRF1	PLLRF0	Sets Reference Frequency f_r of PLL Frequency Synthesizer
0	0	0	0	Setting prohibited
0	0	0	1	Setting prohibited
0	0	1	0	5 kHz
0	0	1	1	10 kHz
0	1	0	0	Setting prohibited
0	1	0	1	Setting prohibited
0	1	1	0	25 kHz
0	1	1	1	50 kHz
1	0	0	0	3 kHz
1	0	0	1	9 kHz
1	0	1	0	Setting prohibited
1	0	1	1	PLL disable**Note**
1	1	0	0	1 kHz
1	1	0	1	Setting prohibited
1	1	1	0	Setting prohibited
1	1	1	1	PLL disable**Note**

Note When PLL disable is selected, the VCOL, VCOH, EO0, and EO1 pins go into a high-impedance state.

Remark Bits 4 through 7 are fixed to 0 by hardware.

(3) PLL unlock FF judge register (PLLUL)

This register detects whether the PLL frequency synthesizer is in the unlock status.

Because this register is an R&RESET register, it is reset to 0 after it has been read.

At reset, the value of this register is set to 0×H[Note 1].

In the STOP and HALT modes, this register holds the value immediately before the STOP or HALT mode was set.

Figure 15-4. PLL Unlock FF Judge Register Format

Symbol	7	6	5	4	3	2	1	⓪	Address	After Reset	R/W
PLLUL	0	0	0	0	0	0	0	PLLUL0	FFA2H	0×H[Note 1]	R[Note 2]

PLLUL0	Detects Status of Unlock FF
0	Unlock FF = 0: PLL lock status
1	Unlock FF = 1: PLL unlock status

Notes 1. The value of bit 0 (PLLUL0) at reset differs depending on the type of reset that has been executed (refer to the table below).

2. Bit 0 (PLLUL0) is R&Reset.

		7	6	5	4	3	2	1	0
At reset	Power-ON clear	0	0	0	0	0	0	0	Undefined
	Watchdog timer	↓	↓	↓	↓	↓	↓	↓	Retained
	$\overline{\text{RESET}}$ input	↓	↓	↓	↓	↓	↓	↓	Retained
STOP mode		↓	↓	↓	↓	↓	↓	↓	Retained
HALT mode		↓	↓	↓	↓	↓	↓	↓	Retained

Remark Bits 1 through 7 are fixed to 0 by hardware.

(4) PLL data transfer register (PLLNS)

This register transfers the values of the PLL data registers (PLLRL, PLLRH, and PLLR0) to the programmable counter and swallow counter.

The value of this register is 00H at reset and in the STOP mode.

In the HALT mode, this register holds the previous value immediately before the HALT mode is set.

Figure 15-5. PLL Data Transfer Register Format

Symbol	7	6	5	4	3	2	1	⓪	Address	After Reset	R/W
PLLNS	0	0	0	0	0	0	0	PLLNS0	FFA3H	00H	W

PLLNS0	Transfers Value of PLL Data Register to Programmable Counter and Swallow Counter
0	Does not transfer
1	Transfers

Remark Bits 0 through 7 are fixed to 0 by hardware.

(5) EO select register (EOCON)

This register sets the current value of the regulator and pin status.

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

The value of this register is set to 00H at reset and in the STOP mode.

In the HALT mode, this register holds the value immediately before the HALT mode was set.

Figure 15-6. EO Select Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
EOCON	0	0	0	0	0	0	EOCON1	EOCON0	FFA4H	00H	R/W

EOCON1	EO1 Pin Status
0	Error out output
1	High-impedance (setting prohibited with the μPD178004A and 178006A.)

EOCON0	Current Value of Regulator of EO1 Pin
0	Selects 4 mA TYP.
1	Setting prohibited.

Cautions 1. Be sure to set EOCON0 to 0.

2. With the μPD178004A and 178006A, be sure to set EOCON1 to 0.

Remark Bit 1 through 7 are fixed to 0 by hardware.

15.4 Operation of PLL Frequency Synthesizer

15.4.1 Operation of each block of PLL frequency synthesizer

(1) Operation of input select block and programmable divider

The input select block and programmable divider select the input pin and division mode of the PLL frequency synthesizer and divide the frequency in the selected division mode, according to the setting of the PLL mode select register (PLLMD).

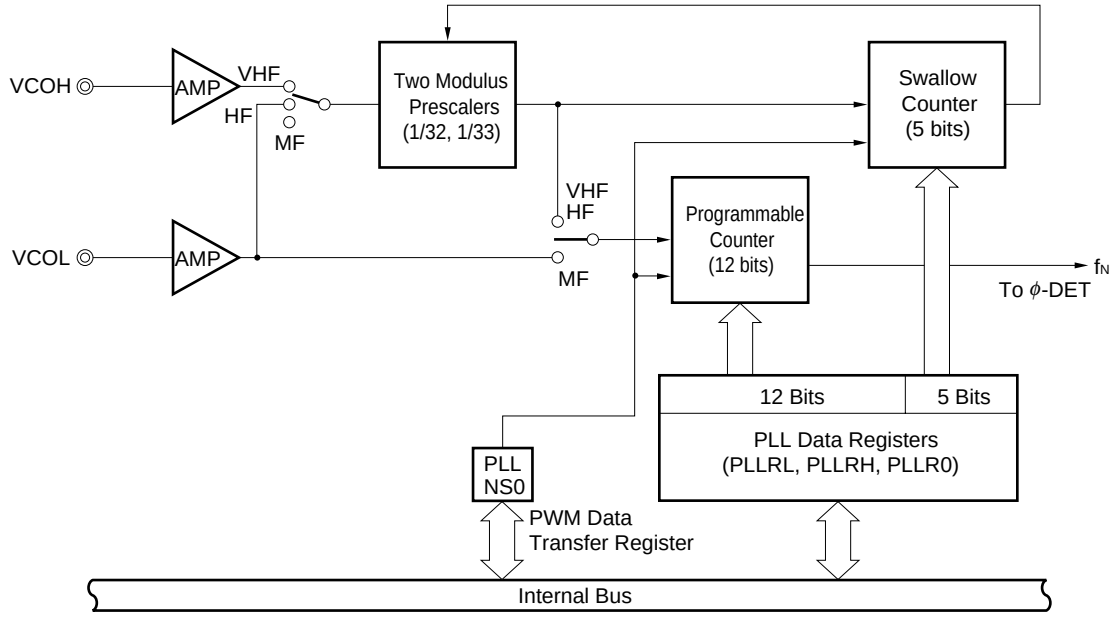
The programmable counter (12 bits) and pulse swallow counter (5 bits) are binary counters.

The division value (N value) is set to the programmable counter and swallow counter by the PLL data registers (PLLRL, PLLRH, and PLLR0).

When the N value has been transferred to the programmable counter and swallow counter, frequency division is performed in the selected division mode according to the status of bit 0 (PLLNS0) of the PLL data transfer register.

Figure 15-7 shows the configuration of the input select block and programmable divider.

Figure 15-7. Input Select Block and Programmable Divider Configuration



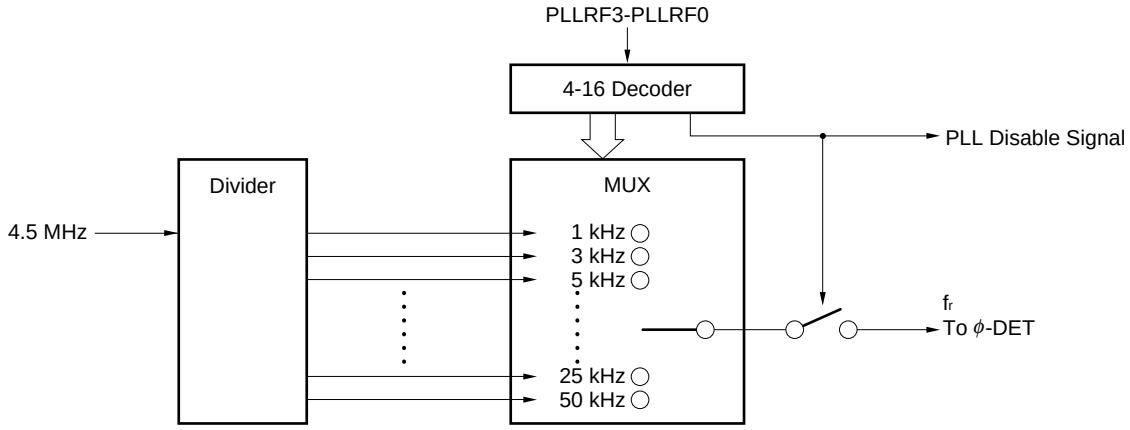
(2) Operation of reference frequency generator

The reference frequency generator divides the 4.5 MHz output of the crystal resonator and generates 12 types of reference frequency f_r for the PLL frequency synthesizer.

Reference frequency f_r is selected by the PLL reference mode register (PLLRF).

Figure 15-8 shows the configuration of the reference frequency generator.

Figure 15-8. Reference Frequency Generator Configuration



(3) Operation of phase comparator (ϕ-DET)

Figure 15-9 shows the configuration of the phase comparator (ϕ-DET), charge pump, and unlock FF.

The phase comparator (ϕ-DET) compares the phase of the divided frequency f_N of the programmable divider with that of the reference frequency f_r of the reference frequency generator, and outputs an up request signal, $\overline{UP}$, or a down request signal, $\overline{DW}$.

If the divided frequency f_N is lower than the reference frequency f_r, the up request signal is output. If f_N is higher than f_r, the down request signal is output.

Figure 15-10 shows the relation among reference frequency f_r, divided frequency f_N, up request signal $\overline{UP}$, and down request signal $\overline{DW}$.

When the PLL is disabled, neither the up nor the down request signal is output.

The up and down request signals are input to the charge pump and unlock FF.

Figure 15-9. Phase Comparator, Charge Pump, and Unlock FF Configuration

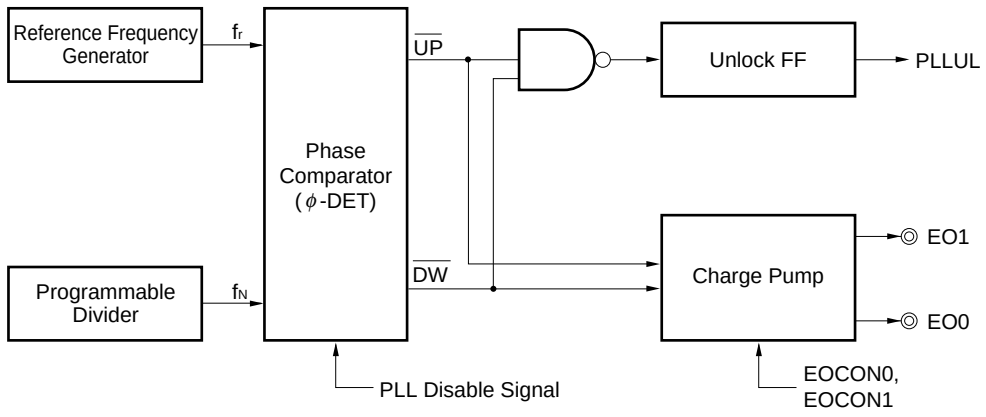
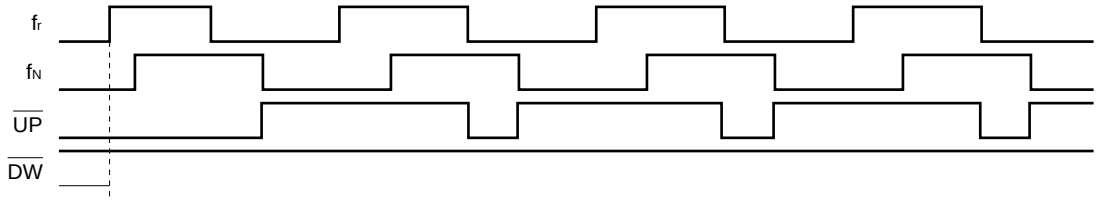
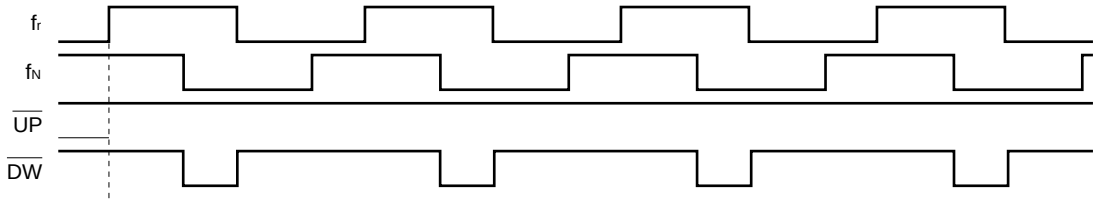


Figure 15-10. Relationship between f_r, f_N, $\overline{UP}$, and $\overline{DW}$

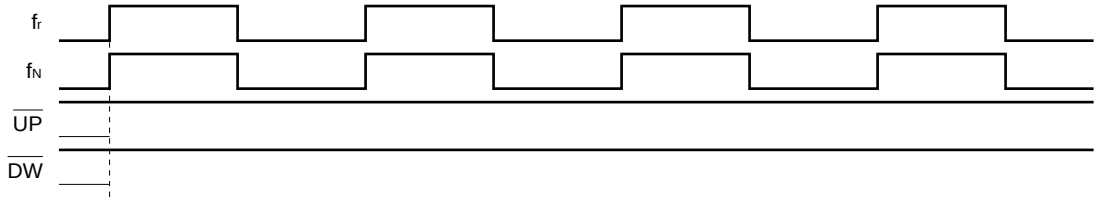
(a) If f_N advances f_r in phase



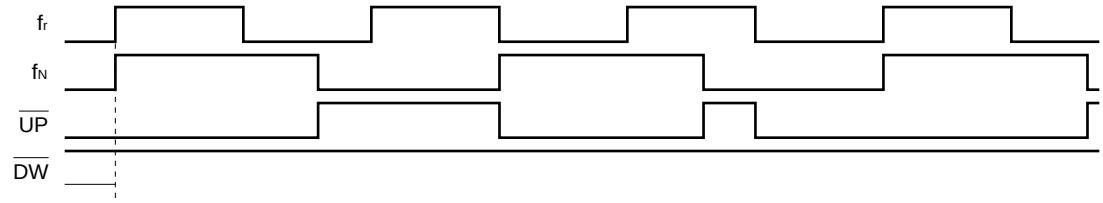
(b) If f_N advances f_r in phase



(c) If f_N and f_r are in phase



(d) If f_N is lower than f_r



(4) Operation of charge pump

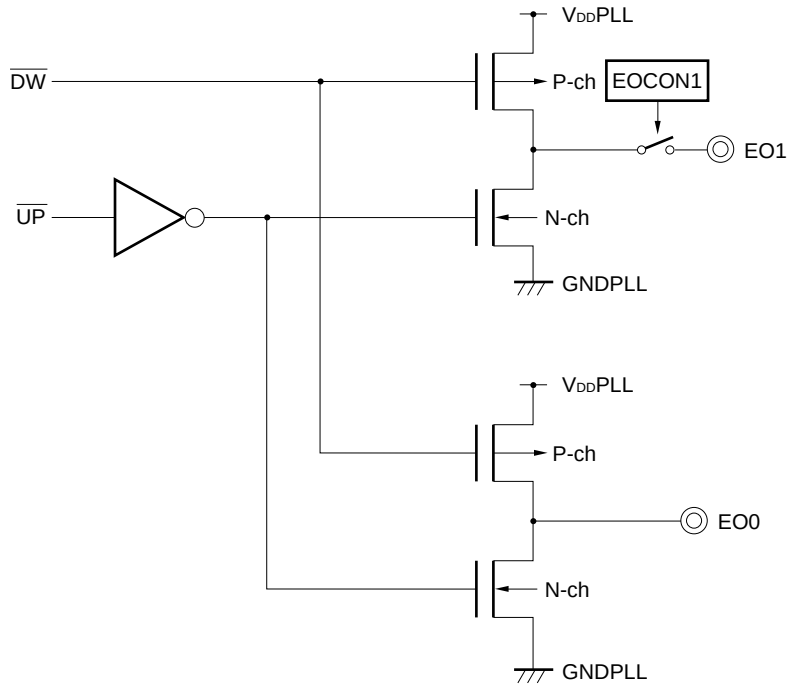
The charge pump outputs the result of the up request ($\overline{UP}$) or down request ($\overline{DW}$) signal from the phase comparator (ϕ-DET) from the error out pins (EO0 and EO1 pins). Table 15-3 shows the output signals. The EO0 pin is of voltage-driven type, and EO1 pin is of current-driven type.

Figure 15-11 shows the configuration of the error out pins.

Table 15-3. Error Out Output Signal

Relationship between Divided Frequency f_N and Reference Frequency f_r	Error Out Output Signal
When $f_r > f_N$	Low level
When $f_r < f_N$	High level
When $f_r = f_N$	Floating (high impedance)

Figure 15-11. Error Out Pins Configuration



Caution With the μPD178016A, 178018A and 178P018A, the status of EO1 pin can be changed from error out output to high-impedance.

(5) Operation of unlock FF

The unlock FF detects the unlock status of the PLL frequency synthesizer.

It detects the unlock status of the PLL frequency synthesizer from the up request signal $\overline{UP}$ and down request signal $\overline{DW}$ of the phase comparator (ϕ-DET).

Because either of the up request or down request signal outputs a low level in the unlock status, the unlock status can be detected by using this low-level signal.

The status of the unlock FF is detected by bit 0 (PLLUL0) of the PLL unlock FF judge register (PLLUL).

The unlock FF is set at the cycle of reference frequency f_r selected at that time.

The PLL unlock FF judge register is reset when its contents have been read.

To read the PLL unlock FF judge register, therefore, it must be read at a cycle longer than the cycle ($1/f_r$) of the reference frequency.

15.4.2 Operation to set N value of PLL frequency synthesizer

The division value (N value) is set to the programmable counter (12 bits) and swallow counter (5 bits) by the PLL data registers (PLLRL, PLLRH, and PLLR0).

When the N value has been transferred to the programmable counter and swallow counter by bit 0 (PLLNS0) of the PLL data transfer register (PLLNS), frequency division is carried out in the selected division mode.

Examples of setting the N value in the respective division modes (MF, HF, and VHF) are shown below.

(1) Direct division mode (MF)

(a) Calculating division value N (value set to PLL data register)

$$N = \frac{f_{VCOL}}{f_r}$$

where, f_{VCOL} : input frequency of V_{COL} pin

f_r : reference frequency

(b) Example of setting PLL data register

An example of setting the PLL data register to receive broadcasting stations in the following MW band is shown below.

Receive frequency : 1422 kHz (MW band)

Reference frequency : 9 kHz

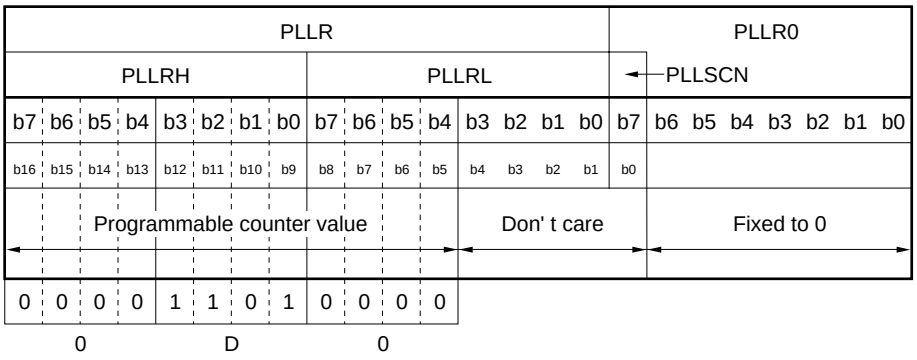
Intermediate frequency : 450 kHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOL}}{f_r} = \frac{1422 + 450}{9} = 208 \text{ (decimal)}$$

$$= 0D0H \text{ (hexadecimal)}$$

Data is set to the PLL data registers (PLLR and PLLR0) as follows:



After setting the above PLL data registers (PLLR and PLLR0), data must be transferred to the programmable counter by setting bit 0 (PLLNS0) of the PLL data transfer register (PLLNS).

(2) Pulse swallow mode (HF)

(a) Calculating division value N (value set to PLL data register)

$$N = \frac{f_{VCOL}}{f_r}$$

where, f_{VCOL} : input frequency of V_{COL} pin
f_r : reference frequency

(b) Example of setting PLL data register

An example of setting the PLL data register to receive broadcasting stations in the following SW band is shown below.

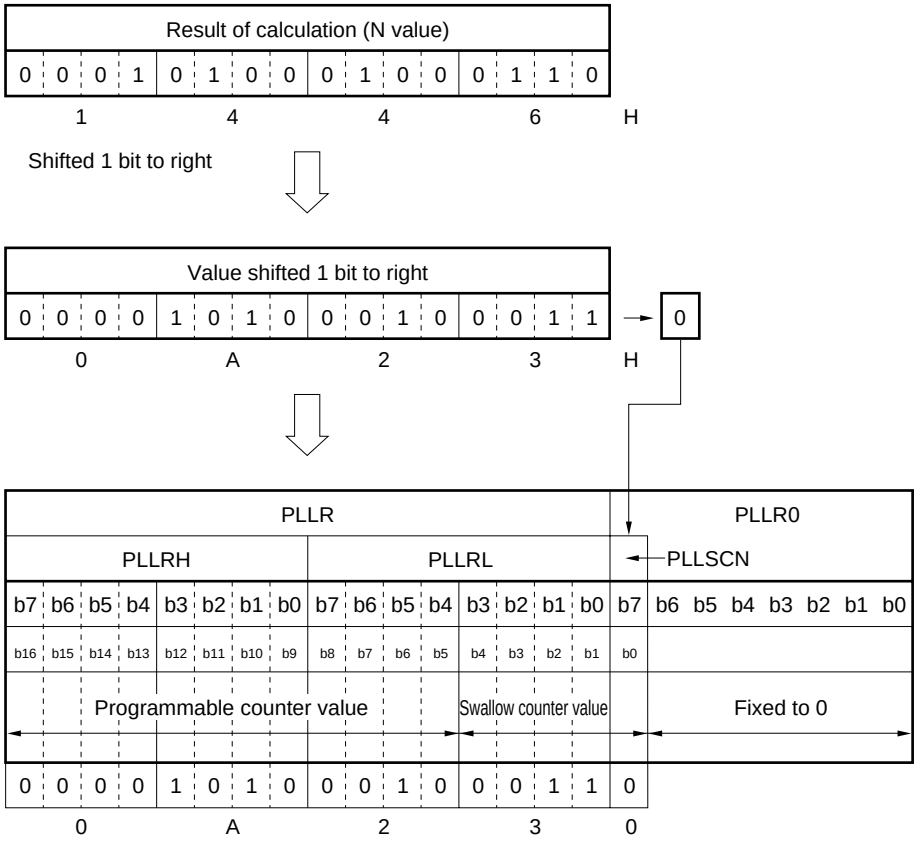
Receive frequency : 25.50 MHz (SW band)
Reference frequency : 5 kHz
Intermediate frequency : 450 kHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOL}}{f_r} = \frac{25500 + 450}{5} = 5190 \text{ (decimal)}$$

= 01446H (hexadecimal)

Because the least significant bit of the division value N must be set to bit 7 (PLLSCN) of the PLL data register 0 (PLLR0), data must be set by shifting the result of the above calculation 1 bit to the right. Data is set to the PLL data registers (PLLR and PLLR0) as follows:



After setting the above PLL data registers (PLLR and PLLR0), data must be transferred to the programmable counter and swallow counter by setting bit 0 (PLLNS0) of the PLL data transfer register (PLLNS).

In this example, a value of half the N value is set to the high-order 16 bits of the PLL data register (PLLR) by shifting the N value resulting from calculation 1 bit to the right.

If the N value is calculated as follows with the least significant bit of the N value in PLLSCN fixed to 0, the result of the calculation (N_{PLLR}) can be set to the PLL data register (PLLR) as is.

If the calculation result is set in this way, however, the input frequency (f_{VCOL}) is $2 \times f_r$ (reference frequency) of the set value N_{PLLR}.

$$N_{PLLR} = \frac{f_{VCOL}}{2f_r}$$

(3) Pulse swallow mode (VHF)

(a) Calculating division value N (value set to PLL data register)

$$N = \frac{f_{VCOH}}{f_r}$$

where, f_{VCOH} : input frequency of VCOH pin
f_r : reference frequency

(b) Example of setting PLL data register

An example of setting the PLL data register to receive broadcasting stations in the following FM band is shown below.

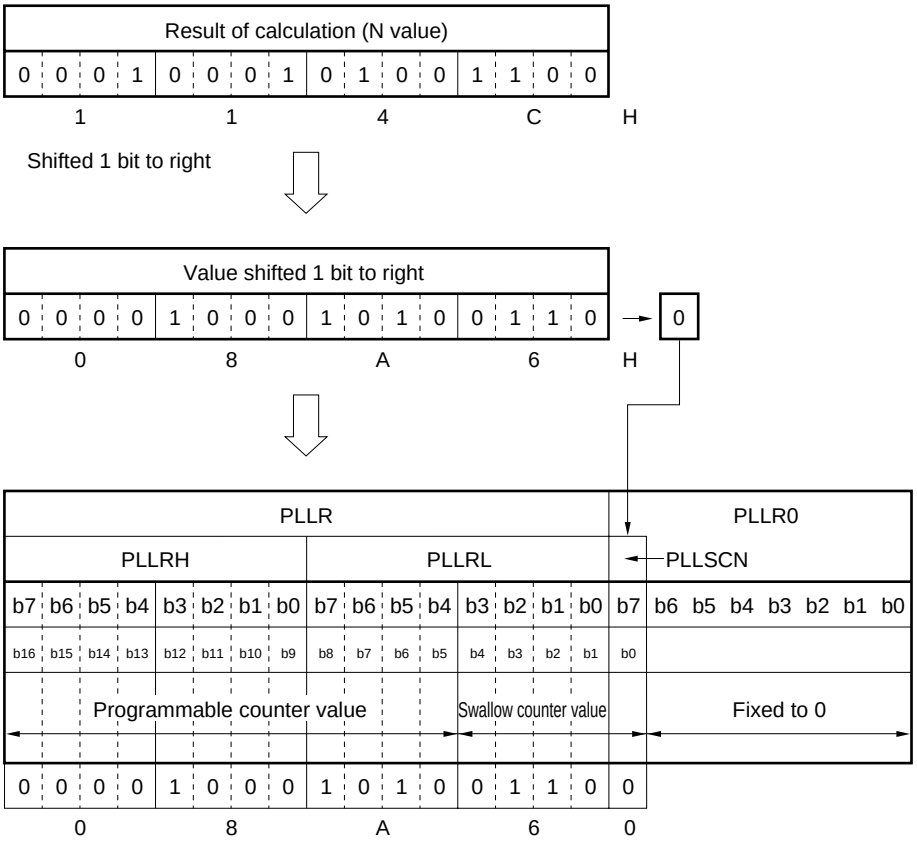
Receive frequency : 100.0 MHz (FM band)
Reference frequency : 25 kHz
Intermediate frequency : +10.7 MHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOH}}{f_r} = \frac{100.0 + 10.7}{0.025} = 4428 \text{ (decimal)}$$

$$= 0114CH \text{ (hexadecimal)}$$

Because the least significant bit of the division value N must be set to the PLL data register 0 (PLLR0), data must be set by shifting the value calculated by the above expression 1 bit to the right.
Data is set to the PLL data registers (PLLR and PLLR0) as follows:



After setting the above PLL data registers (PLLR and PLLR0), data must be transferred to the programmable counter and swallow counter by setting bit 0 (PLLNS0) of the PLL data transfer register (PLLNS).

In this example, a value of half the N value is set to the high-order 16 bits of the PLL data register (PLLR) by shifting the N value resulting from calculation 1 bit to the right.

If the N value is calculated as follows with the least significant bit of the N value in PLLSCN fixed to 0, the result of the calculation (N_{PLLR}) can be set to the PLL data register (PLLR) as is.

If the calculation result is set in this way, however, the input frequency (f_{VCOH}) is $2 \times f_r$ (reference frequency) of the set value N_{PLLR}.

$$N_{PLLR} = \frac{f_{VCOH}}{2f_r}$$

15.5 PLL Disable Status

The PLL frequency synthesizer can be stopped (PLL disabled status) by performing any of the following settings while the PLL frequency synthesizer is operating.

- Setting value of PLL reference mode register to 0BH or 0FH to set PLL disabled status
- Setting STOP mode with the STOP instruction
- Setting reset status with the reset function

The following table shows the operation of each block and the status of each register in the PLL disabled status.

Table 15-4. Operation of Each Block and Register Status in PLL Disabled Status

Block/Register	Status in PLL Disabled Status
VCOL and VCOH pins	High impedance
Programmable divider	Division stops
Reference frequency generator	Output stops
Phase comparator	Output stops
EO0 and EO1 pin	High impedance
Current regulator	Stops operation
PLL mode select register	Retains value on execution of write instruction
PLL data register	
PLL unlock FF judge register	

15.6 Notes on PLL Frequency Synthesizer

- **Notes on using PLL frequency synthesizer**
  Because the input pins (VCOL and VCOH pins) of the PLL frequency synthesizer are provided with an AC amplifier, cut the DC component of the input signal by connecting a capacitor to the input pins in series.
  The potential of the selected input pin is intermediate (about $1/2V_{DD}$). The input pin not selected goes into a high-impedance state.
  For the frequencies that can be actually input and input amplitude, refer to Electrical Characteristics in Data Sheet.

CHAPTER 16 FREQUENCY COUNTER

16.1 Function of Frequency Counter

The frequency counter counts the intermediate frequency (IF) of a tuner.

It counts the intermediate frequency input to the FMIFC or AMIFC pin for a specific time (1 ms, 4 ms, 8 ms, or open) with a 16-bit counter. The count value of the frequency counter is stored to the IF counter register.

For the range of the frequency that can be input to the FMIFC and AMIFC pins, refer to Electrical Characteristics in Data Sheet.

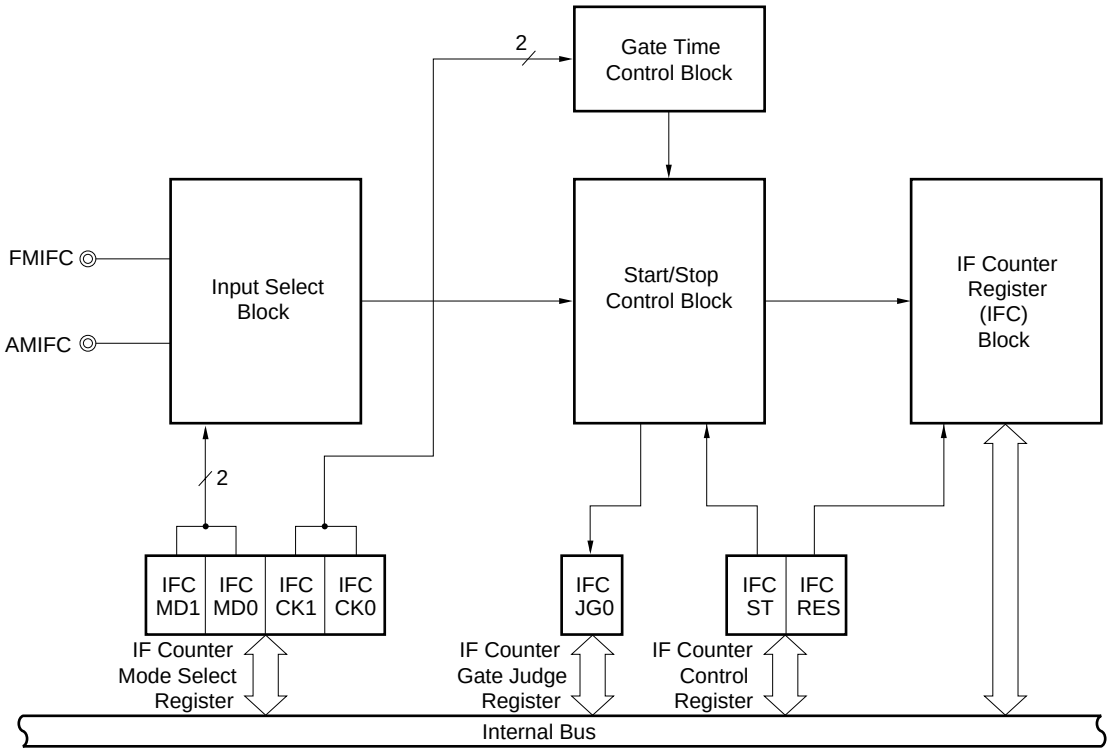
16.2 Configuration of Frequency Counter

The frequency counter consists of the following hardware units.

Table 16-1. Frequency Counter Configuration

Item	Configuration
Counter register	IF counter register (IFC)
Control register	IF counter mode select register (IFCMD) IF counter control register (IFCR) IF counter gate judge register (IFCJG)

Figure 16-1. Frequency Counter Block Diagram



(1) IF counter input select block

The IF counter input select block selects the pin to be used from the FMIFC and AMIFC pins, and a count mode.

(2) Gate time control block

The gate time control block sets a gate time (count time).

(3) Start/stop control block

The start/stop control block starts counting by the IF counter register and detects the end of counting.

(4) IF counter register block

The IF counter register block is a 16-bit register that counts up the frequency input in the set gate time. The count value is stored to the IF counter register (IFC). When the count value reaches FFFFH, the IF counter register holds FFFFH at the next input, and stops counting. The value of this register is reset to 0000H at reset or in the STOP mode. In the HALT mode, it holds the current count value.

16.3 Registers Controlling Frequency Counter

The frequency counter is controlled by the following three registers.

- IF counter mode select register (IFCMD)
- IF counter control register (IFCR)
- IF counter gate judge register (IFCJG)

(1) IF counter mode select register (IFCMD)

This register selects the input pin of the frequency counter, and selects a mode and gate time (count time).

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

The value of this register is reset to 00H at reset or in the STOP mode.

In the HALT mode, this register holds the value immediately before the HALT mode is set.

Figure 16-2. IF Counter Mode Select Register Format

Symbol	7	6	5	4	③	②	①	⓪	Address	After Reset	R/W
IFCMD	0	0	0	0	IFCMD1	IFCMD0	IFCCK1	IFCCK0	FFA9H	00H	R/W

IFCMD1	IFCMD0	Selects Frequency Counter Pin and Mode
0	0	Disables FMIFC AMIFC pins[Note]
0	1	AMIFC pin, AMIF count mode
1	0	FMIFC pin, FMIF count mode
1	1	FMIFC pin, AMIF count mode

IFCCK1	IFCCK0	Selects Gate Time
0	0	1 ms
0	1	4 ms
1	0	8 ms
1	1	Open

Note The FMIFC and AMIFC pins go into a high-impedance state.

Remark Bits 4 through 7 are fixed to 0 by hardware.

(2) IF counter control register (IFCR)

This register starts counting by the IF counter register and clears the IF counter register.

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

The value of this register is reset to 00H at reset and in the STOP mode.

In the HALT mode, this register holds the value immediately before the HALT mode is set.

Figure 16-3. IF Counter Control Register Format

Symbol	7	6	5	4	3	2	①	⓪	Address	After Reset	R/W
IFCR	0	0	0	0	0	0	IFCST	IFCRES	FFACH	00H	W

IFCST	Starts IF Counter Register
0	Nothing is affected
1	Starts counting

IFCRES	Clears Data of IF Counter Register
0	Nothing is affected
1	Clears data of IF counter register

Remark Bits 2 through 7 are fixed to 0 by hardware.

(3) IF counter gate judge register (IFCJG)

This register detects opening/closing of the gate of the frequency counter.

The value of this register is reset to 00H at reset and in the STOP mode.

In the HALT mode, this register holds the value immediately before the HALT mode is set.

Figure 16-4. IF Counter Gate Judge Register Format

Symbol	7	6	5	4	3	2	1	⓪	Address	After Reset	R/W
IFCJG	0	0	0	0	0	0	0	IFCJG0	FFABH	00H	R

IFCJG0	Detects Opening/Closing of Gate of Frequency Counter
0	Gate is closed
1	• If gate time is set to other than open Status until gate is closed after IFCST has been set to 1 • If gate time is set to open Status where gate is open as soon as it has been set to be opened

Remark Bits 1 through 7 are fixed to 0 by hardware.

Caution IFCJG0 remains set even if the IF counter register overflows and stops counting, until the set gate time expires.

16.4 Operation of Frequency Counter

(1) Select an input pin, mode, and gate time by using the IF counter mode select register (IFCMD). Figure 16-5 shows a block that selects an input pin and mode.

(2) Set bit 0 (IFCRES) of the IF counter control register (IFCR) to 1, and clears the data of the IF counter register.

(3) Set bit 1 (IFCST) of the IF counter control register (IFCR) to 1.

(4) The gate is opened only for the set gate time since a 1-kHz internal signal has risen after IFCST was set. If the gate time is set to be opened, the gate is opened as soon as it has been specified to be opened.
Bit 0 (IFCJG0) of the IF counter gate judge register (IFCJG) is automatically set to 1 as soon as IFCST has been set to 1.
When the gate time has expired, bit 0 (IFCJG0) of the IF counter gate judge register (IFCJG) is automatically cleared to 0. If it is specified that the gate be open, however, IFCJG0 is not automatically cleared. In this case, set a gate time. Figure 16-6 shows the gate timing of the frequency counter.

(5) While the gate opens the frequency input to the selected FMIFC or AMIFC pin, the IF counter register counts the frequency.
If the FMIFC pin is used in the FMIF count mode, however, the input frequency is divided by half before it is counted.

The relationship between count value x (decimal), input frequencies (f_{FMIFC} and f_{AMIFC}), and gate time (T_{GATE}) is shown below.

- FMIF count mode (FMIFC pin)

$$f_{FMIFC} = \frac{x}{T_{GATE}} \times 2 \text{ (kHz)}$$

- AMIF count mode (FMIFC or AMIFC pin)

$$f_{AMIFC} = \frac{x}{T_{GATE}} \text{ (kHz)}$$

Figure 16-5. Input Pin and Mode Selection Block Diagram

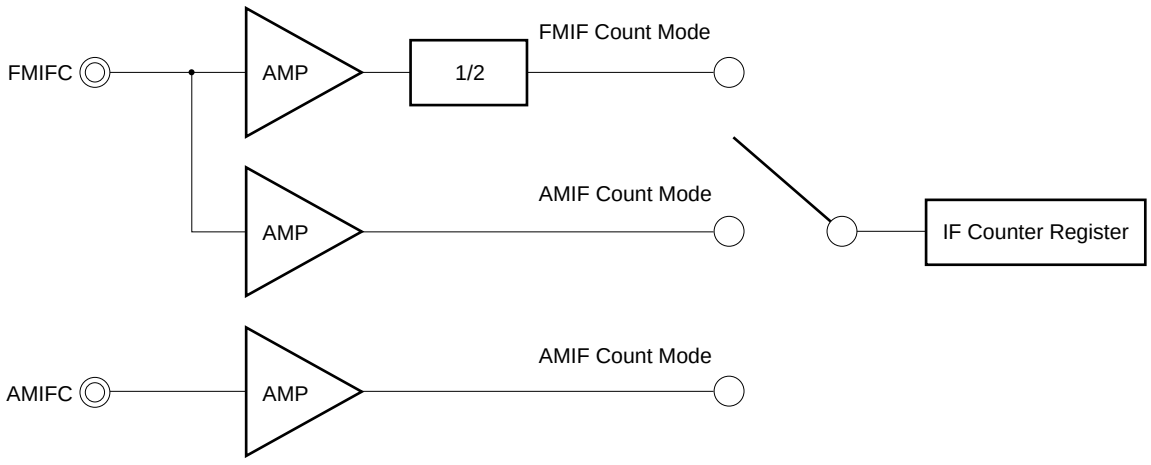
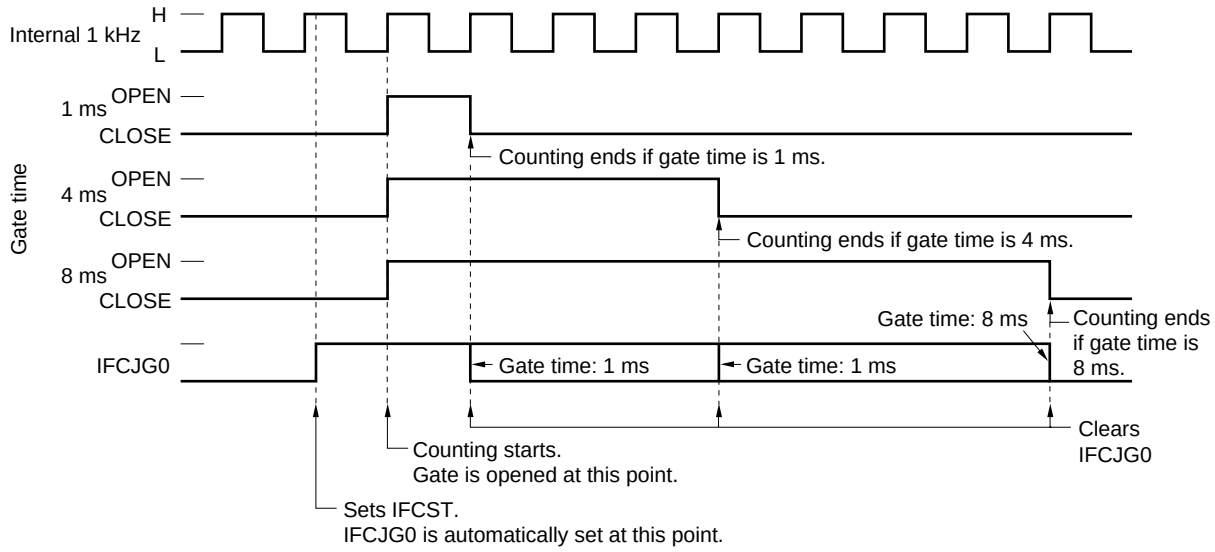
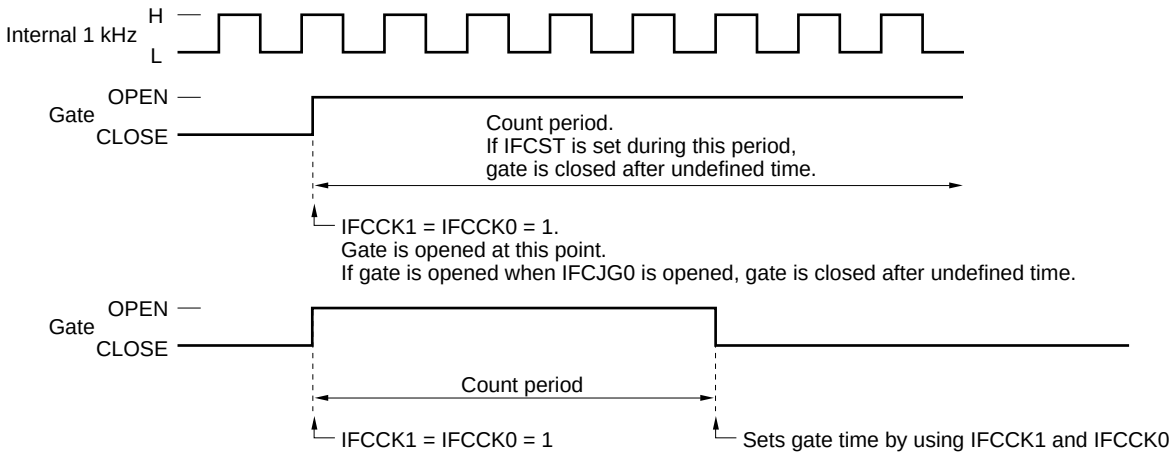


Figure 16-6. Gate Timing of Frequency Counter

(a) If gate time is set to 1, 4, or 8 ms



(b) If gate is set to be open



Caution If counting is started by using IFCST while this gate is open, the gate is closed after undefined time. To open the gate, therefore, do not set IFCST to 1.

Remark IFCST : Bit 1 of IF counter control register (IFCR)
IFCJG0 : Bit 0 of IF counter gate judge register (IFJG)
IFCCK1, 0 : Bit 1 and 0 of IF counter mode select register (IFCMD)

16.5 Notes on Frequency Counter

(1) Notes on using frequency counter

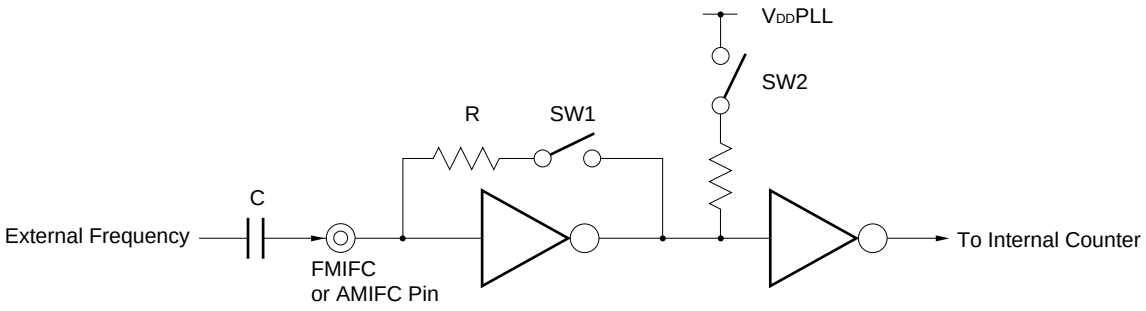
Because signals are input to the frequency counter from an input pin (FMIFC or AMIFC pin) with an AC amplifier as shown in Figure 16-7, cut the DC component of the input signals by using capacitor C.

If the FMIFC or AMIFC pin is selected by the IF counter mode select register, switch SW1 turns ON, and switch SW2 turns OFF. As a result, the voltage on the pin is about $1/2V_{DD}$.

Unless the voltage has risen to a sufficient intermediate level at this time, counting may not be performed normally because the AC amplifier is not in the normal operating range.

Therefore, make sure that sufficient wait time elapses after a pin has been selected and before counting is started (IFCST = 1).

Figure 16-7. Frequency Counter Input Pin Circuit



(2) Notes in HALT mode

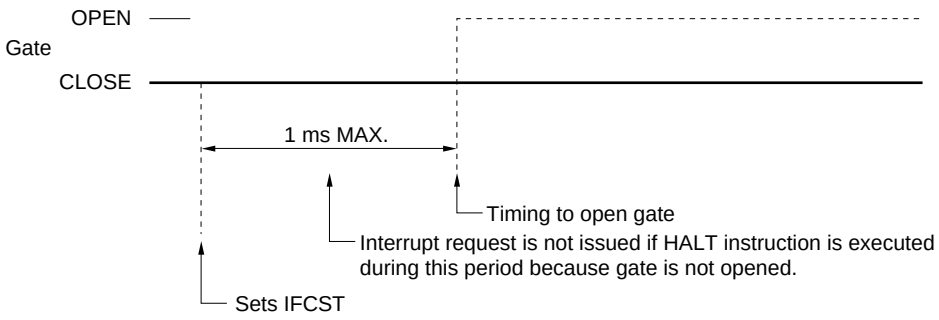
The FMIFC and AMIFC pins hold the status immediately before the HALT status was set.

To release the HALT mode by using the interrupt of the frequency counter at this time, the following point must be noted.

The gate will not be opened if the HALT instruction is executed after counting has been started by IFCST before the gate is actually opened.

Therefore, wait for at least 1 ms before executing the HALT instruction.

Figure 16-8. Gate Status When HALT Instruction Is Executed



(3) Error of frequency counter

The error of the frequency counter includes an error of gate time and a count error.

(1) Error of gate time

The gate time of the frequency counter is created by dividing 4.5 MHz.
Therefore, if 4.5 MHz is shifted “+x” ppm, the gate time is also shifted “-x” ppm.

(2) Count error

The frequency counter counts the frequency at the rising edge of the input signal.
If a high level is input to the pin when the gate is opened, therefore, one excess pulse is counted. When the gate is closed, however, counting is not affected by the status of the pin.
Therefore, the count error is “maximum + 1”.

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function is designed 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 such as in watch applications.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the system clock oscillator stops and the whole system stops. CPU current consumption can be considerably decreased.

Data memory low-voltage hold (down to V_{DD} = 1.8 V) is possible. Thus, the STOP mode is effective to hold data memory contents with ultra-low current consumption. Because this mode can be cleared upon 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 cleared, select the HALT mode if it is necessary to start processing immediately upon interrupt request.

In any mode, 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.

However, in STOP mode, the contents of registers and flags of the 8-bit timer, PLL frequency synthesizer and frequency counter are not held. Refer to **CHAPTER 7 8-BIT TIMER**, **CHAPTER 15 PLL FREQUENCY SYNTHESIZER**, and **CHAPTER 16 FREQUENCY COUNTER**.

Cautions 1. When proceeding to the STOP mode, be sure to stop the peripheral hardware operation and execute the STOP instruction.

2. The following sequence is recommended for power consumption reduction of the A/D converter: first clear bit 7 (CS) of ADM to 0 to stop the A/D conversion operation. Clear bit 3 (ADON) of ADIS to 0 and turn the resistor string power supply off. Then execute the HALT or STOP instruction.

17.1.2 Standby function control register

A wait time after the STOP mode is cleared upon 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.

Reset input sets OSTS to 04H. However, it takes $2^{17}/f_X$, not $2^{18}/f_X$, until the STOP mode is cleared by $\overline{\text{RESET}}$ input.

Figure 17-1. 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		
				MCS = 1	MCS = 0
0	0	0	$2^{12}/f_{XX}$	$2^{12}/f_X$(90 μs)	$2^{13}/f_X$(1.82 ms)
0	0	1	$2^{14}/f_{XX}$	$2^{14}/f_X$(3.64 ms)	$2^{15}/f_X$(7.28 ms)
0	1	0	$2^{15}/f_{XX}$	$2^{15}/f_X$(7.28 ms)	$2^{16}/f_X$(14.6 ms)
0	1	1	$2^{16}/f_{XX}$	$2^{16}/f_X$(14.6 ms)	$2^{17}/f_X$(29.1 ms)
1	0	0	$2^{17}/f_{XX}$	$2^{17}/f_X$(29.1 ms)	$2^{18}/f_X$(58.3 ms)
Other than above			Setting prohibited		

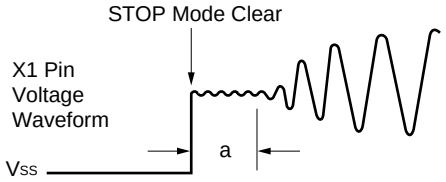
Remarks 1. f_{XX} : System clock frequency (f_X or $f_X/2$)

2. f_X : System clock oscillation frequency

3. MCS : Oscillation mode selection register (OSMS) bit 0

4. Values in parentheses apply to operating at f_X = 4.5 MHz

Caution The wait time when the STOP mode is released does not include the time required for the clock oscillation to start after the STOP mode has been released (see "a" in the figure below), regardless of whether the mode has been released by the $\overline{\text{RESET}}$ signal or an interrupt request.



17.2 Standby Function Operations

17.2.1 HALT mode

(1) HALT mode set and operating status

The HALT mode is set by executing the HALT instruction.

The operating status in the HALT mode is described below.

Table 17-1. HALT Mode Operating Status

Item	Status
Clock generation circuit	Can oscillate system clock. Stops clock supply to CPU.
CPU	Stops operating.
Port	Holds status before HALT mode is set.
8-bit timer/event counter	Holds operation before HALT mode is set and can operate.
8-bit timer (including D/A converter)	
Basic timer	
Watchdog timer	
Buzzer output control circuit	
A/D converter	
Serial interface	Automatic transfer/reception function stops. Other functions hold operation before HALT mode is set and can operate.
External interrupt	INTP0 stops only if sampling clock is $f_{XX}/2^N$. Otherwise, holds operation before HALT mode is set and can operate. INTP1 through INTP6 hold operation before HALT mode is set and can operate.
PLL frequency synthesizer	Hold operation before HALT mode is set and can operate.
Frequency counter	
Power-ON clear circuit	Can operate

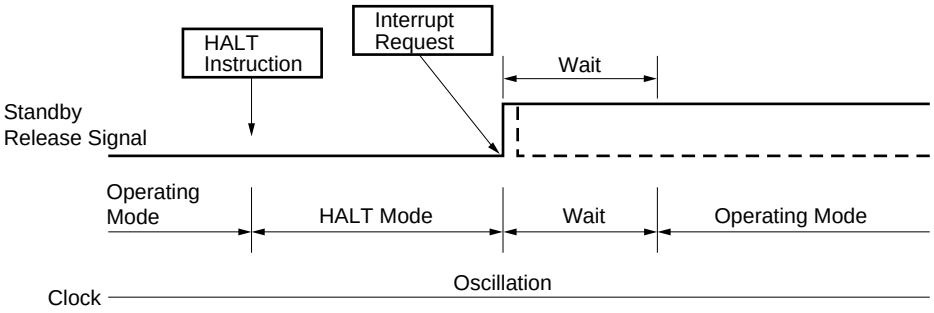
(2) HALT mode release

The HALT mode can be released with the following four types of sources.

(a) Release upon unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledge is enabled, vectored interrupt service is carried out. If disabled, the next address instruction is executed.

Figure 17-2. HALT Mode Release upon Interrupt 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) Release upon non-maskable interrupt request

When a non-maskable interrupt is generated, the HALT mode is released and vectored interrupt service is carried out whether interrupt acknowledge is enabled or disabled.

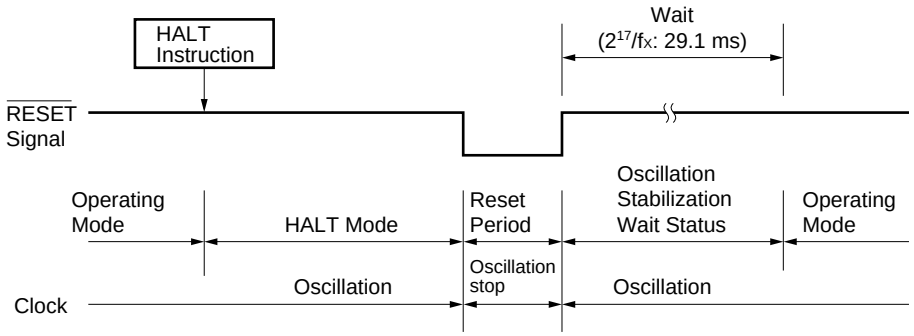
(c) Release upon unmasked test input

If an unmasked test is input, the HALT mode is released and the next address instruction of the HALT instruction is executed.

(d) Release upon $\overline{\text{RESET}}$ input

If a $\overline{\text{RESET}}$ signal is input, the HALT mode is released. As is the case with normal reset operation, a program is executed after branch to the reset vector address.

Figure 17-3. HALT Mode Release by $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. ( ): At fx = 4.5 MHz

Table 17-2. Operation after HALT Mode Release

Release Source	MK××	PR××	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

17.2.2 STOP mode

(1) STOP mode set and operating status

The STOP mode is set by executing the STOP instruction.

Cautions 1. When the STOP mode is set, the X2 pin is internally connected to V_{DD} via a pull-up resistor to minimize the leakage current at the crystal oscillator.

2. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately 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 17-3. STOP Mode Operating Status

Item	Status
Clock generation circuit	Can oscillate system clock. Stops clock supply to CPU.
CPU	Stops operating.
Port	Holds status before HALT mode is set.
8-bit timer/event counter	Operable when TI1 and TI2 are set to count clock. Holds status before STOP mode is set. Other functions stop operation and cannot operate.
8-bit timer (including D/A converter)	Operation stops and cannot operate.
Basic timer	
Watchdog timer	
Buzzer output control circuit	
A/D converter	
Serial interface	Automatic transfer/receive function stops and cannot operate. Other functions hold operation before STOP mode is set and can operate.
External interrupt	INTP0 stops and cannot operate. INTP1 to INTP6 hold operation before STOP mode is set and can operate.
PLL frequency synthesizer	Operation stops and cannot operate.
Frequency counter	
Power-ON clear circuit	Can operate

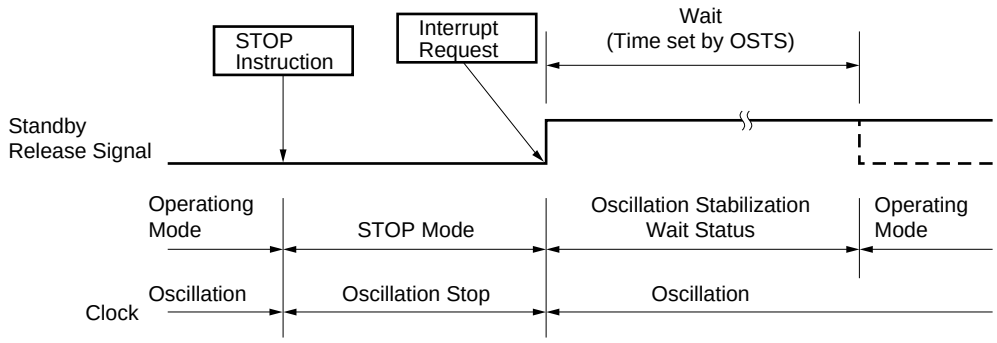
(2) STOP mode release

The STOP mode can be released with the following three types of sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. If interrupt request acknowledge is enabled after the lapse of oscillation stabilization time, vectored interrupt service is carried out. If interrupt request acknowledge is disabled, the next address instruction is executed.

Figure 17-4. STOP Mode Release by Interrupt Request Generation



Remark The broken line indicates the case when the interrupt request which has released the standby status is acknowledged.

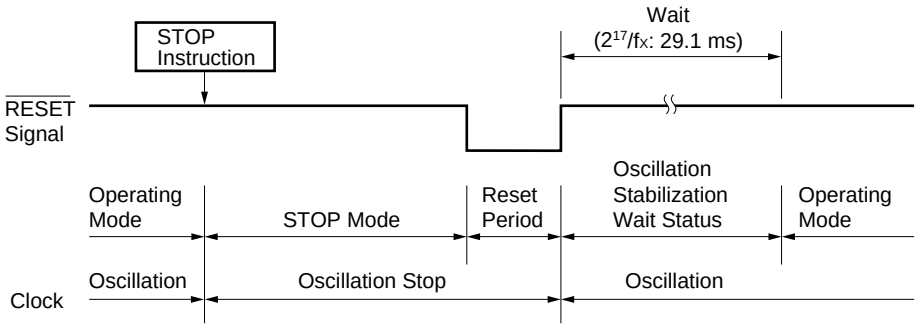
(b) Release by unmasked test input

If an unmasked test is input, the STOP mode is released. After the lapse of oscillation stabilization time, the instruction at the next address of the STOP instruction is executed.

(c) Release by $\overline{\text{RESET}}$ input

If a $\overline{\text{RESET}}$ signal is input, the STOP mode is released, and after the lapse of oscillation stabilization time, reset operation is carried out.

Figure 17-5. Release by STOP Mode $\overline{\text{RESET}}$ Input



Remarks 1. fx: System clock oscillation frequency

2. ( ): At fx = 4.5 MHz

Table 17-4. Operation after STOP Mode Release

Release Source	MK××	PR××	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 18 RESET FUNCTION

18.1 Reset Function

The following three operations are available to generate the reset signal.

(1) External reset input with $\overline{\text{RESET}}$ pin
(2) Internal reset by inadvertent program loop time detection watchdog timer
(3) Internal reset by power-ON clear (POC)

(1) External reset input by $\overline{\text{RESET}}$ pin

When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset, and each hardware unit enters the status shown in Table 18-1. While the reset signal is input and during the oscillation stabilization time immediately after the $\overline{\text{RESET}}$ signal has been deasserted, each pin goes into a high-impedance state (the P132 through P134 pins go low, however).

The $\overline{\text{RESET}}$ signal is deasserted when a high level is input to the $\overline{\text{RESET}}$ pin, and the program execution is started after the oscillation stabilization time ($2^{17}/f_X$) has elapsed.

(2) Internal reset by inadvertent program loop time detection of watchdog timer

Reset is effected and each hardware unit enters the status shown in Table 18-1 when the watchdog timer overflow. While reset is effected and during the oscillation stabilization time immediately after the effect of reset has been cleared, each pin goes into a high-impedance state (the P132 through P134 pins go low, however).

Reset by the watchdog timer is cleared immediately after reset has been effected, and the program execution is started after the oscillation stabilization time ($2^{17}/f_X$) has elapsed.

(3) Internal reset by power-ON clear (POC)

Reset is effected by means of power-ON clear under the following conditions:

- If supply voltage is less than 3.5 V**Note** on power application
- If supply voltage drops to less than 2.5 V**Note** in STOP mode
- If supply voltage drops to less than 3.5 V**Note** while operation is performed with CPU clock of $f_X/2$ or lower (including HALT mode). Also if supply voltage drops to less than 4.5 V**Note** while operation is performed with CPU clock of f_X (including HALT mode)

When these reset conditions of power-ON clear are satisfied, reset is effected, and each hardware unit enters the status shown in Table 18-1. While the reset signal is input and during the oscillation stabilization time immediately after the reset signal has been deasserted, each pin goes into a high-impedance state (the P132 through P134 pins go low, however).

Reset by power-ON clear is cleared if the supply voltage rises beyond a specific level, and the program execution is started after the oscillation stabilization time ($2^{17}/f_X$) has elapsed.

Note These voltage values are maximum values. Actually, reset is effected at a voltage lower than these.

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, system clock oscillation remains stopped.

3. When the STOP mode is cleared by $\overline{\text{RESET}}$ input, the STOP mode register contents are held during reset input. However, the I/O port pin becomes high-impedance. Output dedicated port pin (P132 to P134) becomes low level regardless of the previous status.

Figure 18-1. Reset Function Block Diagram

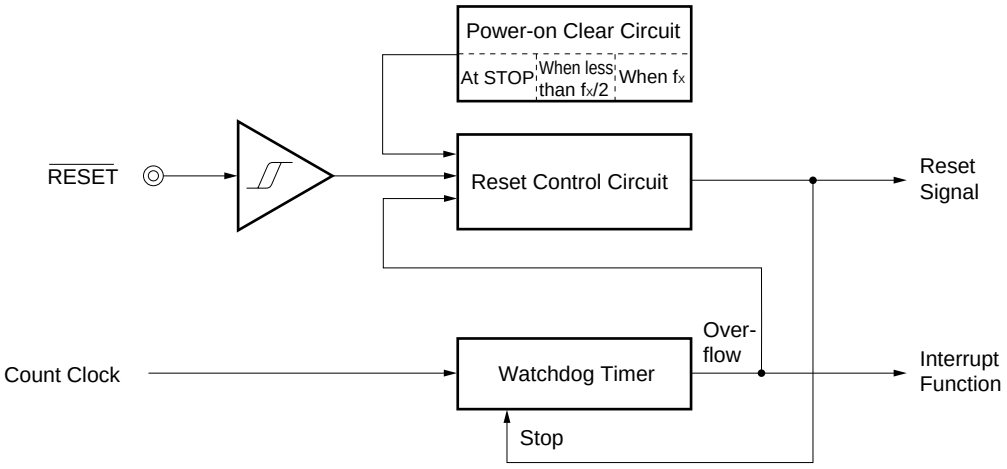
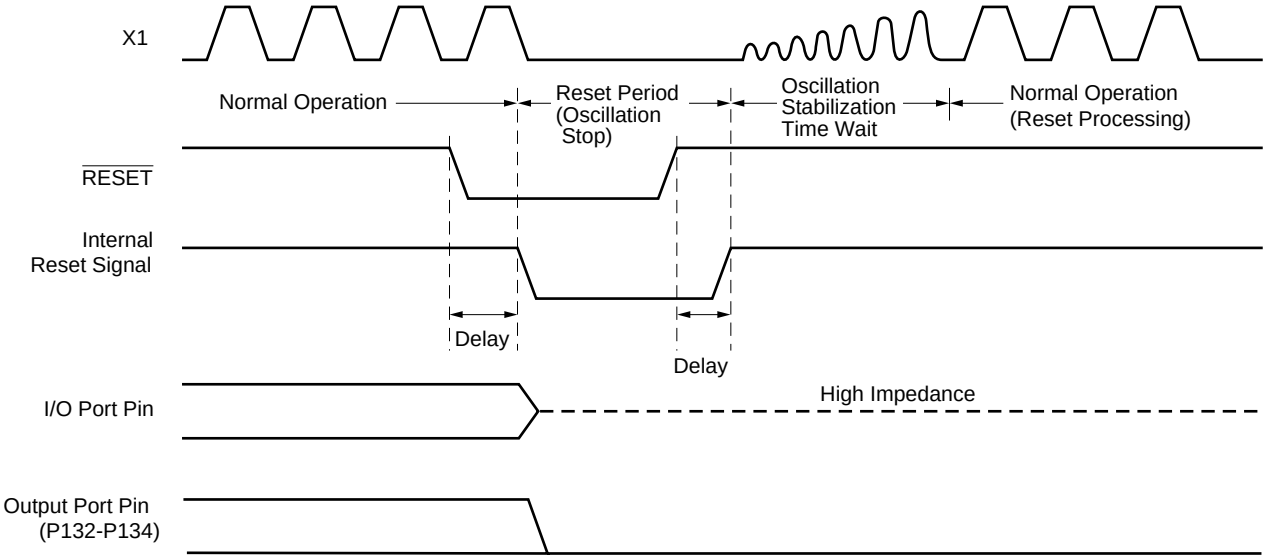


Figure 18-2. Timing of Reset Input by $\overline{\text{RESET}}$ Input

(a) In normal operating mode



(b) In STOP mode

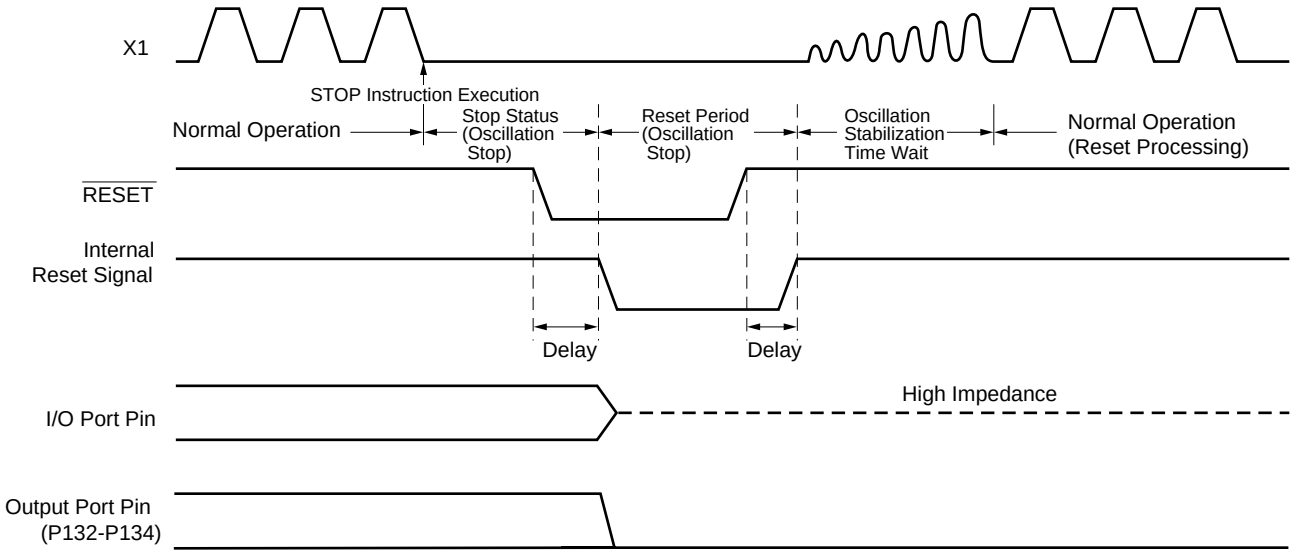


Figure 18-3. Timing of Reset due to Watchdog Timer Overflow

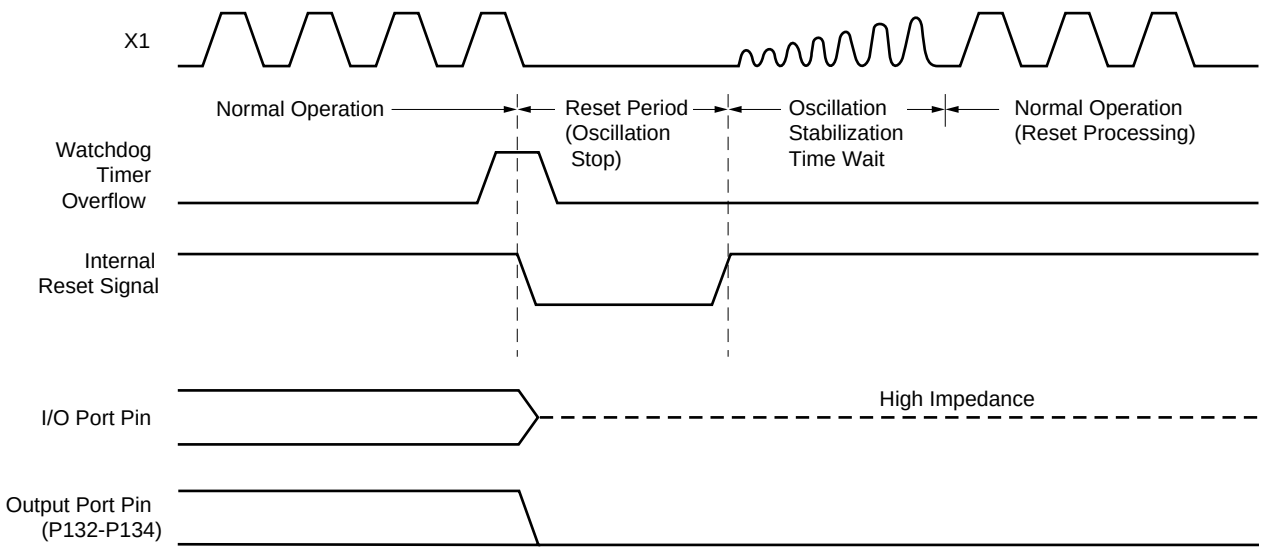
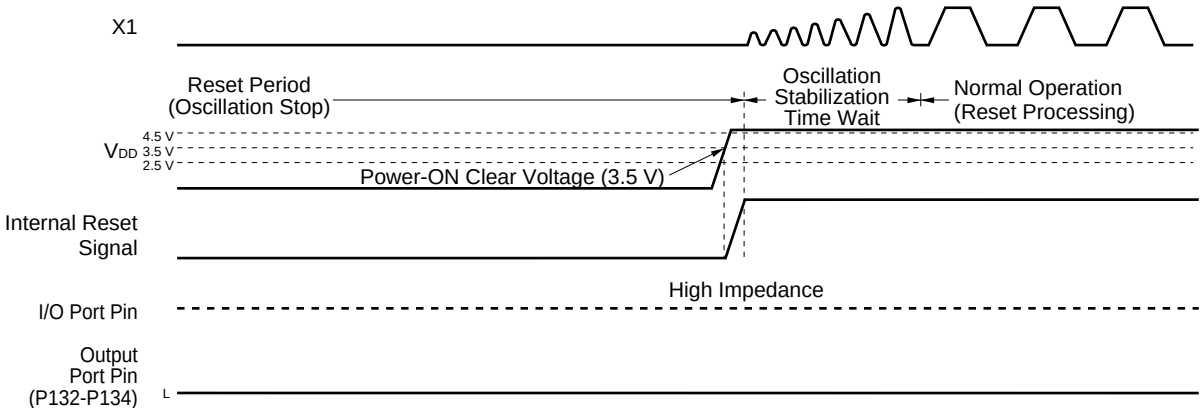


Figure 18-4. Timing of Reset by Power-ON Clear (1/2)

(a) At Power-ON



(b) In STOP mode

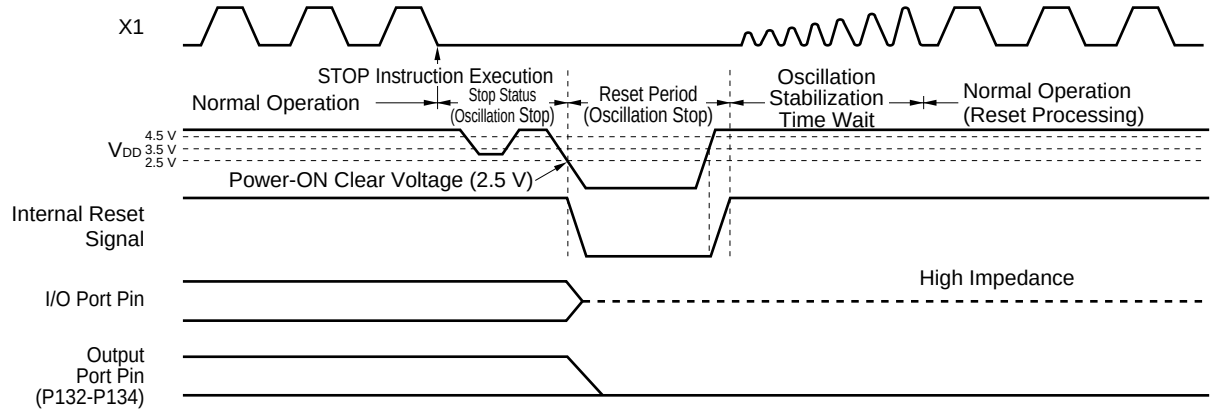
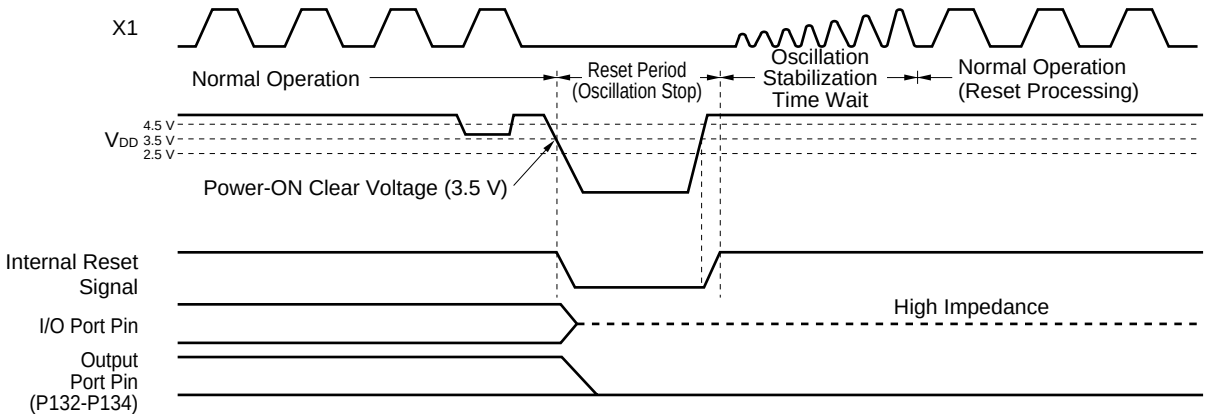


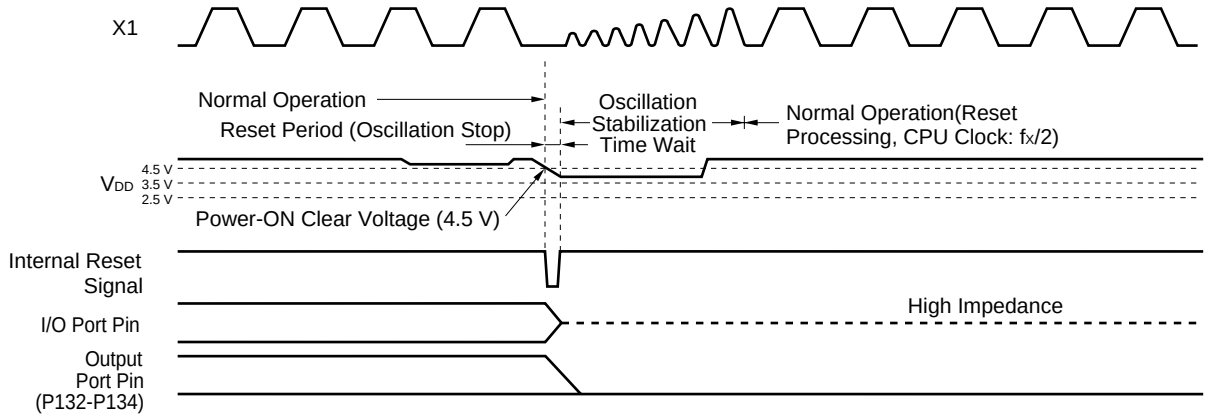
Figure 18-4. Timing of Reset by Power-ON Clear (2/2)

(c) In normal operating mode (including HALT mode)

<1> When CPU clock is $f_X/2$ or less



<2> When CPU clock is $f_X/2$



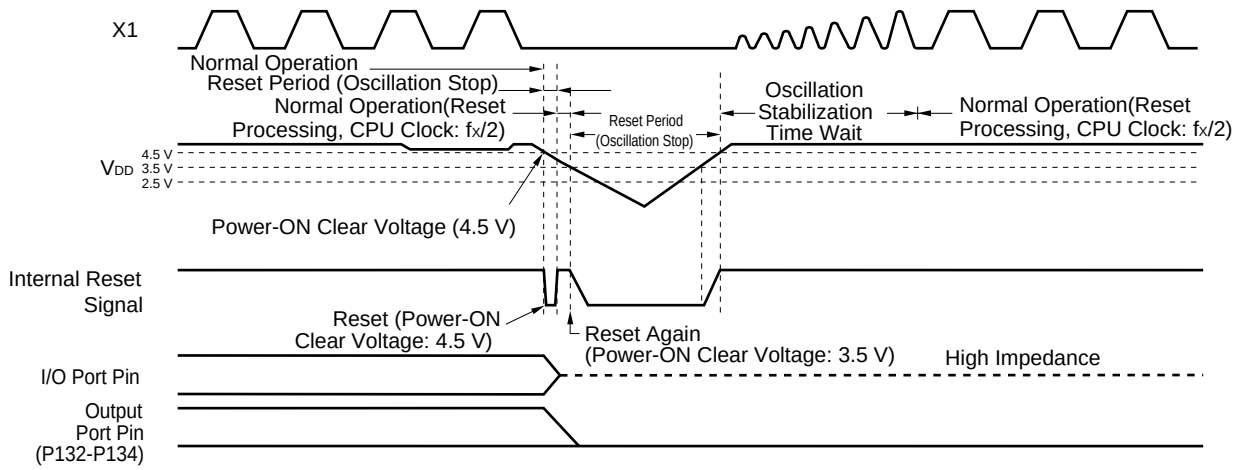


Table 18-1. Hardware Status after Reset (1/2)

Hardware		Status after Reset
Program counter (PC) Note1		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, Port 12, 13 (P0 to P3, P12, P13)	00H
	Port 4 to Port 6 (P4 to P6)	Undefined
Port mode register (PM0 to PM3, PM5, PM6, PM12)		FFH
Port mode register 4 (MM)		10H
Processor clock control register (PCC)		04H
Oscillation mode selection register (OSMS)		00H
Oscillation stabilization time select register (OSTS)		04H
8-bit timer/event counter	Timer register (TM1, TM2)	00H
	Compare registers (CR10, CR20)	00H
	Clock select register (TCL1)	00H
	Mode control registers (TMC1)	00H
8-bit timer (D/A converter)	PWM mode select register (PWMSEL)	00H
	PWM control register (PWMCR)	00H
	PWM data register (PWMR0, PWMR1, PWMR2)	01FFH
	PWM timer register (PWMTMR)	FFH

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remains unchanged after reset.

2. The status before reset is retained even after reset in the standby mode.

Table 18-1. Hardware Status after Reset (2/2)

Hardware		Status after Reset
Watchdog timer	Clock select register (TCL2)	00H
	Mode register (WDTM)	00H
Serial interface	Clock select register (TCL3)	88H
	Shift registers (SIO0, SIO1)	Undefined
	Mode registers (CSIM0)	01H
	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 specification register (ADTI)	00H
	Asynchronous serial interface mode register (ASIM)	00H
	Interrupt timing specify register (SINT)	00H
A/D converter	Mode register (ADM)	01H
	Conversion result register (ADCR)	Undefined
	Input select register (ADIS)	00H
Interrupt	Request flag register (IF0L, IF0H)	00H
	Mask flag register (MK0L, MK0H)	FFH
	Priority specification flag register (PR0L, PR0H)	FFH
	External interrupt mode register (INTM01, INTM1)	00H
	Key return mode register (KRM)	02H
	Sampling clock select register (SCS)	00H
PLL frequency synthesizer	PLL most select register (PLLMD)	00H
	PLL reference mode register (PLLRF)	0FH
	PLL unlock FF judge register (PLLUL)	Retained**Note 1**
	PLL data transfer register (PLLNS)	00H
	EO select register (EOCON)	00H
	PLL data register (PLLRL, PLLRH, PLLR0)	Undefined
Frequency counter	IF counter mode select register (IFCMD)	00H
	IF counter gate judge register (IFCJG)	00H
	IF counter control register (IFCR)	00H
	IF counter register (IFC)	0000H
Power-ON clear	POC status register (POCS)	Retained **Note 2**

Notes 1. Only reset by power-on clear becomes undefined.

2. Only reset by power-on clear becomes 01H.

18.2 Power Failure Detection Function

If reset is effected by means of power-ON clear, bit 0 (POC45) of the POC status register (POCS) is set to 1. If reset is effected by the $\overline{\text{RESET}}$ Pin or the watchdog timer, however, POC45 holds the previous status.

A power failure status can be detected by detecting this POC45 after reset by power-ON clear has been cleared (after program execution has been started from address 0000H).

Figure 18-5. POC Status Register Format

Symbol	7	6	5	4	3	2	1	⓪	Address	After Reset	R/W
POCS	0	0	0	0	0	0	0	POC45	FFBFH	Retained[Note]	R&Reset

POC45	Detects Power-ON Clear Occurrence Status
0	Power-ON clear does not occur
1[Note]	Reset is effected by power-ON clear

Note The value of this register is set to 01H only when reset is effected through power-ON clearing.

Remark The values of the special function registers, other than POCS, are the same as the values at reset that is effected by means of power-ON clear.

[MEMO]

CHAPTER 19 μPD178P018A

The μPD178P018A subseries includes the μPD178P018A as PROM version.

The μPD178P018A replaces the internal mask ROM of the μPD178018A with one-time PROM or EPROM. Table 19-1 lists the differences between the μPD178P018A and the mask ROM versions.

Table 19-1. Differences between μPD178P018A and Mask ROM Versions

Item	μPD178P018A	Mask ROM Versions
IC pin	None	Available
V_{PP} pin	Available	None
Electrical specifications	Refer to the individual data sheet.	

Caution The electrical specifications (supply current) and PLL analog characteristics differ between the μPD178P018A and mask ROM version. For mass production of the application set, therefore, be sure to check these differences.

19.1 PROM Programming

The μPD178P018A incorporates 60-Kbyte PROM as program memory. To write a program into the μPD178P018A PROM, make the device enter the PROM programming mode by setting the levels of the V_{PP} and $\overline{RESET}$ pins as specified. For the connection of unused pins, refer to **1.5 (2) PROM Programming Mode**.

Caution In case of write to the μPD178P018A, write the program in the range of addresses 0000H to EFFFH (specify the last address as EFFFH.)
The program cannot be correctly written by a PROM programmer which does not have a write address specification function.

19.1.1 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 $\overline{RESET}$ pin, the μPD178P018A is set to the PROM programming mode. This is one of the operating modes shown in Table 19-2 below according to the setting of the $\overline{CE}$, $\overline{OE}$, and $\overline{PGM}$ pins.

The PROM contents can be read by setting the read mode.

Table 19-2. PROM Programming Operating Modes

Operating Mode \ Pin	$\overline{RESET}$	V_{PP}	V_{DD}	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	D0-D7
Page data latch	L	+12.5 V	+6.5 V	H	L	H	Data input
Page write				H	H	L	High impedance
Byte write				L	H	L	Data input
Program verify				L	L	H	Data output
Program inhibit				×	H	H	High impedance
				×	L	L	
Read		+5 V	+5V	L	L	H	Data output
Output disabled				L	H	×	High impedance
Standby				H	×	×	High impedance

Remark ×: L or H

(1) Read mode

Read mode is set by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

(2) Output disable mode

If $\overline{OE}$ is set to H, data output becomes high impedance and the output disable mode is set.

Therefore, if multiple μPD178P018A are connected to the data bus, data can be read from any one device by controlling the $\overline{OE}$ pin.

(3) Standby mode

Setting $\overline{CE}$ to H sets the standby mode.

In this mode, data output becomes high impedance irrespective of the status of $\overline{OE}$.

(4) Page data latch mode

Setting $\overline{CE}$ to H, $\overline{PGM}$ to H, and $\overline{OE}$ 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 $\overline{PGM}$ pin while $\overline{CE}$ = H and $\overline{OE}$ = H. After this, program verification can be performed by setting $\overline{CE}$ to L and $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times (X ≤ 10).

(6) Byte write mode

A byte write is executed by applying a 0.1-ms program pulse (active-low) to the $\overline{PGM}$ pin while $\overline{CE}$ = L and $\overline{OE}$ = H. After this, program verification can be performed by setting $\overline{OE}$ to L.

If programming is not performed by one program pulse, repeated write and verify operations are executed X times (X ≤ 10).

(7) Program verify mode

Setting $\overline{CE}$ to L, $\overline{PGM}$ to H, and $\overline{OE}$ to L sets the program verify mode.

After writing is performed, this mode should be used to check whether the data was written correctly.

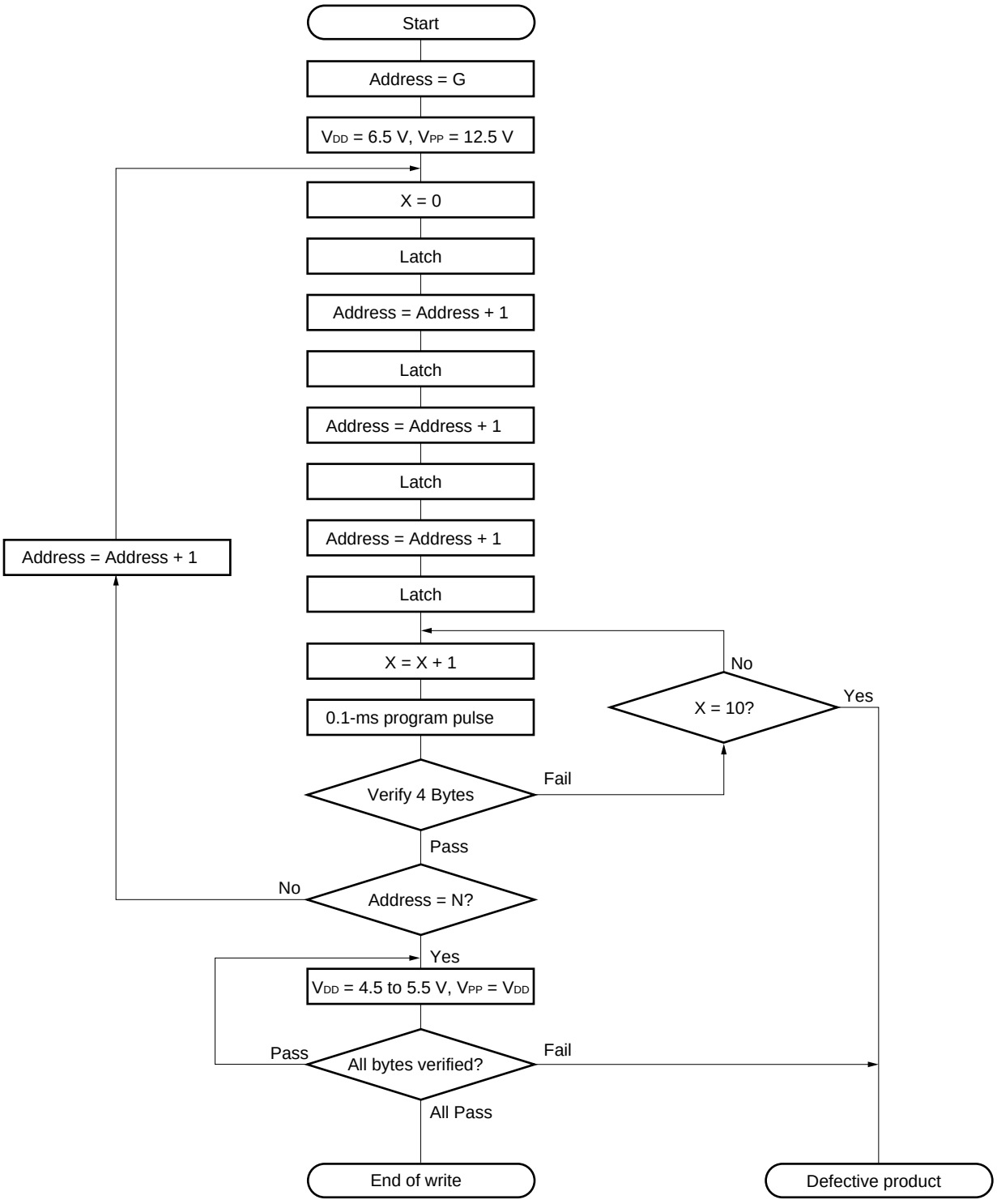
(8) Program inhibit mode

The program inhibit mode is used when the $\overline{OE}$ pins, V_{PP} pins and pins D0 to D7 of multiple μPD178P018As are connected in parallel and any one of these devices must be written to.

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 $\overline{PGM}$ pin driven high.

19.1.2 PROM write procedure

Figure 19-1. Page Program Mode Flowchart



G = Start address

N = Last address of program

Figure 19-2. Page Program Mode Timing

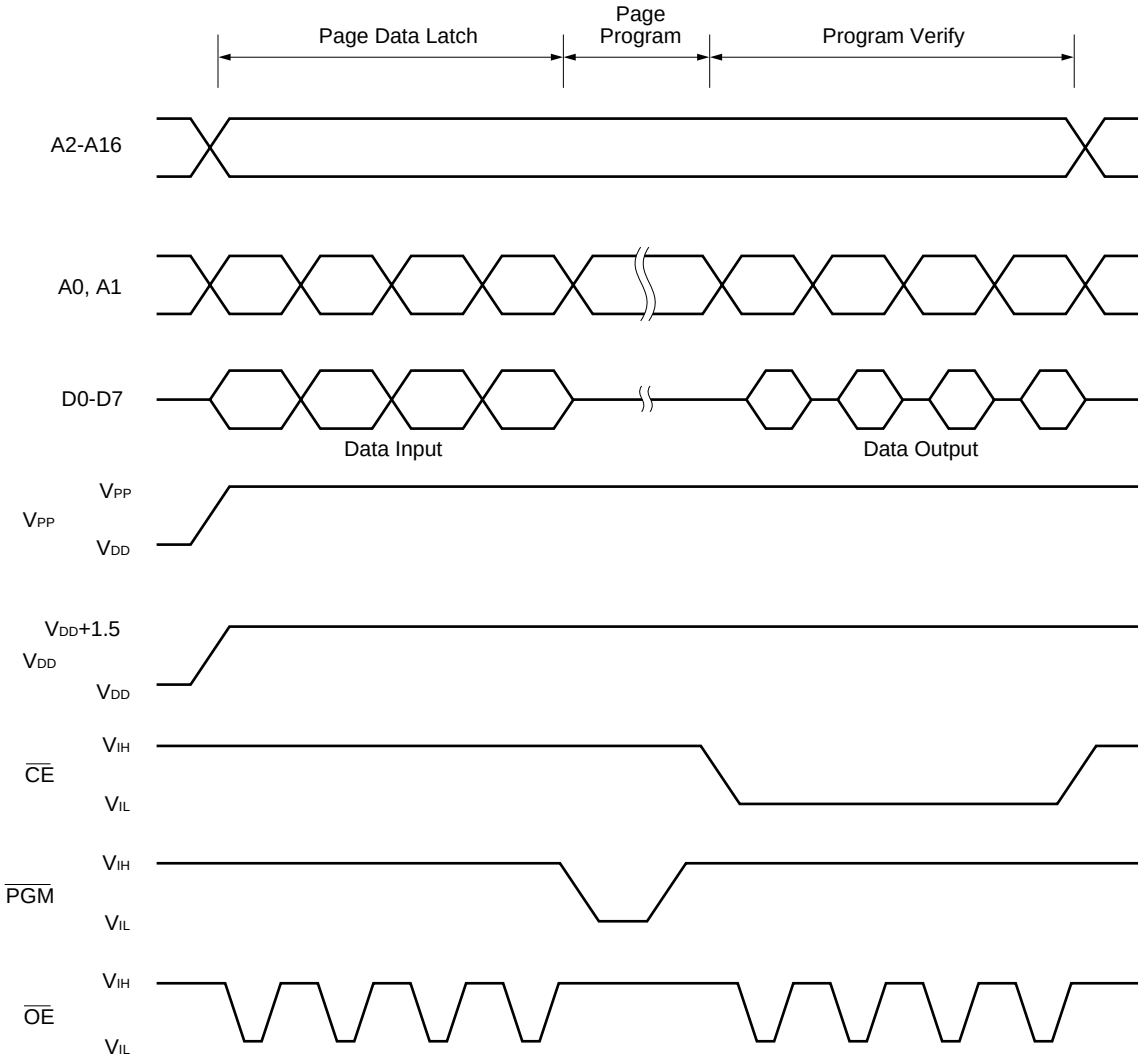
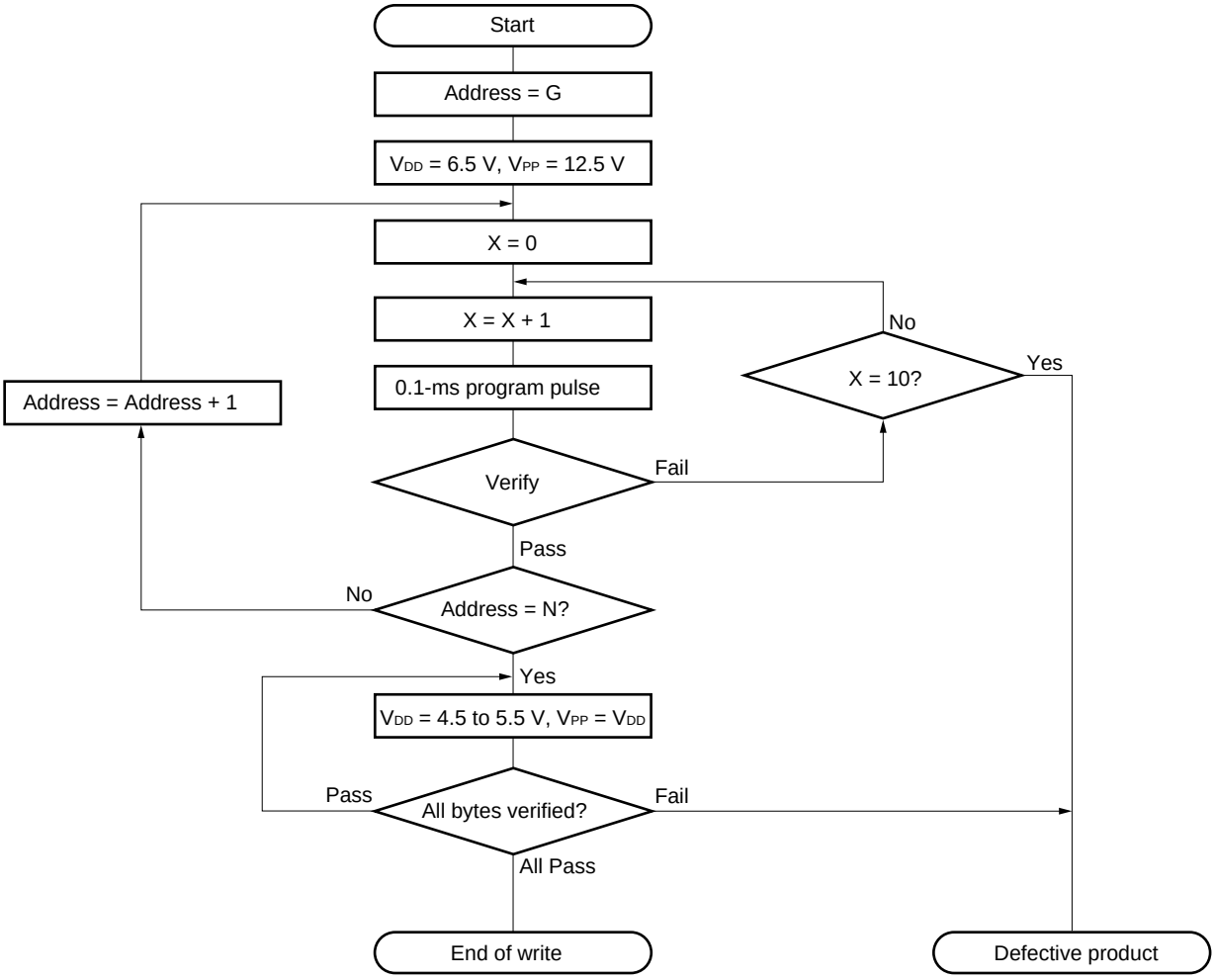


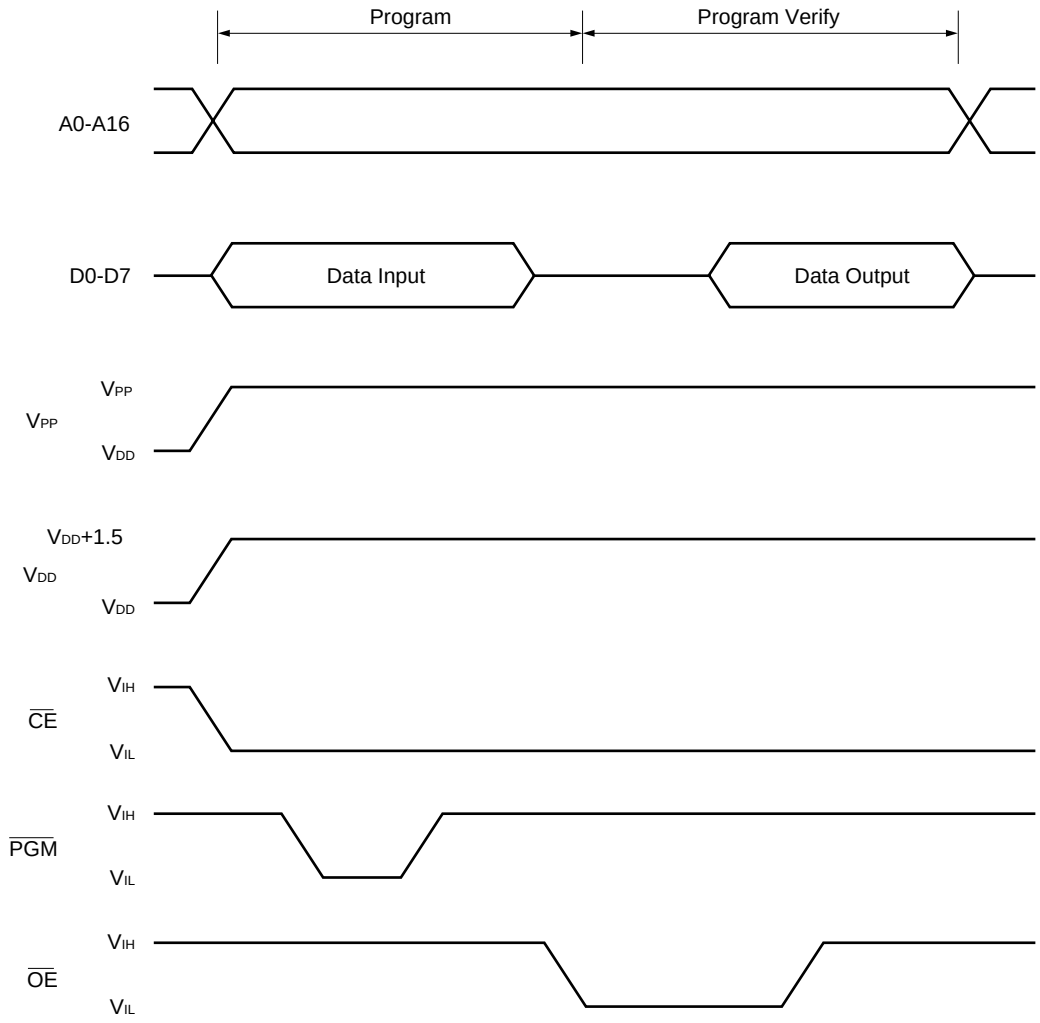
Figure 19-3. Byte Program Mode Flowchart



G = Start address

N = Last address of program

Figure 19-4. Byte Program Mode Timing



Cautions 1. Be sure to apply V_{DD} before applying V_{PP}, and remove it after removing V_{PP}.

2. V_{PP} must not exceed +13.5 V including overshoot voltage.

3. Disconnecting/inserting the device from/to the on-board socket while +12.5 V is being applied to the V_{PP} pin may have an adverse affect on device reliability.

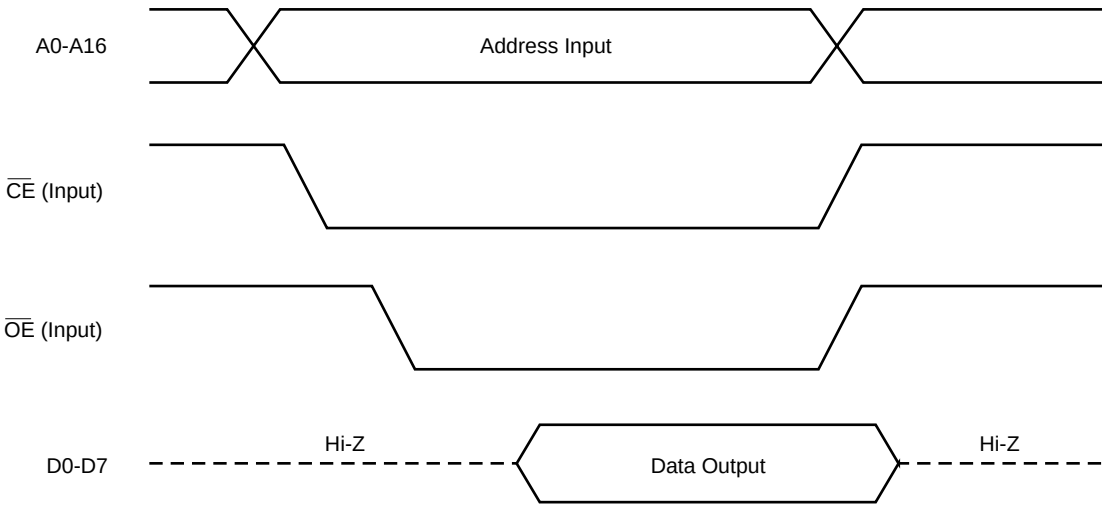
19.1.3 PROM reading procedure

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 (2) PROM Programming Mode**.
(2) Supply +5 V to the V_{DD} and V_{PP} pins.
(3) Input the address of data to be read to pins A0 through A16.
(4) Read mode is entered.
(5) Data is output to pins D0 through D7.

The timing for steps (2) through (5) above is shown in Figure 19-5.

Figure 19-5. PROM Read Timing



19.2 Erasure Procedure (μPD178P018AKK-T Only)

With the μPD178P018AKK-T, it is possible to erase (or set all contents to FFH) the data contents written in the program memory, and rewrite the memory.

The data can be erased by exposing the window to light with a wavelength of approximately 400 nm or shorter. Typically, data is erased by 254-nm ultraviolet light rays. The minimum lighting level to completely erase the written data is shown below.

- UV intensity × exposure time: 30 W·s/cm^2 or more
- Exposure time: More than 40 minutes (using a 12,000 μW/cm^2 ultraviolet lamp. A longer exposure time may be required in case of deterioration of the ultraviolet lamp or dirt on the package window).

When erasing written data, remove any filter on the window and place the device within 2.5 cm of the lamp tube.

19.3 Opaque Film Masking the Window (μPD178P018AKK-T Only)

To prevent unintentional erasure of the EPROM contents by light and to prevent internal circuits from mulfunction due to light coming in through the erasure window, mask the window with opaque film after writing the EPROM.

19.4 Screening of One-Time PROM Versions

One-time PROM versions (μPD178P018AGC-3B9) cannot be fully tested by NEC before shipment due to the structure of one-time PROM. Therefore, after users have written data into the PROM, screening should be implemented by user: that is, store devices at high temperature for one day as specified below, and verify their contents after the devices have returned to room temperature.

Storage Temperature	Storage Time
125°C	24 hours

[MEMO]

CHAPTER 20 INSTRUCTION SET

This chapter describes each instruction set of the μPD178018A subseries as list table. For details of its operation and operation code, refer to the separate document **78K/0 series User's Manual—Instruction (U12326E)**.

20.1 Legends

20.1.1 Operand symbols and description

Operands are written in “Operand” column of each instruction in accordance with the description of the instruction operand symbols (refer to the assembler specifications for detail). When there are two or more descriptions, select one of them. Alphabetic letters in capitals and symbols, #, !, $ and [ ] are key words and must be written as they are. Each symbol has the following meaning.

- # : Immediate data specification
- ! : Absolute address specification
- $: Relative address specification
- [ ] : Indirect address specification

In the case of immediate data, write an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register symbols, 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.

Table 20-1. Operand Symbols and Descriptions

Symbol	Description
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7),
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol**Note**
sfrp	Special-function register symbol (16-bit manipulatable register even addresses only)**Note**
saddr	FE20H-FF1FH Immediate data or labels
saddrp	FE20H-FF1FH Immediate data or labels (even address only)
addr16	0000H-FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H-0FFFH Immediate data or labels
addr5	0040H-007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special-function register symbols, refer to **Table 3-2 Special-Function Register List**.

20.1.2 Description of “operation” column

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
AX	:	AX register pair; 16-bit accumulator
BC	:	BC register pair
DE	:	DE register pair
HL	:	HL register pair
PC	:	Program counter
SP	:	Stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
RBS	:	Register bank select flag
IE	:	Interrupt request enable flag
NMIS	:	Non-maskable interrupt servicing flag
( )	:	Memory contents indicated by address or register contents in parentheses
$\times_H$, $\times_L$	:	High-order 8 bits and low-order 8 bits of 16-bit register
$\wedge$	:	Logical product (AND)
$\vee$	:	Logical sum (OR)
$\veebar$	:	Exclusive logical sum (exclusive OR)
$\overline{\quad}$	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

20.1.3 Description of “flag operation” column

(Blank)	:	Nt affected
0	:	Cleared to 0
1	:	Set to 1
×	:	Set/cleared according to the result
R	:	Previously saved value is restored

20.2 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	1	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"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjust	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit			×
		CY, PSW. bit	3	–	7	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$			×

Notes 1. When the internal high-speed RAM area is accessed or instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

Instruction Group	Mnemonic	Operands	Byte	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	$(SP - 1) \leftarrow (PC + 3)_H$, $(SP - 2) \leftarrow (PC + 3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP - 2$			
	CALLF	!addr11	2	5	–	$(SP - 1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow$ addr11, $SP \leftarrow SP - 2$			
	CALLT	[addr5]	1	6	–	$(SP - 1) \leftarrow (PC + 1)_H$, $(SP - 2) \leftarrow (PC + 1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP - 2$			
	BRK		1	6	–	$(SP - 1) \leftarrow$ PSW, $(SP - 2) \leftarrow (PC + 1)_H$, $(SP - 3) \leftarrow (PC + 1)_L$, $PC_H \leftarrow$ (003FH), $PC_L \leftarrow$ (003EH), $SP \leftarrow SP - 3$, $IE \leftarrow 0$			
	RET		1	6	–	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
	RETI		1	6	–	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP + 2)$, $SP \leftarrow SP + 3$, $NMIS \leftarrow 0$	R	R	R
	RETB		1	6	–	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP + 2)$, $SP \leftarrow SP + 3$	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	$(SP - 1) \leftarrow$ PSW, $SP \leftarrow SP - 1$			
		rp	1	4	–	$(SP - 1) \leftarrow rp_H$, $(SP - 2) \leftarrow rp_L$, $SP \leftarrow SP - 2$			
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP + 1$	R	R	R
		rp	1	4	–	$rp_H \leftarrow (SP + 1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
	MOVW	SP, #word	4	–	10	$SP \leftarrow$ word			
		SP, AX	2	–	8	$SP \leftarrow AX$			
		AX, SP	2	–	8	$AX \leftarrow SP$			
Unconditional branch	**BR**	!addr16	3	6	–	$PC \leftarrow$ addr16			
		$addr16	2	6	–	$PC \leftarrow PC + 2 +$ jdisp8			
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
Conditional branch	**BC**	$addr16	2	6	–	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	$PC \leftarrow PC + 2 +$ jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	$PC \leftarrow PC + 2 +$ 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

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

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

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the PCC register.

2. This clock cycle applies to internal ROM program.

20.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ Second 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

First Operand \ Second 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

First Operand \ Second Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

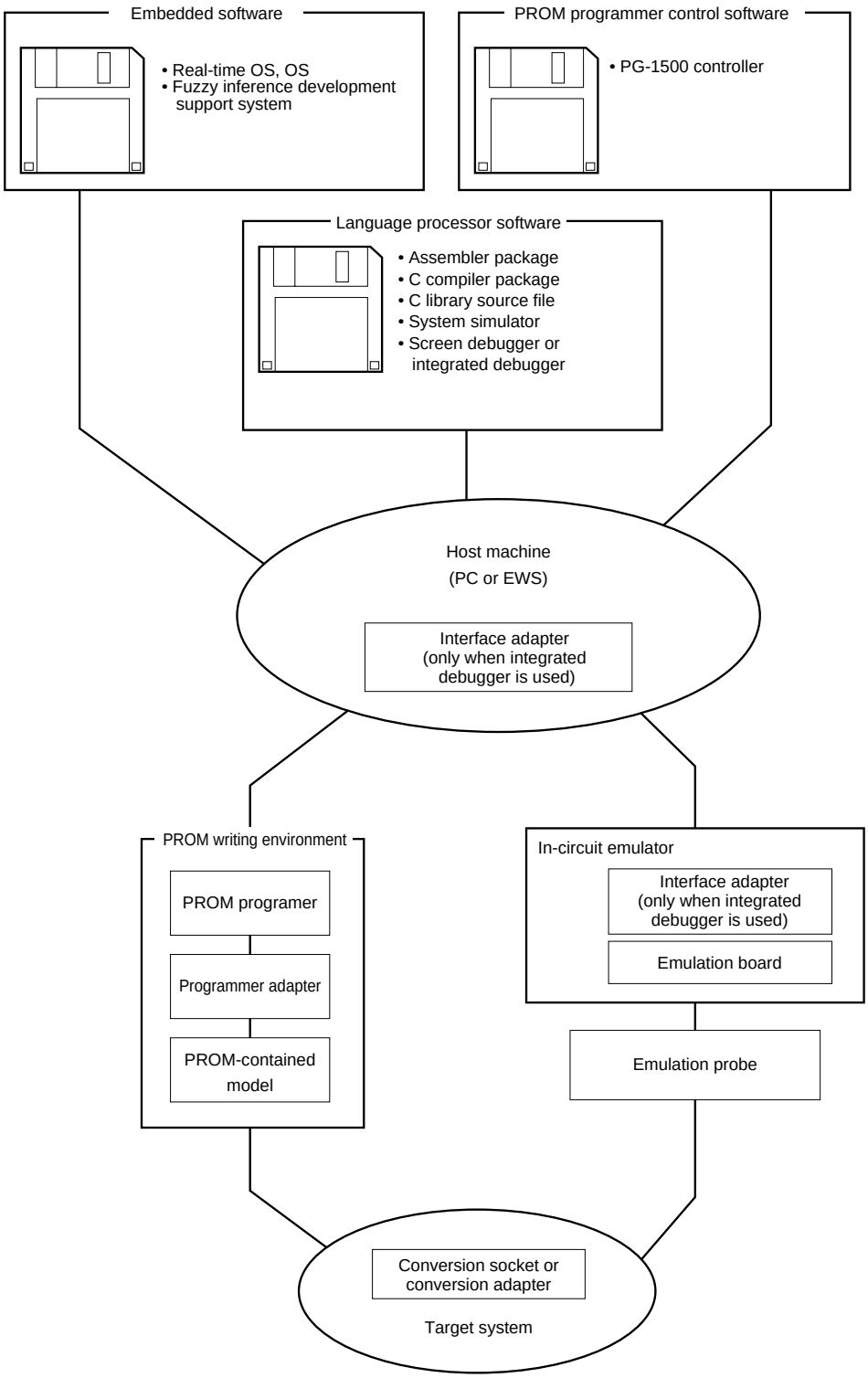
ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

[MEMO]

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems which employ the μPD178018A subseries. Figure A-1 shows the configuration example of the tools.

Figure A-1. Development Tool Configuration



A.1 Language Processing Software

RA78K/0 Assembler Package	This assembler converts a program written in mnemonics into an object code executable with a microcomputer. Further, this assembler is provided with functions capable of automatically creating symbol tables and branch instruction optimization. This data file is used together with DF178018 device file (option).
	Part Number: μS××××RA78K0
CC78K/0 C Compiler Package	This compiler converts a program written in C language into an object code executable with a microcontroller. This data file is used together with RA78K/0 assembler package and DF178018 device file (option).
	Part Number: μS××××CC78K0
DF178018 **Note** Device File	File containing information unique to the devices. This data file is used together with RA78K/0, CC78K/0, SM78K0, ID78K0 and SD78K/0.
	Part Number: μS××××DF178018
CC78K/0-L C Library Source File	Source program of a function configurating object library included in CC78K/0 C compiler. This file is necessary when customers change the object library in CC78K/0 following their specifications.
	Part Number: μS××××CC78K0-L

Note This device file can be used for any of RA78K/0, CC78K/0, SM78K0, ID78K0 and SD78K/0.

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× RA78K0
μS×××× CC78K0
μS×××× DF178018
μS×××× CC78K0-L

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)**Note**	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **A.4**.	3.5-inch 2HC
7B10			5-inch 2HC
3H15	HP9000 series 300™	HP-UX™ (rel.7.05B)	Cartidge tape (QIC-24)
3P16	HP9000 series 700™	HP-UX (rel.9.01)	Digital audio tape (DAT)
3K15	SPARCstation™	SunOS™ (rel.4.1.1)	Cartidge tape (QIC-24)
3M15	EWS4800 series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

A.2 PROM Programming Tools

A.2.1 Hardware

PG-1500 PROM Programmer	This is a PROM programmer capable of programming the single-chip microcontroller incorporating PROM by manipulating from the stand-alone or host machine through connection of an optional PROM programmer adapter and attached board. It can also program repre sentative PROMs ranging from 256K bits to 4M bits.
PA-178P018AGC PA-178P018AKK-T PROM Programmer Adapter	PROM programmer adapter for the μPD178P018A. Used connected to the PG-1500. PA-178P018AGC : 80-pin plastic QFP (GC-3B9 type) PA-178P018AKK-T : 80-pin ceramic WQFN (KK-T type)

A.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 : μS××××PG1500

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× PG1500

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS	3.5-inch 2HD
5A10		(ver. 3.30 to 6.2)**Note**	5-inch 2HD
7B13	IBM PC/AT or	Refer to **A.4**.	3.5-inch 2HD
7B10	compatible machine		5-inch 2HC

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

A.3 Debugging Tools

A.3.1 Hardware

IE-78000-R-A In-Circuit Emulator (supporting integrated debugger)	This in-circuit emulator debugs hardware and software when an application system using the 78K/0 series is developed. It supports the integrated debugger (ID78K0). This emulator is used in combination with an emulation probe and an interface adapter that connects the emulator with the host machine.
IE-70000-98-IF-B Interface Adapter	Adapter necessary when using the PC-9800 series (except the notebook type) as the host machine of the IE-78000-R-A.
IE-70000-98N-IF Interface Adapter	Adapter and cable necessary when using the notebook type PC-9800 series as the host machine of the IE-78000-R-A.
IE-70000-PC-IF-B Interface Adapter	Adapter necessary when using IBM PC/AT as the host machine of the IE-78000-R-A.
IE-78000-R-SV3 Interface Adapter	Adapter and cable necessary when using an EWS as the host machine of the IE-78000-R-A. This cable is connected to the board in the IE-78000-R-A. As Ethernet™, 10Base-5 is supported. If the other methods are used, a commercially available conversion adapter is necessary.
IE-78000-R In-Circuit Emulator (supporting screen debugger)	This in-circuit emulator debugs hardware and software when an application system using the 78K/0 series is developed. It supports the screen debugger (SD78K/0). This emulator is used in combination with an emulation probe. When this emulator is connected with a host machine and PROM programmer, efficient debugging can be performed.
IE-178018-R-EM Emulation Board	This board emulates the peripheral hardware peculiar to a device (4.5 to 5.5 V). It is used in combination with an in-circuit emulator.
EP-78230GC-R Emulation Probe	This is a probe to connect an in-circuit emulator and target system. It is for a 80-pin plastic QFP (GC-3B9 type). One 100-pin conversion socket, EV-9200GC-80, which facilitates development of the target system is supplied as an accessory.
EV-9200GC-80 conversion socket (refer to **Figure A-2.**)	This conversion socket connects the board of the target system created for mounting a 80-pin plastic QFP (GC-3B9 type) and the EP-78230GC-R.
EV-9900	This is a jig used for removing the μPD178P018AKK-T from the EV-9200GC-80.

Remark Five EV-9200GC-80s are available as a set.

A.3.2 Software (1/3)

SM78K0 System Simulator	This simulator simulates operations of the target system from a Windows-installed host computer, enabling debuggung in C source level or assembler level. By using SM78K0, logical and performance verification processes can be performed independently of hardware development work without using in-circuit emulator in-circuit emulator, which leads to reduction in development workload and improvement in software quality. This system simulator is used together with the DF178018 device file (DF78064, option).
	Part Number : μS××××SM78K0

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× SM78K0

××××	Host Machine	OS	Supply Medium
AA13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)[Note] + Windows (ver. 3.0 and 3.1)	3.5-inch 2HD
AB13	IBM PC/AT or compatible machine (on Japanese Windows)	Refer to **A.4**.	3.5-inch 2HC
BB13	IBM PC/AT or compatible machine (on English Wondows)		

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

A.3.2 Software (2/3)

ID78K0 Integrated Debugger	This is a control program that debugs the 78K/0 series. It employs Windows on a personal computer and OSF/Motif™ on EWS as a graphical user interface, and provides the environment and operability conforming to these interfaces. Moreover, C language debugging functions are improved, and the trace result can be displayed at C language level by using a window integrating function that associates the source program, disassemble display, and memory display with the trace result. In addition, the debugging of a program can be made more efficient by using a real-time OS by incorporating function expansion modules such as a task debugger and system performance analyzer. This debugger is used in combination with an optional device file (DF178018).
	Part Number : μS××××ID78K0

Remark ×××× in the parts number differs depending on the host machine and OS used.

μS×××× SM78K0

××××	Host Machine	OS	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 or compatible machine (Japanese Windows)	Refer to **A.4**.	3.5-inch 2HC
BB13	IBM PC/AT or compatible machine (English Windows)		
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 MS-DS Ver. 5.0 or above has a task swap function, but this function cannot be used with the above software.

A.3.2 Software (3/3)

SD78K/0 Screen Debugger	This debugger is a program which controls the IE-78000-R in-circuit emulator from the host computer. The in-circuit emulator must be connected to the host computer via a serial interface (RS-232-C) cable. This debugger is used together with the DF178018 device file (option).
	Part Number : μS××××SD78K0
DF178018 **Note** Device File	File containing information unique to the devices. This device file is used together with the SM78K0, CC78K/0, RA78K0, ID78K0, and SD78K/0 (option).
	Part Number : μS××××DF178018

Note This device file can be used for any of RA78K/0, CC78K/0, SM78K0, ID78K0 and SD78K/0.

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS×××× SD78K0
μS×××× DF178018

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)**Note**	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **A.4**.	3.5-inch 2HC
7B10			5-inch 2HC

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

A.4 Operating Systems for IBM PC

The following operating systems are available for IBM PC.

If SM78K0, ID78K0, and FE9200 (refer to **B.2 Fuzzy Inference Development Support System**) are to be operated, Windows version 3.0 or 3.1 is also required.

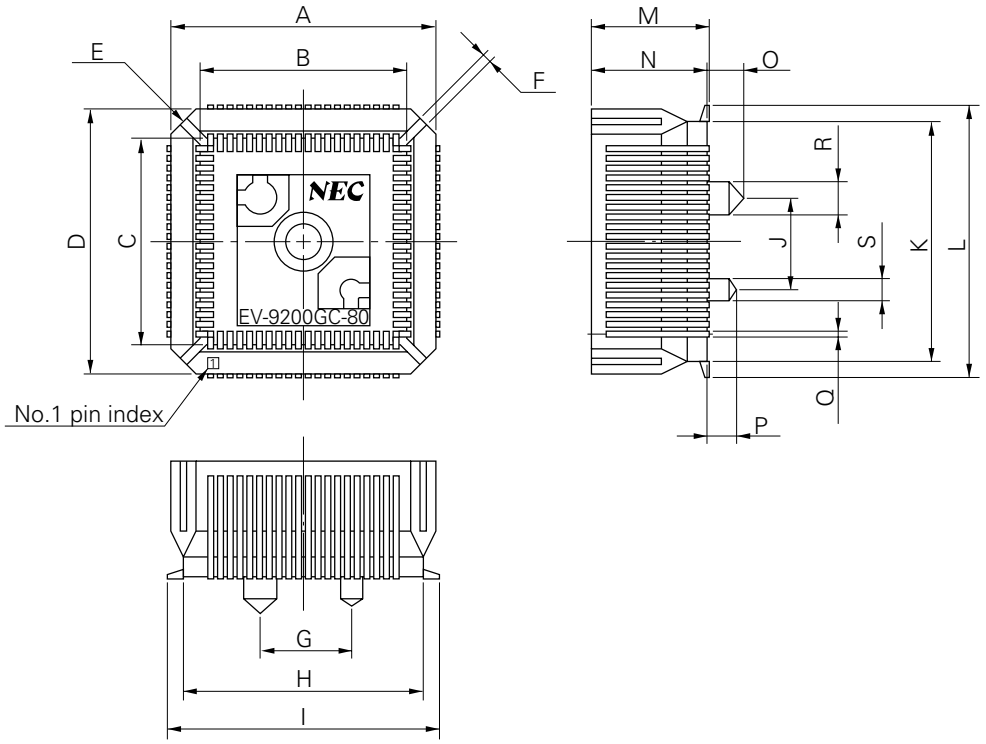
OS	Version
PC DOS	Version 5.02 through 6.3
	J6.1/V through J6.3/V[Note]
IBM DOS™	J5.02/V[Note]
MS-DOS	Version 5.0 through 6.2
	5.0/V[Note] through 6.2V[Note]

Note Supports English versions only.

Caution The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

Drawing for Conversion Socket (EV-9200GC-80) Package and Recommended Board Mounting Pattern

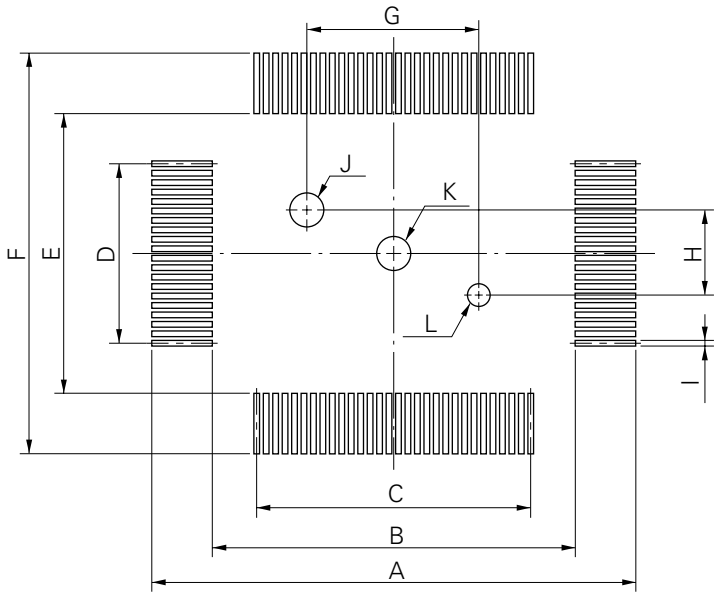
Figure A-2. EV-9200GC-80 Package Drawing (For Reference Only)



EV-9200GC-80-G0

ITEM	MILLIMETERS	INCHES
A	18.0	0.709
B	14.4	0.567
C	14.4	0.567
D	18.0	0.709
E	4-C 2.0	4-C 0.079
F	0.8	0.031
G	6.0	0.236
H	16.0	0.63
I	18.7	0.736
J	6.0	0.236
K	16.0	0.63
L	18.7	0.736
M	8.2	0.323
O	8.0	0.315
N	2.5	0.098
P	2.0	0.079
Q	0.35	0.014
R	ϕ2.3	ϕ0.091
S	ϕ1.5	ϕ0.059

Figure A-3. Recommended Board Mounting Pattern EV-9200GC-80 (For Reference Only)



EV-9200GC-80-P1

ITEM	MILLIMETERS	INCHES
A	26.3	1.035
B	21.6	0.85
C	0.65±0.02 × 29=18.85±0.05	$0.026^{+0.001}_{-0.002} \times 1.142 = 0.742^{+0.002}_{-0.002}$
D	0.65±0.02 × 19=12.35±0.05	$0.026^{+0.001}_{-0.002} \times 0.748 = 0.486^{+0.003}_{-0.002}$
E	15.6	0.614
F	20.3	0.799
G	12±0.05	$0.472^{+0.003}_{-0.002}$
H	6±0.05	$0.236^{+0.003}_{-0.002}$
I	0.35±0.02	$0.014^{+0.001}_{-0.001}$
J	ϕ2.36±0.03	$\phi 0.093^{+0.001}_{-0.002}$
K	ϕ2.3	ϕ0.091
L	ϕ1.57±0.03	$\phi 0.062^{+0.001}_{-0.002}$

Caution Dimensions of mount pad for EV-9200 and that for target device (QFP) may be different in some parts. For the recommended mount pad dimensions for QFP, refer to "SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL" (C10535E).

APPENDIX B EMBEDDED SOFTWARE

This section describes the embedded software which are provided for the μPD178018A subseries to allow users to develop and maintain the application program for these subseries.

B.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). When using the RX78K/0, the RA78K/0 assembler package (option) and the device file (DF178018) are necessary.
	Part Number: μS××××RX78013-ΔΔΔΔ

Caution When purchasing the RX78K/0, fill in the purchase application form in advance, and sign the Use Approval Contract.

Remark ×××× and ΔΔΔΔ of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××RX78013-ΔΔΔΔ

ΔΔΔΔ	Product outline	Max. No. for use in mass production
001	Evaluation object	Do not use for mass production
100K	Mass-production object	100,000
001M		1,000,000
010M		10,000,000
S01	Source program	Source program for mass-production object

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)[Note]	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **A.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 tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

B.1 Real-time OS (2/2)

MX78K0 OS	MX78K/0 is an OS for subsets based on the μITRON specification. Supplied with the MX78K0 nucleus. This OS manages tasks, events, and time. In task management operation, it controls the execution orders of tasks, and switches processing to the task to be executed next.
	Part Number: μS××××MX78K0-ΔΔΔ

Remark ×××× and ΔΔΔ of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××MX78K0-ΔΔΔ

ΔΔΔ	Product outline	Remark
001	Evaluation object	Use for preproduction.
XX	Mass-production object	Use for mass-production.
S01	Source program	Available only when purchasing mass-production object

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 - 6.2)**Note**	3.5-inch 2HD
5A10			5-inch 2HD
7B13	IBM PC/AT or compatible machine	Refer to **A.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 tape (DAT)
3K15	SPARCstation	SunOS (rel.4.1.1)	Cartridge tape (QIC-24)
3M15	EWS4800 series (RISC)	EWS-UX/V (rel.4.0)	

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

B.2 Fuzzy Inference Development Support System

FE9000/FE9200 Fuzzy Knowledge Data Creation Tool	Program supporting input of fuzzy knowledge data (fuzzy rule and membership function), editing (edit), and evaluation (simulation). FE9200 operates on Windows.
	Part Number: μS××××FE9000 (PC-9800 series) μS××××FE9200 (IBM PC/AT or compatible machine)
FT9080/FT9085 Translator	Program converting fuzzy knowledge data obtained by using fuzzy knowledge data preparation tool to RA78K/0 assembler source program.
	Part Number: μS××××FT9080 (PC-9800 series) μS××××FT9085 (IBM PC/AT or compatible machine)
FI78K0 Fuzzy Inference Module	Program executing fuzzy inference. Fuzzy inference is executed by linking fuzzy knowledge data converted by translator.
	Part Number: μS××××FI78K0 (PC-9800 series, IBM PC/AT or compatible machine)
FD78K0 Fuzzy Inference Debugger	Support software evaluating and adjusting fuzzy knowledge data at hardware level by using in-circuit emulator.
	Part Number: μS××××FD78K0 (PC-9800 series, IBM PC/AT or compatible machine)

Remark ×××× of the part number differs depending on the host machine and OS used. Refer to the table below.

μS××××FE9000
μS××××FT9080
μS××××FI78K0
μS××××FD78K0

××××	Host Machine	OS	Supply Medium
5A13	PC-9800 series	MS-DOS (ver. 3.30 to 6.2)[Note]	3.5-inch 2HD
5A10			5-inch 2HD

μS××××FE9200
μS××××FT9085
μS××××FI78K0
μS××××FD78K0

××××	Host Machine	OS	Supply Medium
7B13	IBM PC/AT or compatible machine	Refer to **A.4**.	3.5-inch 2HC
7B10			5-inch 2HC

Note The task swap function is not available with this software though the function is provided in MS-DOS version 5.0 or later.

[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	❑	❑	❑	❑